To Elizabeth, Amanda and Hermione

The Cybernetics of Human Learning and Performance

A Guide to Theory and Research

Professor Gordon Pask

Hutchinson Educational

Hutchinson & Co (Publishers) Ltd
3 Fitzroy Square, London W1

London Melbourne Sydney Auckland
Wellington Johannesburg Cape Town
and agencies throughout the world

First published February 1975

ISBN 0 09 119490 3

Text set in 10/12 pt. Monotype Times New Roman, printed by letterpress, and bound in Great Britain at The Pitman Press, Bath

Contents

Preface

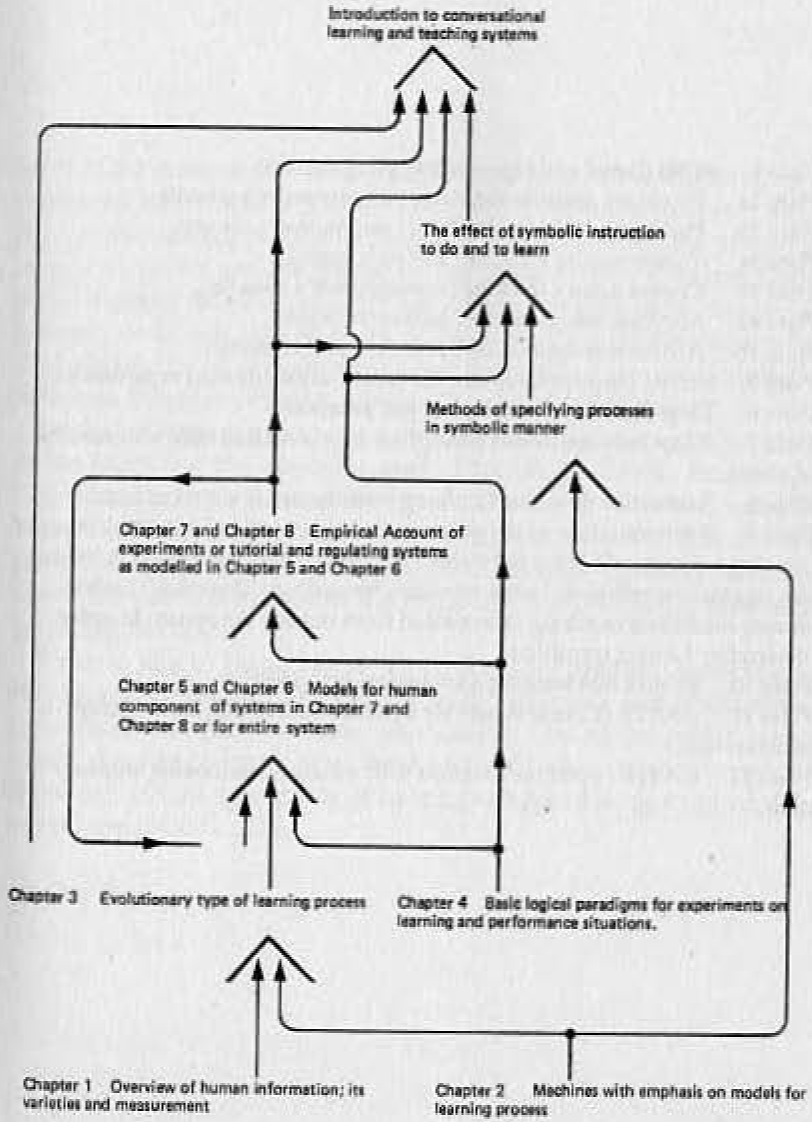
This book assumes a background knowledge commensurate with having read its predecessor *An Approach to Cybernetics* (Pask, 1961) of which some passages are a direct continuation. All that is needed from a system/cybernetic point of view is available in Ashby's (1965) lucid and still up-to-date *Introduction to Cybernetics*, though the reader with a bent and liking for mathematics will find Glushkov's (1966), *Introduction to Cybernetics* profound and comprehensive (the nearly identical title is due to translation). There is a glossary covering all essential technical terms at the end of the book. Readers may find this helpful, not so much because of obscure symbolism (very little is employed) but because ordinary language phrases are used, from time to time, with rather exact meaning; in order to avoid symbols. This trick is played quite often with logical and mathematical terms; sometimes with the relatively comprehensible jargon of psychology, educational science, and philosophy. The other offending speciality is electronics. Knowledge of the subject is unimportant, since the function of components is explained or illustrated. Some otherwise abstract notions are made tangible by the description of electrical machinery; people who are versed in the field may find the details interesting and amusing (especially in Chapter 5 where methods are quaintly redolent of the late 1950s and early 1960s). But there is no need to labour these points; only the function of the machinery bears directly upon the main theme.

An overview of the argument is charted on the next page. Nothing about the chosen order of presentation is sacrosanct. Though several back-tracking and forward-looking excursions are suggested in the text, other routes are possible. The chart shows the main prerequisites and precedences that any such forays need to observe.

The argument in this book is extended to cognition and innovation in a sequel provisionally entitled 'The Cybernetics of Intellect, Imagination and Individuality', Hutchinson (forthcoming). The phrase, 'the next volume', used from time to time in the text, refers to this book.

Professor Gordon Pask 1975

RELATIONSHIPS BEETWEEN THE CHAPTERS
Main pre-requisites and precedences.



List of Plates

Acknowledgements

Most of the research into perceptual motor and complex skill learning was supported (under various grants) by the Office of Scientific Research, USAF. All the research into intellectual learning and the educational process was supported (under various grants) by the Social Science Research Council and is ongoing. Some of the more specific projects were supported by other patrons; as follows: the Ministry of Defence, Dept. of Employment and Productivity; Dept. of the Environment; Burroughs Research Corporation; Solartron Electronic Group; the Gas Council of Great Britain.

Several colleagues of exceptionally long standing have contributed both to the ideas and the empirical work: Prof. B. N. Lewis; Prof. H. Von Foerster; R. T. McKinnon Wood; A. D. Watts; Bernard Scott; and Dionysius Kallikourdis. Numerous associates, mentioned in the text or the references, are responsible for specialised findings. Particular thanks are due to Ms Elaine Hopton and Karl Acton of Hutchinson for their help in preparing the book.

I would like to thank Mr B. C. E. Scott for reading, correcting and editing the manuscript and placing both references and illustrations in good order and Miss Kirshner who assisted him in the earlier stages. I would also like to thank Miss Barsby without whose management of Mr Scott, me, and the typewriting of an unruly wadge of manuscript,we should never have got it to press.

Gordon Pask

Introduction

Since my last book (Pask, 1961) large changes and advances have taken place in cybernetics (or, equivalently, apart from a slightly different orientation) in general system theory. Similar transformations have also taken place in the structure, interpretation and methodology of the special fields of study with which cybernetics as an interdisciplinary pursuit is chiefly concerned. These transformations have the magnitude and coherence of a renaissance like phenomenon; so far as I can see they have no particular national boundaries and they are typically fostered by rather small groups of people, closely knit enough to catalyse conceptual developments, and interacting with one another across the usual geographical and cultural dividing lines.

One pragmatically crucial feature of cybernetics or general system theory is to provide a language in which this interaction can take place. It was the intention of the pioneers (Ashby, Beer, Weiner, McCulloch, Svoboda, Glushkov, Ivahnenko, Bateson, Waddington, Von Bertalanffy, Rapaport, Von Neumann, Von Foerster, and Rashevsky) that this should be so. The theme of Margaret Mead (at the General Systems meeting in Vienna in 1972) that it *is* so and the responsibility for this happy development lies, though it is rarely acknowledged, with a body of scientific coordinators; people with a flair for organisation as well as research; for example Boulanger and Hermann (in cybernetics) and their counterparts in the field of general systems.

The scope of the 'renaissance phenomenon' is very wide. For example it involves architecture, molecular biology, genetics, epistemology, even physics. But ethology, the social and psychological sciences, together with computation and information science are more overtly and deeply implicated than the others. This book makes no attempt to give a truly comprehensive picture of what is going on and touches only incompletely upon the fields that are surveyed (psychology, social science, teaching and training, the description of tasks and bodies of learnable knowledge, some facets of computation). I decided (after some efforts to write in a more general vein) to recount, for the most part, only research in my own laboratory in which I have directly participated. On the one hand, due to a generalist inclination, these researches, theoretical developments and methods have spanned quite a wide spectrum so that my own journey is representative of other people's voyages; on the other hand there are few things quite as tedious as books full of dilutions and regurgitations of work that ought to be read in the original. It is certainly not claimed that the examples employed to point out salient developments are optimum in any absolute sense; many colleagues have more elegant or incisive demonstrations but, like a good guide book (Egon Ronay's *Guide to Food and Wine* or Kate Simmond's *Guide to London*) the material is first hand and unashamedly personalised. The reader who likes to do so may use the book in this way, as a guide book to cybernetics or general system theory.

Whoever uses the book as a guide is cautioned of some outstanding but easily remediable deficiences. The chief empirical content is concentrated in two fields: (a) The acquisition of skill, which is intimately related to issues of selective attention and the mechanisms of long and short-term memory (Chapters 5–8; parts of Chapters 10 and 11), (b) intellectual and symbolic learning of the kind encountered in schools and similar institutions (Chapters 9, 10 and 11). So far as area (b) is concerned, the coverage is adequate, in view of the fact that educational problems are widely discussed and commonly encountered. Area (a), however, is treated, almost exclusively, from a limited point of view, namely, the development and application of cybernetic/system-theoretic experimental methods and means for measurement or regulation.

That is certainly not all of the story. The precedent for a cybernetic/system-theoretic approach to experimental psychology was set by Craik, more than 30 years ago; mostly in area (a). The concepts and achievements are vividly portrayed in his collected essays (Craik 1966; Ed, Sherwood). Since those days, cybernetic disciplines, notably information theory, have been applied very extensively. No attempt is made to cover the ground staked out by area (a) because two authoritative books already do so; namely, Broadbent's *Decision and Stress* (Academic Press, 1971) and Welford's *Fundamentals of Skill* (Methuen, 1968). The scholar should consult both of them; they fill the gap already noted and also give another, entirely compatible, perspective upon the whole field. Both books use cybernetic (information theoretic) techniques to explain particular psychological phenomena (vigilance, the division of attention, memory and recall). In contrast, this book insofar as there is any overlap, is written from a cybernetic but global point of view; for example, to pick out generally useful experimental or regulatory devices. Moreover, both Broadbent and Welford deal (in an instructively different manner) with the important issue of 'signal in noise' measurement and 'receiver operating curves', which are merely mentioned in the present volume.

Let us turn to cybernetics/system-theory itself, rather than the cybernetics/system-theory of human beings.

Amongst other changes it is fair to say that the original definition of what cybernetics is about has, to some extent, been modified. Weiner's definition, 'control and communication in the animal and the machine', still stands, but only provided that communication is given a broad enough connotation (roughly the one he gave) and that the notion of 'machine' is generalised to cover some quite unusual types of processor that bear little direct relation to general purpose computers as commonly used at the moment.

Taking a broader view, the emphasis has shifted in several ways, as follows.

1 In dealing with *systems* of any kind, cybernetics is primarily concerned with establishing isomorphisms (one to one correspondences) rather than the validation of propositions that are true (or have a chance of being true) or else are false. The basic mode of argument and development involves analogy. Strict analogies of which isomorphism is a special case. The analogy expressed or represented in the language employed to account for events is a *metaphor*. In this sense, cybernetics is the science or the art of manipulating defensible metaphors; showing how they may be constructed and what can be inferred as a result of their existence.

2 Isomorphisms, and material analogies of a less complete variety, are ubiquitous in physical science and technology and are essential to its progress; for example, the isomorphism between an operating electrical circuit and a hydraulic system, and between a plan (for a building) and the various artifacts that might be born of the plan. Further, apart from the operational use of analogy, which everyone relies upon in practice, arguments by analogy mark major innovations (and minor innovations as well for that matter). Polya's (1966) comments on this issue are particularly illuminating; readers who prefer to think in broader terms rather than the esoteric language of science may like to note how often Schneidmann's 'modes of concludifying' constitute analogical arguments.

It is true, but less commonly recognised that correspondences of this type and the arguments based upon them lie at the root of reflective and relativistic schemes, i.e. they are bound up with the activity of a *participant* observer rather than a classical external observer. These notions are developed at some length. We also have occasion to draw strict analogies, for example, between perceptual motor and cognitive skills, valid under a particular class of model. But the full gist of the discussion is exhibited in the next volume, where it is possible to detail the *participant* languages in which analogies of one kind or another play much the same role as propositions in a conventional language.

3 Cybernetics approaches a problem from the general to the particular rather than vice versa. The distinction is especially obtrusive in the context of computation where (traditionally) a large program is built up, by the programmer, from unitary instructions. In contrast, programming languages such as PLANNER or MICROPLANNER take basically global statements (of the kind 'achieve a goal by any method at the machine's disposal') that are qualified in so far as certain operations or methods are to be excluded. This orientation will be evident throughout the book, without specific mention, and is often contrary to the tenets of reductionism; though it is not in the least measure 'sloppy'.

4 In this connection the word 'goal' has undergone very appreciable revision. It used to be popular to think of *goals* as goal *states*. Nowadays, at least in the context of cybernetics, the goal is interpreted as a class of intentions or processes. Under this caveat it is possible, without difficulty, to posit *underspecified* goals (Pask, 1970a). Since underspecified goals are in the majority, *purpose* no longer has the singlemindedness of the prophet Job, so ably criticised by Bateson (1972) in *Our Own Metaphor* or Brodey in numerous publications.

5 Perhaps the most important mathematical tool is the theory of self reproducing machines. Burkes (1969) argues, for example, that whereas Wiener emphasised feedback regulation, due to his preference for a certain brand of mathematics, Von Neumann emphasised the theory of reproductive automata (noting that any such machine contains masses of feedback loops). So, for example, the idea of stability can be represented as a reproductive act that maintains a stable entity or, as an alternative, the entity can be regarded as stabilised by certain kinds of feedback. The approaches, in other words, are not incompatible, by virtue of which Wiener's (1954) theory of self reproduction is itself expressed in these terms.

6 The journey takes us into the territory of artificial intelligence but scant justice is done to the elegant concepts developed in this field. Other people's programs are often far more intriguing than our own; especially if the reader likes to view the intelligent operations in a restricted field, such as scene recognition. Hence, this book can be usefully augmented by further reading and the journal *Artificial Intelligence* and the series *Machine Intelligence* are particularly germane. As an overall dogma I see no need to use the qualifier 'artificial'. If a system is intelligent (more interestingly, if it evidences the exercise of intellect) it may be biological or not. Biological computations, executed in the brain as processor, constitute very special and interesting cases of intellect pure and simple.

Many other points can be made and come out in context. For example, there is increasing emphasis on the logic of distinctions (Spenser Brown, 1971), temporal logics as conceived by Gabbay (1972) and Rescher and Urquhart (1971), and the logic of norms (Von Wright, 1966), commands and questions. These and other developments underlie the entire argument. They are mentioned but not fully exhibited until the next volume.

7 Two of the original concepts of general system theory have recently been rescusitated (there is an admirable discussion in Ackoff (1973). The first of the two concepts is that whereas causal argument considers necessary and sufficient (or double implication) conditions, system theory/cybernetic argument deals primarily with a logic of necessary conditions (Singer's producer/product relations; alias the goal relations in Somerhoff, 1969). This notion underlies the more sophisticated treatment of *goals* and *intentions*; conceivably, partly specified intentions.

The other concept, of about the same importance, is the presupposition of a systematic universe. There is a tacit assumption that things, objects, and other elementary entities are interdependent (rather than being isolated units, which is the assumption behind the majority of sciences). Further as a result of their interdependence (equisignificantly, of the supposition that things, objects, and so on are not really unitary) these entities form *systems* and it is systems which may be observed and manipulated.

8 In his otherwise excellent book *Symbols, Signals and Noise* Pierce (1962) makes acid, even pejorative, comments upon the status of cybernetics. There is more than a grain of truth in his remarks. Much of the literature either consists of original piffle or diluted recapitulations of original work by commentators who are usually unaware, directly, of what was done.

This book is open to much the same criticism. It may be trite and trivial; surely it is designed to minimise technical mathematics and (so far as possible) equations etc. are confined to the appendices. Obviously I do not think it is trite and trivial (the reader must judge for himself) and it is, at any rate, original (not an unmixed blessing because the discussion is specific and patchy by virtue of it). However, I am thwarted by lack of space, and have made liberal use of reference to a further volume which contains the more recent and exciting developments for which this material is an essential foundation. This expedient is justified by the fact that the work in question is either in progress or has been fully reported in other monographs and papers. All the same I find the journey, even up to the point reached in this volume, quite enthralling and hope the reader will share that fascination.

Gordon Pask
Richmond 1973

1 Information

'Information', used in the technical sense, is the converse of an uncertainty (an uncertainty decrease is an increase in a state of information). Appropriate measures of information/uncertainty are designated H (equation 1 in Appendix A) and computed in respect of one universe which is believed to be independent of others. In order to carry out the computation the 'universe' in question must be partitioned, by a description, into a set of exclusive and (for the universe) exhaustive alternatives, one of which must obtain and only one of which may obtain.

For example, a universe U may be a choice set, the (set of) values of a state variable or a set of states, determined by the conjoint values of several state variables.[1]

Two universes, U_α and U_β are distinct if they are constructed to consist in exclusive and exhaustive alternatives or if they are consistently viewed in this fashion. Thus, U_α and U_β might represent the 'state sets' of two 'machines' or the alternatives in two lotteries. If they are thus distinguished, U_α

1. This terminology is consistently employed but a confusion is possible. With equal rectitude 'universe' may mean a 'universe of discourse'; i.e. a set of static objects, $s \in S$, pointed out in an observers (formal) language L. To reconcile these ideas consider m tests an observer can perform on the objects in S. These are property tests (for properties E, F . . . described in L). Elementary test statements are such basic utterances as 's has property F' (equivalently 's belongs to a subset $F \subset S$') or 's does not have property F' (equivalently 's belongs to the complement of a set $F \subset S$') and the strongest L statements are conjunctions like

(s has E) *and* (s does not have F) *and* . . . so on

extended over all m properties. These *strongest* L statements are exclusive (only *one* can be true and *one must* be at any instant). They, or any other alternative sets formulated in L, describe the occurrences or options, u, that belong to the present universe U.

This U_α and U_β may refer to different sets of objects (S_1, S_2), to distinct descriptions of sets that have objects in common, or even to distinct descriptions of the same set. Fundamentally, U is formulated in terms of tests that are made (of E's value, of F's value) or events that take place. U_α and U_β are formulated in terms of supposedly independent tests or events.

and U_β may coexist as simultaneously observable, being indexed (as α and β) by a sampling operation.

Using the notation $U(t)$ to represent a universe at time t (with t = time measured in observation intervals 1, 2, and so on) it is possible to establish the special equivalences ($\equiv$) of $U_\alpha(t) \equiv U_\alpha(t+1)$. . . or $U_\beta(t) \equiv U_\beta(t+1)$ . . . and thus to count event occurrences in 'the same' universe U_α and 'the different' universe U_β. But the usual rider (in an alternative set) that 'one and only one of the possible events occurs *at once*' provides an essential clue.

Time is another way of indexing universes; in general, $U(t) \neq U(t+1)$. So, for example, it is quite possible, and in fact quite usual, to set $U(t) = U_\alpha$ and $U(t+1) = U_\beta$. Conversely, we may set the (Cartesian product) $U_\alpha(t) \times U_\beta(t) = U(t)$ and $U_\alpha(t+\Delta t) \times U_\beta(t+\Delta t) = U(t+\Delta t)$ (with Δt enough time for event observation) so as to observe or talk about some configurations of (joint) events in $U(t)$ in relation to an independently sampled configuration of (joint) events in $U(t+\Delta t)$. This point is quite crucial in establishing the notion of a *frequency* of event occurrence: that is, a count of independently sampled events in *one* universe which, for this purpose, is regarded as a series of (equivalent) universes such as U_α defined as $U_\alpha(t) \equiv U_\alpha(t+1)$ . . . and so on.

To each alternative in a universe must be assigned a positive fractional number p (say p_i to the ith alternative) with the property that the p_i summed over all the values of the index i, equals unity. The interpretation of the p_i, generally called 'probabilities', determines amongst other things who or what experiences the uncertainty or receives the information. It does not bear upon the manipulative calculus.

1 Multivariate Selective Information Indices

Suppose there are two or more universes U_α, U_β, which *may* be independent. In other words, it is possible to entertain an hypothesis to this effect and to *disconfirm* it or affirm it tentatively. If so, it is possible to compute uncertainties/informations $H(\alpha)$ and $H(\beta)$. Further, by examining joint contingencies (alternatives indexed x in α and indexed y in β) it is possible to compute a joint uncertainty $H(\alpha,\beta)$. This is shown in equation 2 in appendix A. The quantity

$$T(\alpha,\beta) = H(\alpha) + H(\beta) - H(\alpha,\beta)$$

is the *transmission*, or coupling, between contingencies in these supposedly distinct, and surely distinguishable, universes. If $T(\alpha,\beta) = 0$, the universes are (tentatively) regarded as *independent*: if $T(\alpha,\beta) > 0$, they are *dependent*.

For example, the alternatives x in α may be inputs to a 'Black Box' in Ashby's sense (α is a universe of inputs or a set of *input states*) while the

alternatives y in β may be outputs (β is a universe of outputs or a set of output states). As another example, α is any set of alternative messages labelled by values of x proper to one man or machine, the transmitter perhaps, and β is any distinct set of alternative messages labelled by values of y proper to another man or machine, the receiver. Moreover, the universes may represent the same (or different) systems at different times ($U_\alpha = U(t)$, $U_\beta = U(t+1)$) when the transmission or coupling measured by $T(\alpha,\beta)$ is, in a special sense developed in Ashby (1970), the amount of 'memorisation work' needed to establish this much coupling. In a final example, again due to Ashby (1970), α is the universe of prescriptive actions available to a designer who can select (in designing a plant or process) from a family, F, of functions, $f_x \in F$, specifying operations to be performed and β is the universe of 'goal directed behaviours', y, of the plant or process as governed by its design. The example is especially interesting insofar as the degree of control which can be exerted (by the designer in this example) is known to be limited by the informational coupling. Conant's theorem (Conant and Ashby, 1970). The case cited above, is usefully compared with the case in which a designer can only operate upon the external input and output of the plant or process (i.e. upon its behaviour *per se*). The saving due to design of the first kind is gigantic.

For more than two universes it is possible to compute further coupling terms; residuals or interactions, Q (equation 3 in appendix A), and thus to derive a general caculus of multivariate information analysis (the deterministic analogue for which is constraint analysis (Ashby (1964a)). The canonical inequalities of multivariate information analysis are stated as tersely as possible in Ashby (1969).

One extremely useful index is the redundancy, Z, of a given universe (equation 4 in appendix A) which is a measure of constraint, either deterministic or probabilistic; *i.e.* the extent to which the universe is coherent or organised. As noted in the Appendix, there are many plausible definitions of redundancy, but all of them serve as indices of uncertainty and are functions of at least two variables. One indexes the amount of variation that might occur within the stipulated constraints (if any) upon the universe, the other indexes the variation that is manifest. In other words, the redundancy is a measure of the transmission between a specification of *one* universe and the events occurring within it.

If it happens that the p numbers can change over time (for instance, the 'time and memorisation' example given above) and it also happens that the *universe* can change over time (either because *it* or its irreducible constraints are modified) then there may be a non-trivial form of self organisation (equation 5 in appendix A).

Call any or all of the quantities H, T, Q, Z or others derived in a similar manner 'uncertainty/information indices'. In general, they are *selective* information indices since information is gained by selecting amongst a set of alternatives. The interpretation of these indices depends upon the genesis of the alternative sets in the universe(s) concerned and upon the origin of the p numbers.

2 Statistical Indices

An external (impartial, unbiassed) observer may overlook a system which in the last resort *he* has defined; for example, the input set and output set of an organism; stimuli it receives and responses it emits. Further, he may compute the p numbers in any systematic manner. He could, for example, estimate them by guessing (if he were a Bayesian statistician he could do so rationally) or, more often, he subscribes to the common statistical dictum that a probability p_i (that x has the value x_i) is obtained, as the limit, for infinite sequence length of the ratio

$$\frac{\text{Number of occurrences of } x = x_i}{\text{Number of occurrences of any value of } x}$$

The formulation depends upon 'stationarity' of the source of occurrences (a notion considered further in Chapter 2). Roughly, 'stationarity' means that the underlying structure of the source is unchanging.

Under this interpretation, the uncertainty/information index is a *statistical* uncertainty/information measure. Given either method of computing the p numbers, any uncertainty information indices reflect *his* uncertainty/information about the system. For example, subscribing to the last dictum, the external observer may record event occurrences in a contingency table to show the frequency with which events tagged by values of x (input states) and y (output states) occur, including the events due to joint occurrences. The entries in the contingency table may now be taken as estimates of the actual probabilities (a good example is given in McGill (1963) who also provides the mathematical apparatus needed to compute in terms of count numbers without transforming them into p numbers). Some instances of this type of calculation are cited in Chapter 7.

Next, an external observer may presuppose that some other observer (a participant, like an operative in industry, or an experimental subject) sees an *environment* in the same way that *he* sees it. If so, he may infer that the *statistical* uncertainty/information indices computed with reference to *this* environment provide estimates of its variability, complexity, regularity, or whatever (depending upon the chosen index), from the participant's point of view. Some examples of large-organisation task-analysis along these lines are shown in Table 1 and Fig. 1. Similar studies have been carried out

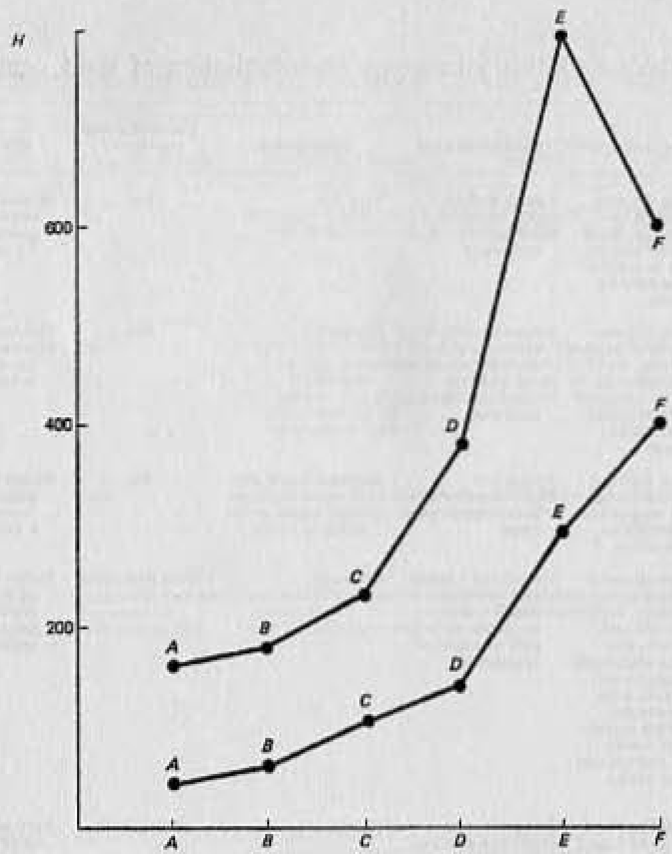


Figure 1 Information analysis for six industrial/commercial tasks *A, B, C, D, E, F.* Upper curve (appropriate to novice) is gross uncertainty. Lower curve (appropriate to a skilled operative) is uncertainty, given that all constraints are known. Difference (a transmission) is an index of the amount to be gained by experience. The lower curve is an index of relative difficulty of tasks for any (idealised) proficient operative.

in the timber industry by Van Gich (1971) who relates the indices to ergonomic and work study measurements. Stanisland (1966) has used similar techniques in connection with laboratory tasks and has devised a plethora of redundancy measures for this purpose.

3 Selective Operations and Selective Work

Suppose that α and β are the input and output sets of an organism, perhaps a human being (as an expository convenience; the argument is quite

Table 1 Ordinary language characterisation of tasks compared in terms of information indices in Fig. 1 (study by: Scott, Elstobff and Pask).

Job label	*Brief description*	*Outline structure*	*Interruptions*	*Communication important?*	*Effect of errors*	*Repetition*	*Decision making*	*Input to the process*	*Special knowledge required*	*Specialised vocabulary used*
A	Checking the correctness of invoices, etc., from suppliers and invoices as correct for payment by treasurer	3-stage process Single operator Batch processing of each stage	Very few	No	Moderately serious since operative is passing invoices for payment	Highly repetitive	No	All input available at start of process	Little special knowledge required	Small specialised vocabulary used About 15 terms
B	Collation and preparation of batches of invoices for processing prior to input to computer Checking of copied invoice against originals	3-stage process, each operative able to carry out all stages Batch working Parallel working at each stage	Infrequent	No	Not very serious Most errors picked up at later stage in processing	Highly repetitive routine work	No	Input to the process is complete at the start of the process No information gathering or waiting for information required	Very little special knowledge required	Quite small specialised vocabulary About 25 terms
C	Updating staff list with information about resignations, new employees and transfers	2-stage job Single operator batch processing of each stage	Frequent (clerk also operates telephone switchboard whilst doing this job)	No	Errors important since this job involves updating a master record	Highly repetitive	Some	All input available at start of process	Little special knowledge required	Small specialised vocabulary used About 10 terms
D	Determining customer's requirements for service, provision removal, cessation, etc. Checking equipment and plant available, arranging appointments, obtaining acceptance of conditions, charges and issuing advice notes	Average of 5 stages Operators work in parallel, each responsible for a part or whole of organisation	Frequent	Very important	Errors fairly serious as work of other divisions depends on operator's output	Varies as to areas served Many cases similar, others unusual or infrequent	Difficulties may be referred to superintendent	Information gathering is necessary Waiting often required	Considerable special knowledge required	Large specialised vocabulary used About 200 terms
E	Preparation of legally required load documents for a craft, using information on load plan	4-stage process Operators working in parallel each operator responsible for one flight at a time	Occasional	Occasionally	Very serious: safety laws demand accuracy	Repetitive process largely routine	No	No information gathering necessary, but waiting for required information is necessary	Considerable special knowledge required	Large specialised vocabulary used About 150 terms
	Assembly of data and preparation of a loading plan	6-stage process A separate operator carrying out each stage Sequential processing, i.e. each stage being fed by its predecessor Concurrent processing of load plans within some stages	Frequent	Yes	Very serious: safety laws demand accuracy	Repetitive process but not routine The load planning of each flight is a separate problem	Yes	Information gathering necessary Waiting for required information necessary	Considerable special knowledge required	Large specialised vocabulary used About 150 terms

general). Let x (indexing elements of α) be determined by an external agency and consider the value of $H(\beta)$, the output uncertainty. If there are just M values of y (indexing the M elements of β) then surely $\log M \gtrsim H(\beta)$ with equality if the alternatives are equiprobable. Now envisage the organism carrying out a selective operation in the following sense; it is designed, or has evolved, or is told that, for each value of x, there is a correct value $y = R(x)$ of y and it has the 'goal' of selecting this value of y for each x. The selective operation it performs entails so much selective work and if we hypothesise that a general dichotomising process takes place which occupies a unit interval, then the larger M is the larger will be the interval (the reaction time or the latency) for any correct response.

Hick's (1952) law asserted such a correspondence which was later established by Crossman (1953) and others; that is, if the correct reaction time or latency is written RT then

$$RT = \text{constant} \times \log M$$

In fact, by using frequency-weighted selection categories, it is possible, for some tasks, to assert that

$$RT = \text{constant} \times H(\beta)$$

or (as appropriate, on varying R)

$$RT = \text{constant} \times T(\alpha,\beta)$$

though the results (obtained, for example, by Fitts (1954); Bricker (1955)) are sometimes enigmatic. The deviations from Hick's law are primarily due to the fact that (apart from the type of experiment Hick employed in his empirical studies) the external observer's computation does not tally with that of the subject or organism. However, it is still true that when $H(\beta)$ is less than M because the selection probabilities differ, the correct selection can be expressed as a selection from a number, EM, of (imaginary) equiprobable alternatives equivalent to the actual set of biassed alternatives, and the selective work is measured by EM; that is

$$RT = \text{constant} \times \log EM = \text{selective work}$$

The notion is not trivial. The selective operation is a *non-specific computation*. It (and information theory as a rule) makes no comment upon *how* a computation is performed; given x there are indefinitely many ways of computing the required value of y. The index of selective work is an average taken without discrimination over all of these possibilities, and selective work, indexed by RT, may be taken as a gross index of the uncertainty that

must be resolved by a subject regardless of whether or not the external observer's view of the environment is in agreement with the subject's view.

Similar, but differently phrased, arguments underlie the excellent work of Tanner and Swets (1954) on signal detection in a 'noise' background.

The selective work index is used implicitly in expressing latency-weighted measures of task performance in terms of information measures, in Chapter 7 and Chapter 8 especially.

4 Capacity Limits

One of the main results in the commonest application of statistical and selective information theory (namely, systems composed of a receiver and transmitter coupled by a channel which is imaged in the abstract by a transmission) is the limiting channel capacity theorem (Shannon and Weaver, 1949). It is assumed that perfect reception of a message is hampered by the introduction of 'noise' irrelevant to the transmission; it is also assumed that the receiver and transmitter have statistically stationary properties so that conditional and joint probabilities can be estimated by ratios formed on averaging over long sequences of inputs and/or outputs.

The capacity theorem says that it is possible to encode messages, transmitted along the channel (i.e. to impose a structure upon the transmission which relates values or sequences of values of one variable (x) to the values that are assumed by another variable (y)) in order to secure nearly error free transmission as the rate of transmission is increased (Fig. 2); beyond that limit (the 'capacity' of Fig. 2) distortion is increased by any increment in rate.

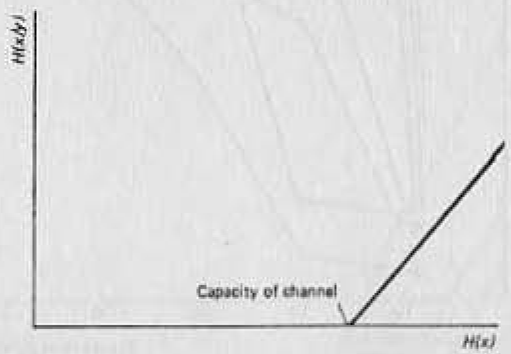


Figure 2 By suitable encoding the channel may transmit without distortion (or error) for any rate up to the capacity but, beyond that, the minimal error increases with increasing rate. $H(x/y)$: uncertainty about value of input x if value output y is given, $H(x)$: input information rate or uncertainty about value of x.

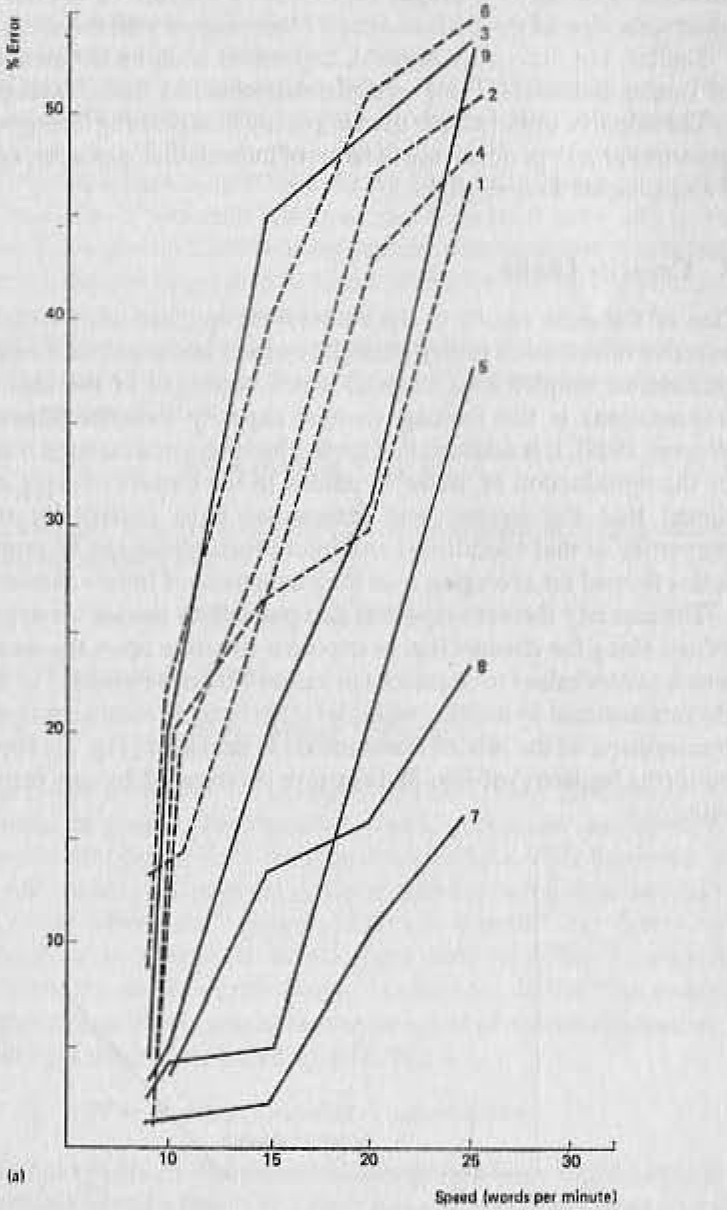
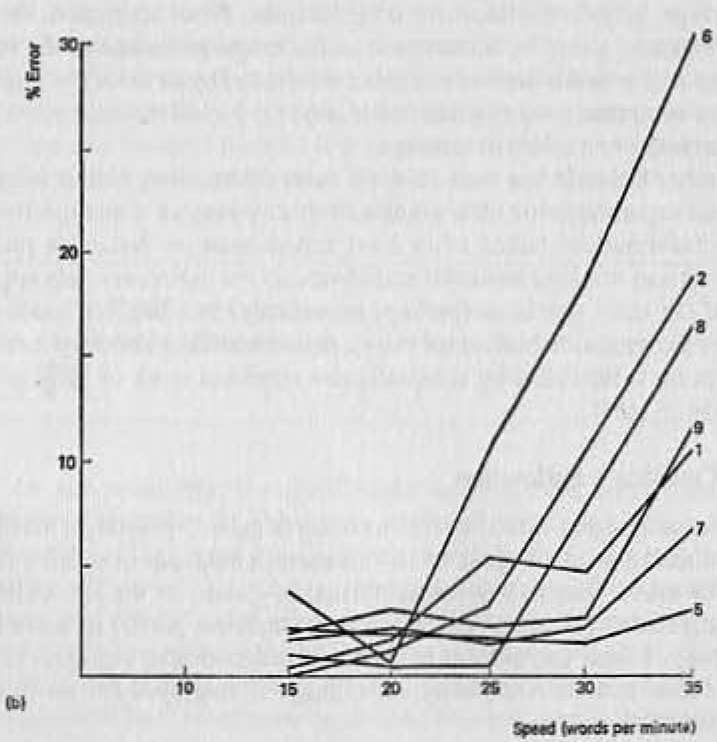


Figure 3 Pace error curves: (*a*) at an early stage of training; (*b*) at a later stage of training.

5 Human Capacity Limits

Some human experiments or task situations can be associated with plausible criteria of relevance and it is also approximately true that the human being has stationary characteristics (that is, a fixed processing structure underlying statistical variations). This is true, for example, of many psychophysical experiments where the human being is regarded as an essentially physiological device which can be imaged as a noisy channel that couples an input (x) to an output (y). When the human being learns and when he performs symbolic transformations it is neither possible to specify relevance (in an adequate sense) nor to assume stationarity. Hence, under these circumstances a human being does not have a channel capacity in the strict sense. But it is often true that Hick's principle (in precis, the 'rate of gain of information is constant') determines a *loading* which is analagous to a channel capacity. The loading limit is particularly evident in the context of paced-input skills, like teleprinting in class with key-depressions synchronised by a metronome beat. The absolute degree of selection depends upon the trainees level of competence and (generally) increases as he learns. But, for all trainees who do learn, there is a marked inflexion in the error/pace



graph. Beyond a certain fixed pace, any trainee commits errors. If he is able to compensate for errors up to a limit at which breakdown occurs (by analogy only, by effective 'encoding'), there is a discontinuity as shown in Fig. 3a (by analogy only, the 'capacity'). In contrast, some trainees commit errors at any pace of operation and there is no real limit. These trainees also fail to learn the repetitive skill (Fig. 3b shows error pace curves for the same subjects, after a further 18 hours of training).

6 Logical Uncertainty/Information Indices

The general (method-of-computation independent) index 'selective work' is usefully contrasted with an information index which is, in one sense, its polar opposite—an information measure based upon one class of possible computing methods. Measures of this kind are called logical information measures and are specified in respect of a formal language which is very restricted; the class of computations being the class of derivations obtainable by using the inference and production rules in this language. These measures were devised by Carnap (1950), Carnap and Bar-Hillel (1953) and Bar-Hillel (1964). Some important generalisations are due to Chiaravigillio (1971), and Harrah (1966). The strongest statements in such a language, together with their complements, form alternative sets (the 'information given' by a statement in the language measures the number of logically possible derivations that are excluded by virtue of the statement) but, *if* computed over alternative sets *only*, the logical measure corresponds numerically to a selective measure.

Rather little use has been made of these information indices in psychological experiments or task situations, at any rate at a quantitative level (the 'information' talked of in most experiments on deductive problem-solving is of this *kind* however) and obviously the indices are only appropriate if the mind functions (perhaps imperfectly) in a 'logical' mode (*i.e.* it obeys the prescribed inference rules). An outstanding and very interesting exception is furnished by the *qualitative* empirical work of Seigmann and Stapleton (1971).

7 Confidence Estimation

Consider an agent who is a sentient being (a man or possibly a machine or an animal) able to interpret and comprehend a language in which questions can be asked. The very restricted formal languages of the last section can be augmented (as in the Seigmann and Stapleton paper) to serve in this capacity. I shall use natural language in order to give examples; in fact, some intermediate complexity of language is employed for much of our current work.

The agent is asked a multiple-choice type of question. Syntactically, it is an exclusive disjunction of statements. But these statements have a semantic interpretation whereby the agent refers them to a body of internal or external data and the question is prefaced, when it is addressed, by the *requirement to satisfy* a relation (which is one meaning of the many-faceted and often misused word 'goal'). Succinctly, a question of this type poses a problem; to find (say) the one and only one alternative answer, satisfying the relation 'bridge-building engineer', within an interval, Δt (which may be indefinite), given a semantic interpretation of the language in respect of industrial history and the following alternative statements:

(a) 'Brunel built the Bristol Suspension Bridge',
(b) 'Stephenson built the Bristol Suspension Bridge',
(c) 'Wakefield built the Bristol Suspension Bridge'.

The semantic interpretation admits, for example, such factual data as: 'the Bristol (Road) suspension bridge was built by an engineer who built railways' and 'Wakefield built no railways' and 'Stephenson did not design Temple Meads Station at Bristol'; such inferences as 'Wakefield did not build the bridge' and 'if it is true that the bridge was built by an engineer who built a railway to Bristol then it is not false that Brunel built the suspension bridge'. The pragmatic interpretation has an imperative/interrogative part (a question of this kind is isomorphic with a command to answer) and it specifies the relation deemed 'correct' (namely between *road* bridges and *builder's names*). It is improper to gloss the issue by conceiving the question as the stimulus for a correct response, even though this identification is suggested by the usual *format*, namely:

Who built the Bristol Suspension Bridge: (a), (b), or (c)? (one and only one is right)	(a) Brunel (b) Stephenson (c) Wakefield

7.1 As one possibility, the agent might *derive* a statement to resolve the exclusive disjunction (a), (b), (c), by computation in the language (the object language of the scheme), using inferences of the kind suggested above and starting off from the existing and comprehended data. As a comment, if he did so and merely ticked or marked the alternative corresponding to his derived (alias computed) object language statement, then this kind of enquiry gives little of the potentially available information. The agent might, for example, have been asked how he arrived at his solution to the problem.

However, this possibility depends upon the existence (in the agent) of a computing method and upon there being enough time to execute it; that is, $\Delta t \gtrsim$ object language computing time.

7.2 If the agent had *no* method, then he could not compute an answer. Since one answer must be given he may either construct a method or guess an answer within Δt. The former activity is learning and the latter is a minimal instance of choice or decision.

7.3 If a computing method is available to the agent, without the necessity of constructing one by learning, then its application still depends upon a sufficiently lengthy Δt (*how* long is determined by the method, the agent's state of knowledge and so on). But even if Δt is too short, the agent may carry out a partial computation that eliminates some of the alternative statements; as a result he may be able to guess between fewer alternatives and thus to make a more discriminating choice. (Parenthetically, 'cueing information', delivered in the course of computation, is information in the logical sense; cueing information that excludes the *names* (a,b,c,) of the alternatives is selective.)

7.4 Suppose there is a *metalanguage* (able to accommodate descriptions of classes of object language statements, inference rules, etc.). In particular, suppose the metalanguage is a statistical metalanguage (formal or not) of which the formal grammar consists in rules of statistical inference; for example Baye's rule. (This example is due to Phillips, 1970.)

If so, the agent is in a position to compute weights or likelihoods attached to the alternatives, and to bias any otherwise random guess (in the limit by giving one or more alternatives a weight of zero but, in any case, refining his choice or decision).

7.5 The data treated 'statistically' (either by formal precept or rules that are not fully axiomatised) may be derived from partial computations, relevant to the questioned relation, that are carried out in the object language.

7.6 The data might, however, be quite different; for example, the frequency with which alternative statements that happen to be labelled '*a*' have been deemed correct at previous trials.

7.7 There is also a special type of distortion, noted by Shuford (1965), Shuford, Massengill, and Albert (1966) and Finetti (1962) which is obtrusive if correctness is assigned a score value, if the answers to a test sequence of questions are scored cumulatively by correctness, and if the agent aims to

maximise his score. Acting as a good statistician (at any rate) he will appreciate that the mathematical expectation of score is maximised by a deterministic choice of the most likely alternative and not by a biassed guess between alternatives weighted to represent a current state of knowledge; from an observer's point of view this clever calculation is *irrelevant* to the agent's state of knowledge. (See Appendix B.)

7.8 With 7.5, 7.6 and 7.7 in mind, it is possible to ask the agent for a confidence estimate over the alternatives instead of a selection. This is the result of a metalinguistic computation, constrained by the metalinguistic syntax to yield as many positive or zero fractional numbers as there are alternatives, with the property that all of them sum to unity, i.e. these numbers have the form of probability numbers, p_i, attached to the members of an alternative set (in the example $i = a$ or b or c). In other words, the confidence estimate is the output of a process that would, if selection were demanded, bias the otherwise random throw of a 'mental dice' with as many sides as there are alternative statements

7.9 Under certain circumstances, confidence estimation is impossible or uninformative. Presumably a metalinguistic computation occupies a certain interval of time; thus the estimate cannot be elicited unless $\Delta t \gtrsim$ metalinguistic computation time. Again, if the object language computation time is shorter then the metalinguistic computation time (and an object language computation or derivation method exists) the appearance of confidence equal to unity in one alternative (by the imposed syntax, the other p_i being zero) is due, in fact, to the object language derivation of a statement. If Δt is too short either for object language computation or metalinguistic computation, then the agent is bound to guess in any case. But, in general, the confidence estimate gives more information to the experimenter or external observer than a direct selection and, insofar as the pathologies noted in sections 7.6 and 7.7 are avoided, this is relevant information.

7.10 The confidence estimate *itself* is interpretable as the agent's 'degree of belief' in each of the alternatives and is usually elicited by a request to express degrees of belief in terms of a betting or wagering distribution. Under this interpretation an uncertainty information index, calculated over the alternative set, using the degrees of belief p_i, is an index of subjective or 'to the agent' or 'as seen by the agent' uncertainty/information. All of the uncertainty/information indices used in Chapter 11, section 3 onwards, are subjective indices.

The argument of this section is summarised in Fig. 4.

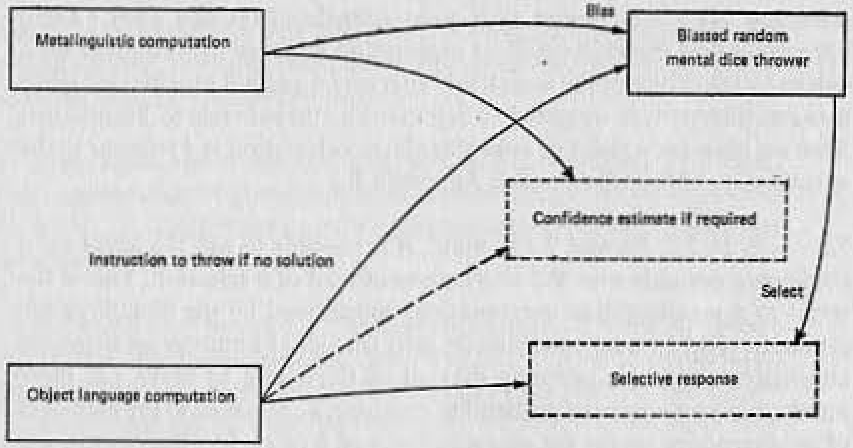


Figure 4 Summary of discussion of confidence estimation (see text). Dotted line represents exclusion of alternative statements by deletion buttons on BOSS.

8 Methods

Various precautions are taken to ensure that data of this kind, indicating subjective uncertainties/informations, is not spurious.

8.1 Concentrating upon the pathology of section 7.7, Shuford and his colleagues have devised schemes that defeat the 'good statistician' trick. Two are outlined in appendix B (for the minimal case of two alternative statements).

8.2 Clearly, an agent can be certain and correct or certain and mistaken; in doubt and favouring the correct alternative or in doubt and favouring a mistaken alternative. Only if he assigns equal probability to each alternative is the issue of correctness immaterial.

8.3 Hence, indices of subjective uncertainty/belief are augmented by an index of correct belief calculated from the same data. A Shuford score (the simplest of them all) is used for this purpose:

$$\text{correct belief} = \begin{cases} 1 + \log p_i \text{ for } p_i \gtrsim 0{\cdot}1 \\ -1 \qquad \text{for } 0{\cdot}1 < p_i \end{cases} \quad \left(\begin{matrix}\text{if the } \textit{correct} \\ \text{alternative is} \\ \text{the } i\text{th}\end{matrix}\right)$$

8.4 It is possible to guard against irrelevant estimates of section 7.7 in several ways; one of them is the Shuford method since, in fact, the statistician's maximising strategy is a valid but irrelevant metalinguistic computation and the act of thwarting it discourages irrelevant computation in general. Other techniques are noted below.

8.5 The burden of satisfying the metalinguistic 'syntax' (that any well formed statement for M alternatives is a set of M numbers p_i such that $1 \gtrsim p_i \gtrsim 0$ and $\sum p_i = 1$) may be appreciable. To relieve it, the syntactic constraints are satisfied externally by a normalising equipment which only permits 'well-formed' statements. Baker (1969) uses a histogram; one bar to each alternative, bar heights representing degrees of belief. It is displayed on a computer controlled cathode-ray tube. The heights of the histogram bars are adjusted at will, one at once, by the agent; but the bar heights are automatically normalised so that they sum to unity. A very simple equipment will suffice for many purposes. The p_i are displayed on meters wired in series with variable resistances and fed from a constant current source; the agent adjusts variable resistances (one to each meter) to change his asserted degree of belief.

8.6 The crucial requirement is to reduce the metalinguistic processing time since (from section 7.9) this favours the various sampling conditions that are propitious from the external observer's point of view.

Provision of an external normalising circuit is a step in the right direction. Some improvement is effected if the equipment is automatically reset for each questioning trial. It is still better to complete the externalisation by providing a feedback signal indicating the degree of freedom used up (or the constraint imposed by the agents previous adjustments) in order to arrive at his confidence estimate.

Since the external observer does not, and cannot in this format, discriminate between *relevant* metalinguistic computations and object language computations, it is important to allow for the rapid exclusion of alternatives as a result of object language computations that are partially carried out. All of these specific expedients also act as devices that encourage the agent to produce p_i values by relevant computation. They do not ensure that he will do so; that depends upon the normative scheme in which the agent accepts the goal of treating questions as problems that are solved by satisfying a goal relation.

8.7 The sampling equipment (Belief and Opinion Sampling System, BOSS) used at present in my own laboratory is shown in Plate 1. Questions are inscribed on cards bearing code holes for various data, including a designated correct alternative. Insertion of the card resets the circuitry and activates those out of the alternatives (maximum of 8) that are in use at a given trial. Degrees of belief are displayed on the meters which are initially assigned equal readings. The agent adjusts this profile of readings by pushing levers to increase or decrease meter readings corresponding to each of the active alternatives. Members of the active set may be excluded by pressing deletion buttons. The profile is automatically normalised and the 'existing constraint' feedback is simulated by a variable lag or delay in the meter response. When the agent is satisfied with the profile he submits it for evaluation by the sampling system; if appropriate, he receives knowledge of results, and is permitted to withdraw the question card and insert another one.

8.8 Two further points should be stressed.

(a) The confidence estimation paradigm does not demand an external criterion of correctness; it is only necessary that one alternative be selected. So, for example, the agent may be asked to express his belief that he will purchase one and only one of several alternative products, in which case the only correctness criterion is internal, i.e. his personal preference. To distinguish trials for which there is *no* external correctness criterion, they are called option trials and the profile of readings is referred to as an 'opinion' statement rather than a statement of *belief*.

(b) The paradigm does not demand that the alternatives (provided they *are* exclusive and exhaustive) have been chosen by an experimenter or an external observer. Unless inter-agent comparisons are essential it is convenient to allow the agent to select his own sets of alternatives quite freely (he inscribes them on blank question cards and inserts them himself). Often, also, it is convenient actively to solicit the agent to specify his own alternative statements (for example, in studies of strategic uncertainty, noted in Chapter 11) and several variants on the repertory grid technique (Kelly, 1955) have been used for this purpose.

9 Experimental Examples

The following instances illustrate the use of uncertainty/information indices based on confidence estimation (interpreted as indices of subjective uncertainty/information).

9.1 Figure 5 shows an information analysis of a clerical operation. Given an invoice (and invoices are processed in lengthy batches) the clerk must find and record the code numbers of the several items referred to. The item codes are specified in a code book in which items are grouped under categories and, from this book, it is possible to compute the actual dependency between the relevant variables (supplier's name, origin name, area, item group, item, and so on). However, it would be impracticably tedious

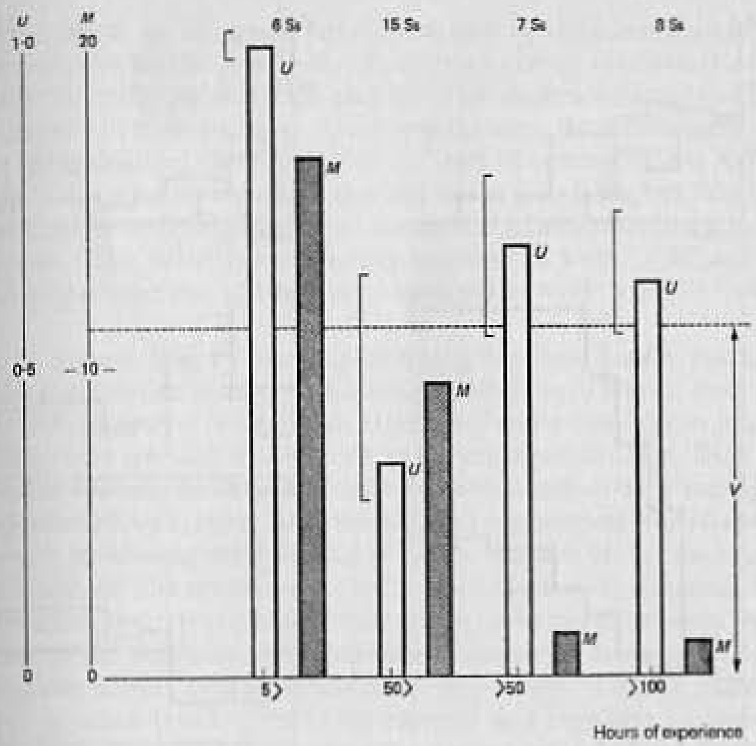


Figure 5 Information analysis of a clerical operation: U is average uncertainty, given a document regarding the values of code variables. The uncertainty remaining, if the contents in a code book were known perfectly, is designated V. Values of $U < V$ indicate spurious overlearning. Points are plotted for novices, clerks with about fifty hours of continuous experience, usually obtained in hourly spells over a month or more, and fully experienced clerks with more than a year's experience. M is a gross estimate of the number of mistakes or omissions at each level of experience.

for anyone but a novice to engage in a 'look up' operation except for a rare or ambiguous item, and, as they become more proficient, trainee clerks do not do so. The uncertainties they experienced, at various stages in training, about the values of the variables when initial values are given (on picking up the input document), were determined by confidence estimation over ranges of possible values. The actual uncertainty is an average over stages in a single pass of the coding operation since variable values are determined in stages and fresh transmission terms have to be considered. In practice, it is impossible to obtain confidence estimates for all trials (the clerk would have no chance to do his work!) so confidence estimates were sampled occasionally, on marked invoices.

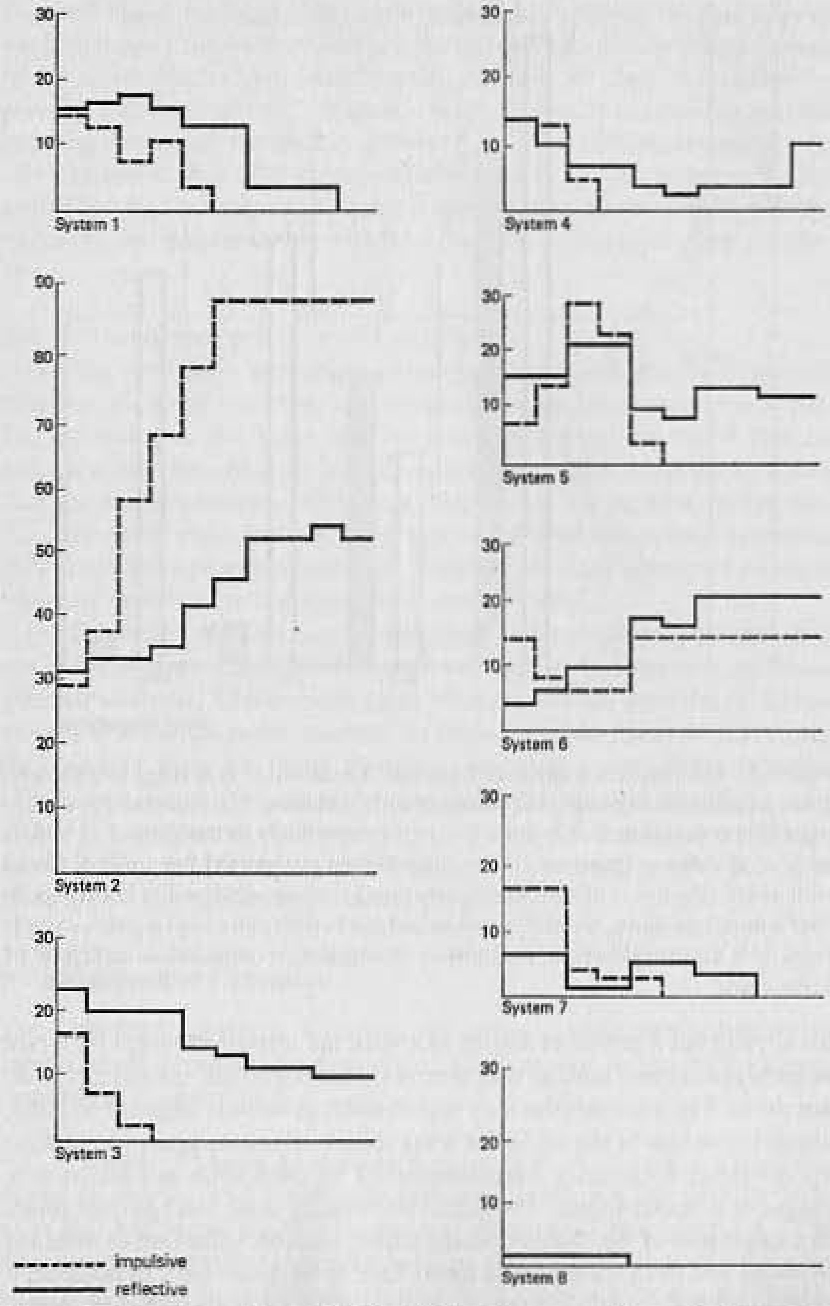


Figure 6 Time changes in probability (of choosing one or other system) averaged over impulsive and reflective groups. Abscissae, time; ordinates, subjective probability of purchasing the system. - - -, impulsives; ——, reflectives.

Each bar in the histogram represents a level of experience (number of hours of prior familiarity with the job) and is an average over data from the number of clerks (Ss) cited at the head of it. The brackets indicate 1 standard deviation unit from the mean. Apart from learning, the dominant effect is over-generalisation. Clerks at the '50 >' level of experience have learned many of the genuine constraints specified by the code book, but they have also imagined or falsely inferred the existence of general constraints that do not exist. This defect is substantially remedied at level '> 50' and the necessary discriminating details are established by level '> 100'.

9.2 Figures 6 and 7 show opinion-sampling data from a purchaser decision-simulation conducted with equipment shown in Plate 2. Respondents with an interest in buying an expensive consumer durable (a heating system) were recruited and brought to the experimental room. Each respondent was independently classified by tests for a 'reflective' or 'impulsive' disposition (Kagan, 1966). After briefing, they spent several hours (between 2 and 4) in choosing between types of system available on the market and one 'catch all' (the system owned by the respondent at the moment). It is, in fact, true that one purchaser would only want to buy one system. In the course of the simulation run, confidence estimates (of the 'opinion' type; there is no 'correct' system) were obtained periodically.[1] Figure 6 shows the mean p_i values ($i = 1, \ldots, 8$) for reflective and impulsive respondents separately. The uncertainty/information indices are calculated from the individual data and are shown in Fig. 7. The simulation itself is an arrangement that allows and encourages the respondent to explore veridical sources of data; advertising, advisory services, comments of other users, informative brochures, graphic material and so on. It is also possible to institute purchase and installation arrangements to which the proper delays are attached. The simulation is paced by a 'clock' which runs at a rate determined by the current transaction (exploratory or otherwise) but, on average, condenses a four month decision period into three hours (probably about half the interval spent, during the four months, in relevant decisive activity).

The transactions are recorded. It would be possible to estimate information *transmissions* from the data gained in exploration of the data base

1. In fact, the respondent is periodically requested to state his goal (i.e. the domestic requirements he chiefly wishes the system to satisfy) by setting rank ordering controls which determine his requirements as a region in a 'space' of product 'dimensions' (the character of which was determined by extensive field studies using interviews with respondents and a repertory grid technique). The same dimensions cross index the data base. Further, goal data, as well as confidence estimation data, are fed into an on line computer model which, in addition to record keeping, predicts the respondent's choices and provides him with recommendations. All this, however, is part of a larger story.

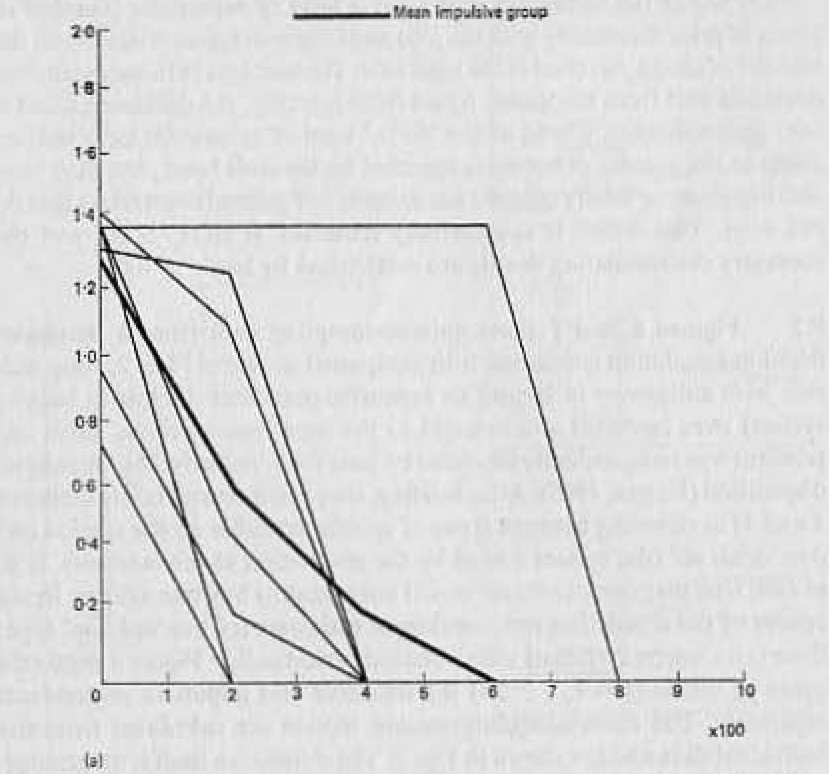
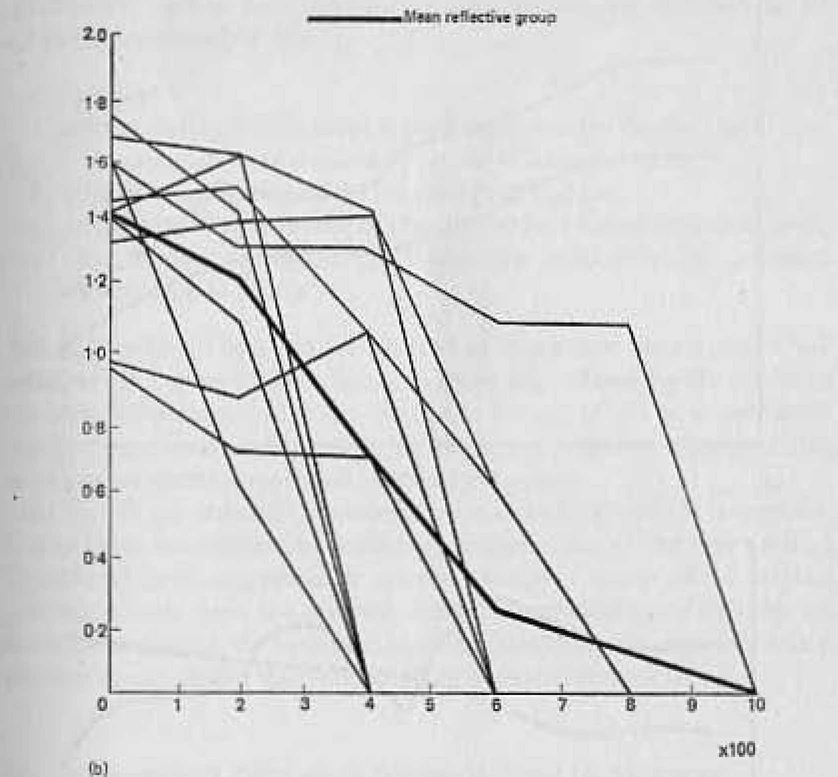


Figure 7 Uncertainties for each respondent and mean values for: (*a*) impulsive group; (*b*) reflective group. Abscissae, time; ordinates, subjective uncertainty.

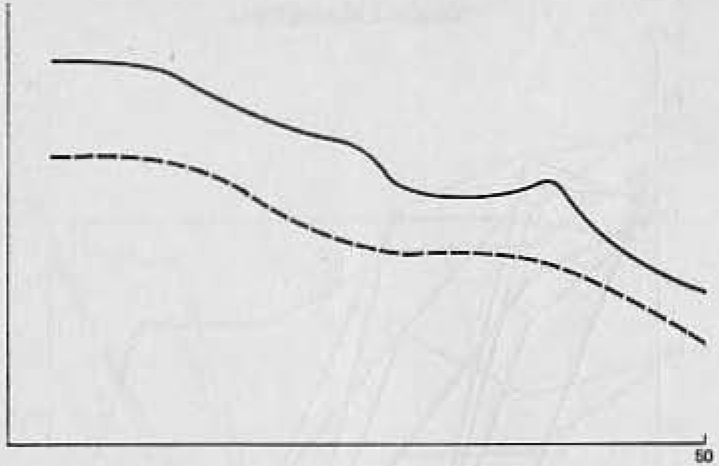
by auxiliary confidence estimates. For this simulation the transmissions were estimated by reference to the microscopic statements in each part of the data base and on the assumption that any piece of data obtained in exploration was noticed and retained.

9.3 Figure 8 is a record of belief-sampling; that is, confidence estimation in which a correct alternative is specified for each trial. The questions refer to items in a data base (very similar to the decision simulation data base of the last section) over which people have a measure of doubt. As might be expected, the provision of knowledge of results after each trial gives rise to a regular reduction in the uncertainty index and a corresponding increase in correct belief (the index specified in section 8.3). In the absence of knowledge of results a respondent becomes increasingly familiar with the items and his uncertainty also is reduced; but his correct belief is not always in register, i.e. the respondent becomes certain about falsehoods.



9.4 Quite complex experimental designs are feasible and may be used to tap *opinion* about such options as products to be purchased, products to be developed, or politicians to be elected; or to ascertain *belief* about the qualities of a product or the actual implications of an asserted policy, or, for that matter, any confirmable issue.

For example, in one study (forty-seven respondents in the socioeconomic C_1/C_2 category) it was found that *if* the effect of advertising produces a change in opinion it always leads to *increase* in the uncertainty of choice. As a practical corollary, once opinion has been changed in favour of a product (or a politician who is promoted or a given line of propaganda) then, to establish this change firmly in mind, confirmatory data must be provided. After that, due probably to cognitive dissonance in respect of the changed hypothesis (Festinger, 1957), advertising in favour of the original product has little effect upon opinion. Some data (with products obfuscated but otherwise of the correct order of magnitude and numerosity) are shown in Figures 9 and 10.



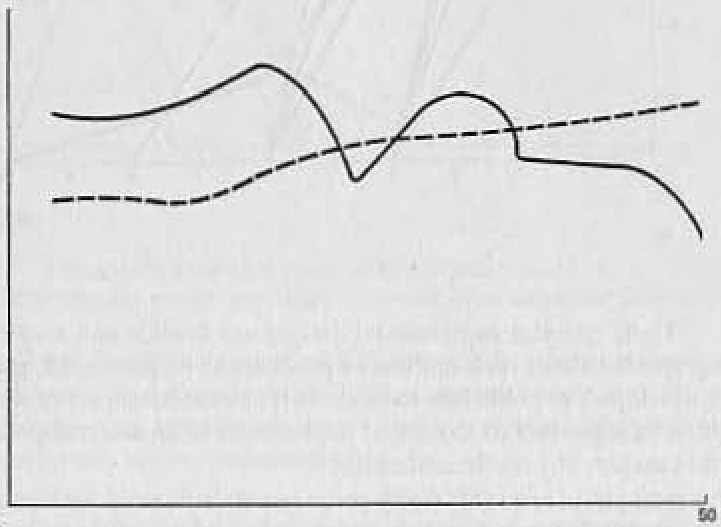


Figure 8 Belief (as uncertainty) and correct belief as a function of repeated questioning: (*a*) with knowledge of results; (*b*) without knowledge of results. Plain line, uncertainty (H); dotted line, correct belief; n, number of trial.

The experimental session begins with pretraining in the use of BOSS (enough to secure probabilistic judgements over genuinely risky or aleatory alternative sets and also *change* of opinion due to *conditional* information). The pretraining materials are now set aside and successive judgements over

development options and *purchase package options* are recorded in the following sequence (see Fig. 9).

1. Initially;
2. after sampling beliefs about factual aspects of the product field;
3. after exposure to advertising in favour of a new product;
4. after an attempt to confirm the change of opinion;
5. after advertising in favour of the original (not the *new* product) and;
6. after training in respect of factual data *with* correct or mistaken 'knowledge of results'.

The last (training) operation is inserted to check that the measured 'uncertainty' is a quantity that is, as it should be, reduced by the receipt of objective information. Opinion change in favour of the new product is clearly perpetuated even though the original product is advertised (the products are alternatives for all practical purposes).

In Fig. 10 the order of presentation of materials (4) and (5) is reversed. Under these conditions (no immediate confirmation of the new product hypothesis) there is appreciable opinion change in favour of the original product. Hence, there is a reversal. Since the new product advertising has not been confirmed, the presentation of data favouring the *original* product leads to an ambiguity, that is resolved by reversion to the original.

10 A Theoretical Point about Information and Its Measures

It is well known that the mathematical form of expression used to represent a selective uncertainty/information index mirrors the mathematical form of an *entropy*; that is, the physical quantity which increases over time in a physical closed assembly as energy is degraded into less specific and usable forms, for example, in the operation of a heat engine or the metabolism of a cell where work is done. The correspondence comes about because, at a molecular level, the increase in entropy can be expressed in terms of an increasing disorder amongst such systems as motions of particles, distributions of elements, as they appear in the descriptive framework of physics. The quantity that decreases when entropy increases '− entropy' is called *negentropy* and as argued by Brillouin (1956, 1964) or, in the domain of living systems, by Schroedinger (1944) any *use of* information demands an increment in negentropy which must be furnished by the provision of energy or degradable materials, such as a foodstuff. Brillouin makes a distinction between 'Bound Information' concerned with the configurations of a physical system and 'Free Information' concerned with the possible configurations of an abstract system having no specially physical interpretation.

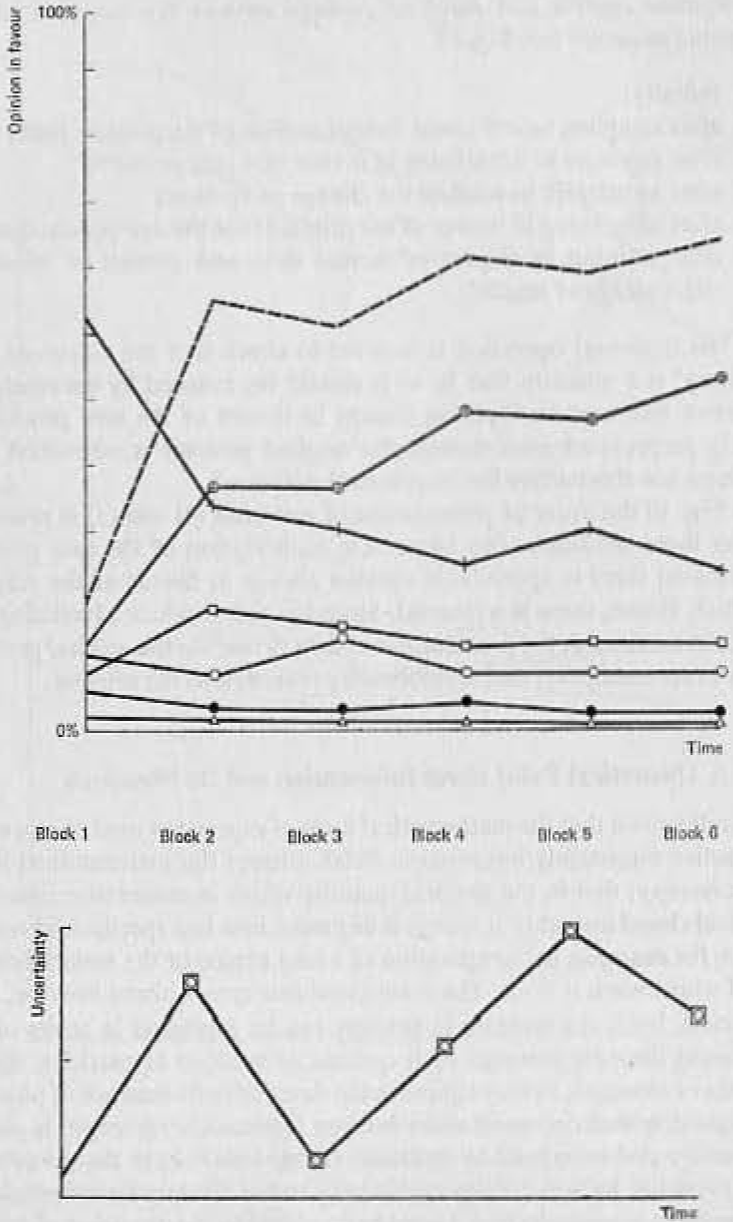


Figure 9 Forty-seven respondents, six products or installations. Key: ⊕, □, new offers; ○, ×, original offers (same fuel); ●, △, original offers (rival fuel); {⊕ ∨ □}, new product alone or in package (original products are thus (100 per cent−(⊕ ∨ □)).

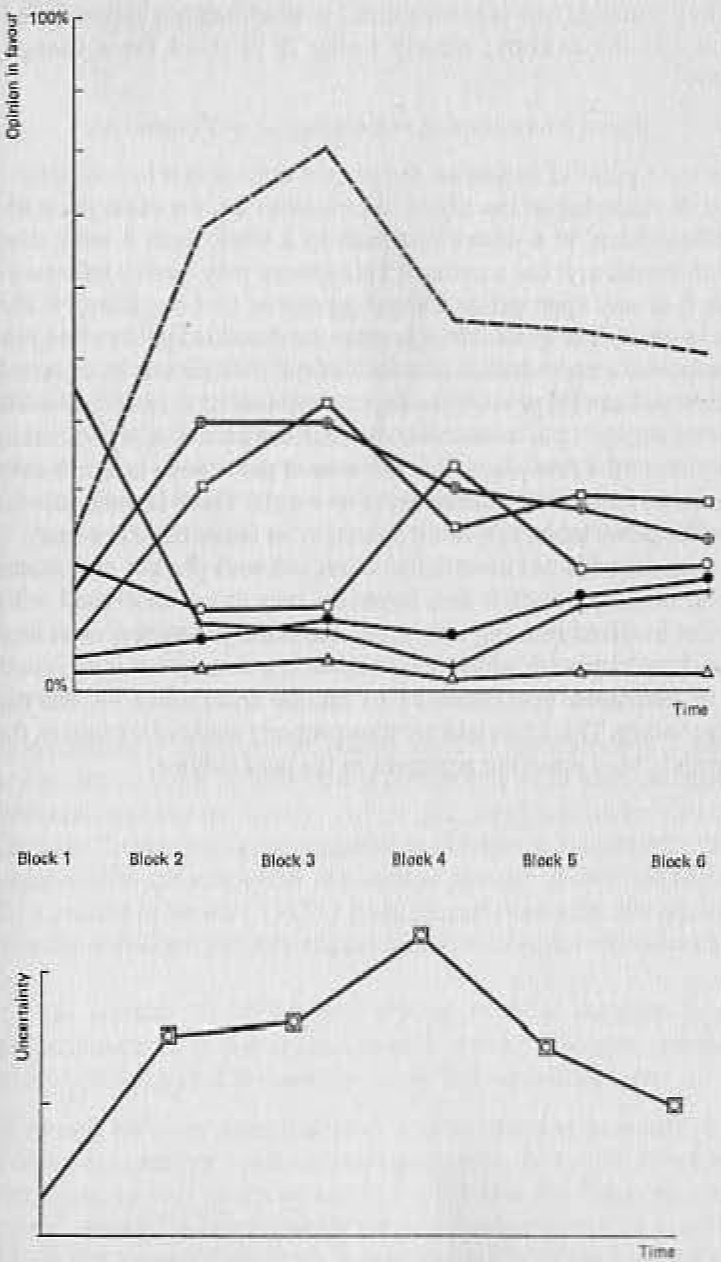


Figure 10 Seventeen respondents, six products or installations. Key: as shown in *Figure 9*. (Study by: E. Pask and P. J. Mason.)

Only in the former case is there a direct correspondence between the information and the entropy; namely (using Δ to stand for a change in a quantity)

$$\text{Bound information} = -\Delta\ \text{entropy} = +\Delta\ \text{negentropy}$$

The main point of impact on the present argument is in connection with the act of retrieving or inscribing 'information' as, for example, retrieving the 'information' in a description such as a table. Such a thing does not *have* information; it has a pattern. This pattern may convey information to you or I, or any appropriate computing engine that can interpret the language in which it is written. In this sense the described pattern has *potential information to a user* which is obtainable (a) if the user can interpret what is inscribed and can (b) provide the negentropy needed to read the description and thus engage in a transaction. Similar comments apply to making an inscription in the first place. For this reason pains were taken to calculate information indices over occurrences or events. There is only information in a contingency table, recording events, in so far as there is a user.

The argument is not essentially concerned with physics and mechanics as such. Some parts of it are, however, very much concrened with the processes involved in using information and the coherency, or at any rate the synchronicity, with which information in a description is activated and used to determine operations or to reduce uncertainty on the part of sentient beings. These remarks are thus properly made at the outset, though they mainly bear upon the argument in the next volume.

2 Machines

Most engineers think of a machine as a tangible entity such as a lathe or a diesel engine. Naturally they include telephone exchanges and computers as machines also, so that 'machineness' does not depend in an essential way upon energetic transformations; to crunch numbers in data processing is as legitimate a machine activity as crunching rock from a quarry. All the same the engineers' image has a healthy tang of materiality about it and at a later stage we shall retrieve the materiality as a general notion of 'efficiency in embodiment and execution'. But at first it is more profitable to look at the other side of the coin and to emphasise the 'process' inherent in a machine and machine organisation.

1 Abstract Machines

In sharp contrast to the engineer, Ashby captures the essence of a machine in the abstract; a point of view that is compatible both with the engineer, and abstract automaton theory. All of the machines under discussion should, until further notice, be regarded as abstract automata but Ashby's treatment of the subject, which the serious student should read (Ashby, 1964b; reprinted in Stewart, 1967) is more general than most and has a built-in guarantee of realisability (find the guarantee, as it is a non-trivial problem).

1.1 The essence of the simplest type of abstract machine, a state-determined machine, is that it constitutes a 'coding of simple succession'.

The crux of Ashby's definition lies in the following concepts:

1. One and only one state occurs at a time. It being assumed that an observer or designer can point out, independently, the system which is said to have states, so that states are sets of (exclusive and exhaustive) alternatives, $z \in Z$, where Z is the *state set* (of whatever is observed to be a machine).

2. That the constitution of Z is unchanging over the interval of the observation.

3. A state is a complete account of the condition of whatever is observed; if it happens that certain properties were manifest in advance of the states,

then a state corresponds to their conjoint values and is expressed by a conjunctive expression (of a language in which these properties are named as unary but (in general) many valued predicates) that refers to all these predicates.

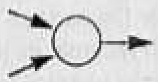
4. The notion of 'at a time' is crystallised by the idea that 'succession' is represented by an ordering index with a successor operation; for example, the index of the integers or the real numbers $\tau \in \mathrm{T} \ldots$

5. It is postulated (a) that there is a machine clock, beating out machine time units τ (for example, as impulses from an oscillator driving a shift register); (b) that any observer has a stopwatch, indicating *his* time units which we refer to as $t = 0,1 \ldots$; (c) there is a one to one correspondence relating τ and t (with due respect to units of measurement for the τ clock might be fast or slow or even variable) that allows for synchronisation between t and τ; and (d) as a waiver, which will be useful later, there may be several machine clocks but if so they are synchronously operated (for example, one clock may start when another clock stops or the pace of one clock may depend upon a master clock). Further, there may be several submachines with different clocks. But, if so, there exists a (cartesian) product machine with states that are a product of the states of the several submachines such that, whatever the clocking characteristics of the submachines, only one state of the machine occurs at once.

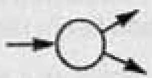
Under these circumstances a state determined machine in isolation is specified by a function or rule

$$Ru; Z \rightarrow Z \text{ or } Z(\text{next}) = Ru\,(Z \text{ existing})$$

Since *Ru* is a function, a picture such as



in which nodes represent states and the arcs stand for changes of state, is perfectly legitimate, whereas the following is not.



As a result, a machine occupying an initial state at some fiducial value '$\tau = 0$' of τ will have a state trajectory $\langle z_0, \ldots, z_\tau \rangle$ that leads either to an equilibrial state or to a cycle of states. The particular terminal condition may or may not depend on the initial state but usually it does so. Hence a complete specification of the machine is given by

$$Rul; Z, \mathrm{T} \rightarrow Z$$

or given the assumptions which have been made by

$$Ru; Z \rightarrow Z \text{ and } z = z_0 \text{ at } \tau = 0$$

It will be noted that in any state-determined system the successor operation is redundant, which restates Ashby's cannon.

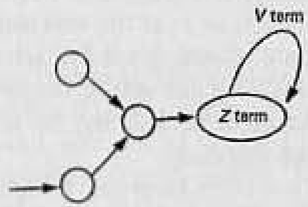
The crux of the matter becomes transparent if discrete *outputs* are assigned to the machine by associating certain of the state transitions with labels, usually numerical labels, v_τ (including a label *null* or no output), so that a state trajectory, $\langle z_0, \ldots, z_\tau \rangle$, might be rewritten as a behaviour (namely, a description of the state trajectory, obtained by labelling) such as $\langle v_0, \ldots, v_\tau \rangle$. Since *Ru* is a function and since it does not change, there can be no more labels than states. Usually there are fewer labels. Hence the mapping 'Output', namely

$$Output; Z \rightarrow V; \text{(defined if } z \text{ changes otherwise } v = null\text{)}.$$

is usually many to one.

Suppose that there are two labels 1 and Null; then periodically *V* assumes the value '1' and the machine is an oscillator. Suppose that a one-to-one correspondence is established (using numerical tags) then the machine is a clock with state numbers on its display.

1.2 As I have chosen to define it, a machine is a dynamic entity. The only sense in which it stops or halts is that its state trajectory enters some terminal state thereafter executing a 1 cycle around this state.



If the machine is state-determined (as so far supposed) then it is isolated (i.e. *Z* is invariant, *Ru* is invariant). Conversely an isolated system *is* such a machine, given agreement with postulates 1 to 5.

1.3 A collection of machines (of this type) is also a machine of this type; trivially, but correctly, if the machines are unconnected. This is so since it is always possible to refer to a state of the entire machine as the product of the states of the smaller machines. Suppose there is a limit upon storage (in the sense of an amount of order that can be accommodated in a unit of fabric), it would be possible to ensure that any physical machine is finite. An appropriate limiting principle is furnished by Bremmerman's limit (Bremmerman, 1962; Ashby, 1968). Take any physical universe you like, where postulates 1 to 5 apply. It is, by the definition of 'physical universe', isolated and, by its nature, subject to Bremmerman's limit; hence, it is, if these postulates apply, a machine of this type, or a bag of them which is itself a machine of this type. In particular, any real (engineer's) machine is a machine of this (abstract) type.

It is worth noting that though I respect the utility of postulates 1 to 5 and the elegance of their development, I do not regard these principles as generally applicable or 'true'; nor do I concur in the philosophical conclusion about the universe or isolated bits of it. However, it is quite important to stress that my disagreement is not based on grounds directly related to the next (probablistic) development of machines.

1.4 Suppose that *Ru* is redefined as a relation rather than a function, so that pictures like the ones shown become permissible. Call the revised rule

RR. It is possible to characterise a new kind of machine (preserving all of the other requirements) by a mapping

$$RR: Z \to Z$$

which may be one to many. If so, the new kind of machine in state z_a (say, at τ) may go into state z_b or z_c at the next instant $\tau + 1$. But it must be in one and only one state at once. Since *RR* fails to prescribe which state this will be, and since it will be just one of the possibilities, it is usual to invoke the notion of chance or probability, i.e. to conceive of a random device that determines the outcome.

It is usually the case, and often harmless, to think of the random device that picks out labels for each next state permitted by *RR* as a wheel of chance, or, considered in a more sophisticated way, an uncorrelated source of events like a radioactively decaying material.

Whichever interpretation you choose, the following points are valid.

(a) Given a state at τ (say $z_\tau = z_a$) and given *RR*, the set of next states (up to all of them) permitted by *RR* is a set of exclusive and exhaustive alternatives.

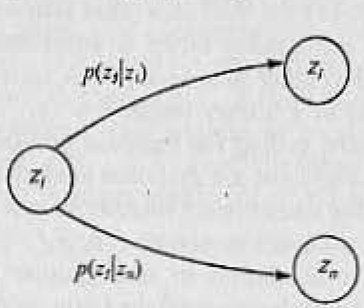
(b) Suppose this set is $\{z_b, z_c\}$ then one of $\{z_b, z_c\}$ is selected as $z_{\tau+1}$, but only one.

(c) An unbiassed random source would have the property that for any value of τ, if z_a occurs, then in the long run (over many occurrences of

$z_\tau = z_a$) on half the occasions the selected state would be z_b and on half z_c; a fact expressed by assigning a positive fractional number, p, to the states z_a, z_b with the property that the p numbers sum to unity and stating '$p(z_a) = p(z_b) = \frac{1}{2}$'. In the general case of m alternatives named by the value of an index j, then $p(z_j) = 1/m$. In so far as this metrical assertion tallies with the frequency count, as the number of instances is increased, $p(z_j)$ is the probability of z_j (and remains so, even if $p(z_j)$ departs from the equiprobability condition that $p(z_j) = 1/m$ for all values of j).

(d) Further, under the same restrictions, the conditional probability $p(z_j|z_i)$ is the probability, given the machine is in state z_i at τ, that it will be in state z_j at instant $\tau + 1$.

(e) As a refinement in machine design, it is possible to assert, for each z_i in Z a conditional probability $p(z_j|z_i)$ that a certain transition will occur. This trick amounts to using a systematically biassed random source or to biassing unbiassed random sources. Any machine of this kind is called probabilistic and is represented graphically as shown below



or, in so far as Z and *RR* are invariant, by a matrix with rows corresponding to all possibles states (those in Z) and columns corresponding to the same states (one instant later) and with entries (typified by the cell $\langle z_i, z_j \rangle$ with entry $p(z_j|z_i)$) that are conditional probabilities. If these entries do not change in value the machine is a Markovian device. Two representations are convenient. On the one hand *RR* can be stated together with a list, for each state z_i, of the probability numbers displayed in the graph of the machine. Call the list *Prob*: the machine is prescribed by

$$\langle Prob, RR \rangle : Z \to Z$$

Equally, if all of the caveats hold good, the conditional probability *matrix* represents the machine on its own.

(f) These are prescriptive expedients. There are also descriptive expedients, i.e. given an unknown but dynamic entity (which it is possible to isolate as an observable in some manner independent of the observation) its state transitions can be observed. Such a machine is generally called a stochastic source rather than a probabilistic machine.

(g) If neither *Prob* nor *RR* change, the source is a *stationary* source (as the term is used in Chapter 1).

(h) Commonly, for reasons of accessibility, it is only possible to observe behaviours $\langle v_0, \ldots, v_\tau \rangle$ and not state trajectories $\langle z_0, \ldots, z_\tau \rangle$. If so, the source is incompletely observable.

(i) On examining any two or more sources, an observer, by virtue of having separated them, believes himself able to isolate them. This commits him to a tentative hypothesis that the machines are isolated; they are isolated to the best of his knowledge.

More exactly, using his observational language he has described state sets that are, according to his prevailing beliefs in the matter, distinct universes U_α and U_β (as in Chapter 1).

On the presupposition of isolation he can base an hypothesis that the machines might be related *beyond his knowledge*, that is, there might be a statistical dependency. Let the machines (alias sources) be called α and β. Let the observer use contingency tables or some comparable method to record event frequencies (either state occurrences, state transitions, outputs, behavioural sequences of arbitrary length $\hat{v} = \langle v_1, \ldots, v_\tau \rangle$ for each of the machines separately, calling the recorded entities r^α (from α) and r^β (from β) and let him carry out a correlation analysis, which is a standard statistical technique for determining whether or not joint events such as the event $\langle r_\tau{}^\alpha, r_\tau{}^\beta \rangle$ or the event sequences $\langle \langle r_0{}^\alpha, r_0{}^\beta \rangle \langle r_1{}^\alpha, r_1{}^\beta \rangle, \ldots, \langle r_{\tau-1}{}^\alpha, r_{\tau-1}{}^\beta \rangle, \langle r_\tau{}^\alpha, r_\tau{}^\beta \rangle \rangle$ have a relation to one another (the *magnitude* and *sense* (+ or −) of the correlation coefficient says more than this).

(j) If, on this criterion, α and β have no relation then they are statistically independent. Computing the (*raw*) correlation as a transmission, we obtain the statement of Chapter 1

$$\text{Independence if and only if } T(\alpha,\beta) = 0$$
$$\text{dependence if } T(\alpha,\beta) > 0$$

1.5 Suppose a machine is not independent but receives an input, at synchronous instants, τ (or that it inspects an input *state* at these instants). Input states are exclusive and exhaustive alternatives like internal states; in particular there is one and only one input state at any instant. Amongst the inputs there is a value *null* that does nothing so that 'no input' and 'input is null in value' are synonymous. The input state at instant τ is designated u_τ. The set of inputs is U.

To accommodate an 'input' or external condition that acts upon the machine, *Ru* must be redefined, as follows, to specify a finite-state machine (FSM) of which the deterministic machine is a special case; namely

$$\text{FSM} \begin{Bmatrix} F: U \times Z \rightarrow Z \\ G: U \times V \rightarrow V \end{Bmatrix} \text{ together with an initial state } z_0 \in Z$$

or, writing these mappings as functions of two variables indexing U and Z

$$\text{FSM} \begin{Bmatrix} z_{\tau+1} = f(u_\tau, z_\tau) \\ v_{\tau+1} = f(u_\tau, z_\tau) \end{Bmatrix} \text{ together with an initial state } z_0 \in Z$$

The alternative forms, derived from *Ru1* rather than *Ru* is

$$\text{FSM} \begin{Bmatrix} F: U \times Z \times \mathrm{T} \rightarrow Z \\ G: U \times Z \times \mathrm{T} \rightarrow V \end{Bmatrix}$$

$$\text{or} \quad \text{FSM} \begin{Bmatrix} z_{\tau+1} = f(u, z, \tau) \\ v_{\tau+1} = g(u, z, \tau). \end{Bmatrix}$$

The rules (the next internal state rule F, and the next output rule G) are still functions, but since they are functions of two variables (indexing the internal and the input states) it is quite possible for the machine to undergo a transition from one internal state into several next internal states depending upon the value of its input. It is convenient to represent all this by a state graph and to comment that (since the inputs form a state set) the inputs can be tagged by symbols, {null, a, b . . .} from an alphabet and outputs (since they also belong to a state set) can be tagged by symbols from a different alphabet {null, A, B . . .}. To avoid confusion, we omit *null* outputs completely in labelling machine graphs, of which the following example (Fig. 11) is typical.

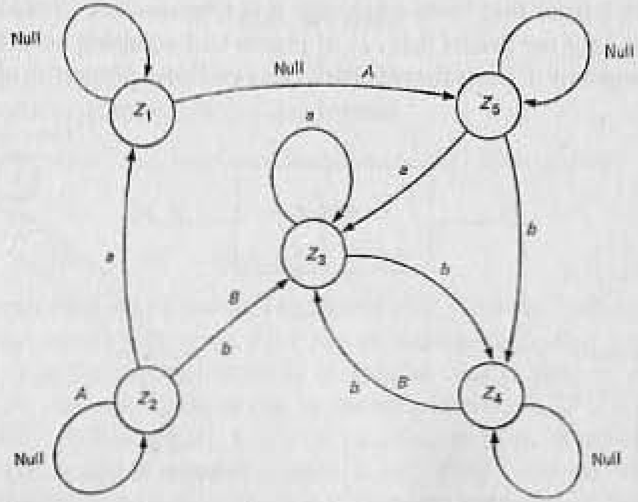


Figure 11 Typical state graph of a machine: internal states, $\{Z_1, Z_2, \ldots\}$; input states, (*null*, $a, b, \ldots$); output states, $\{A, B, \ldots\}$. (Note: to avoid confusion null output states are not labelled.)

If more than one arc emerges from a node (representing an internal state) then these arcs must bear different labels. A machine of this kind is open, in a very real sense, to its input which is sometimes imaged as a tape (advanced by one step for each clock impulse) and sometimes as an environment.

The degree of freedom is considerable in view of the fact that we might just as readily regard the input as changing the rules of operation. That is, instead of writing some gigantic function (F,G) as a function of two variables it might have been supposed that F (say) is made up from different rules F_u and that u selects amongst them. This replaces

$$z_{\tau+1} = F(z_\tau, u_\tau)$$

by $$z_{\tau+1} = F_{u_\tau}(z_\tau)$$

However, there are restrictions. It is essential that the internal state set (or, as will be seen later, the set of irredundant internal states) does not change and the same applies to the input state set.

Further, inputs are strictly clocked and, being states, occur one and only one at a time and are unique. This requirement is usually represented by an input tape (see Fig. 12) read by the machine, bearing one state label on each segment and advanced one step for each τ. Similar comments apply to the output tape (except that its synchronisation is guaranteed by the machine which controls it and prints onto it). Ratchet wheel X moves for each state transition (on each output if the null state is counted). Ratchet wheel Y is moved by the τ clock. Without the arrangement or something equivalent the FSM must decode the actual and possibly asynchronical input into a form that looks as though it is synchronised (conceivably by regarding only the orders 0,1, . . . of inputs and counting an event as an input if, and only if, it has the exhaustive and exclusive properties of a state).

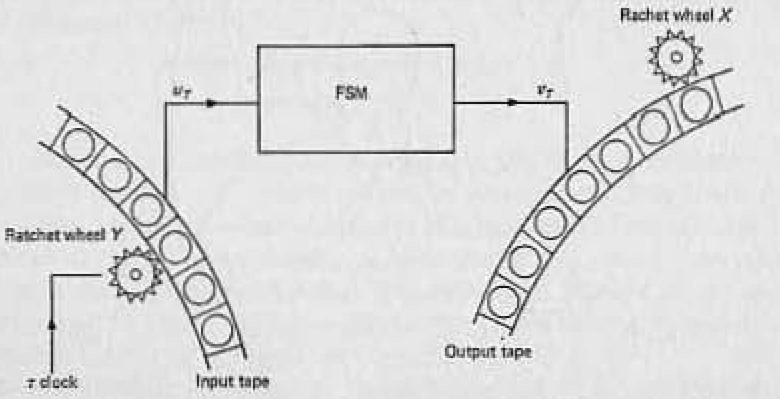


Figure 12 Machine with input and output tapes. Each tape segment bears a state label.

1.6 Just as there are probabilistic versions of machines with rule *Ru* (namely machines with rule *RR*) there are also probabilistic finite state machines obtained by essentially the same construction. It is quite important to notice *how* they are probabilistic.

There is a state transition matrix. But the entries in this matrix may depend upon the input state.

Thus: Probability of state at $\tau + 1 = Matrix_{u_\tau}$ (State at τ)

The inputs are not probabilities, and as before, the internal state set is unchanged, either by the stochastic operation of the device or by inputs applied to it (the converse construction is equivalent; namely, a deterministic automaton, and a probability vector input).

The most succinct and intelligible account of probabilistic automata is an article (on stochastic computation) by Gaines (1969a). The closely related problems of combating the 'noise' or 'random perturbations' that convert the computing elements of a real machine into devices that act like probabilistic automata, rather than finite deterministic automata, are clearly and elegantly discussed in Chapter 3 of Arbib (1964), which covers both the Von Neumann theory of 'redundant', or multiplexed, computation as a means of overcoming 'noise' and the Cowan–Winograd theory.

1.7 Other representations of machines are possible, for example, a listing of ordered sets of states shown as *n*-tuples. Thus, *Ru* is given by a list of pairs like

⟨state, next state⟩

or $\langle z, Ru(z)\rangle$

Similarly *RR* is given by a list of quadruples

⟨internal state, input state, output state, next internal state⟩

or $\langle z, u, v, \text{next } z\rangle$

or $\langle z, u, g(z,u), f(z,u)\rangle$

All these representations use the convenient idea of a state. The problem is that a large price must be paid for the convenience; for any reasonably sized machine the representation is impossible and is anyway generally unnecessary since the machine can be described in terms of 'components' or submachines into which it can be partitioned. Any 'component', of course, must compute a definite function and have a known relation to other components. But, given these requirements, a figure like Fig. 13 can be meaningfully partitioned into components that are FSMs interconnected together (Fig. 14). The picture is particularly neat if the unitary FSMs inside the large FSM belong to only a few categories (for example, see the discussion, by Minsky (1967), concerning machines using McCulloch/Pitts elements as the units). The state of the FSM (if it is an FSM and is properly constructed/partitioned) is retrievable as the product of the states of the units.

Figure 13 Simple representation of a finite-state machine.

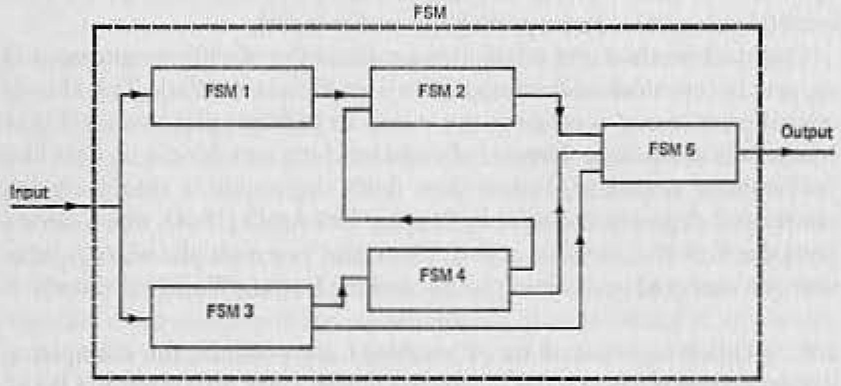


Figure 14 Example of partitioning a finite-state machine into components that are interconnected finite-state machines.

Such figures are simpler to manipulate, for all practical purposes, than state representations. The main justification for talking, improvidently, of states is because a state representation is more susceptible to logical analysis. One way of putting the matter is to note that an FSM can be regarded, in its state representation, as the 'grammar' or 'set of rewriting rules' (list of quadruples) and a formal language. Legal expressions in this formal language are read (symbol to input state) from the machine input tape and are printed (symbol to output state) on to the machine output tape. The internal states are non-terminal symbols in the language. These statements are couched in the terminology for that part of psycholinguistics concerned with phrase structure, grammars and the like. The crucial issue is the existence of a state-to-symbol correspondence. Because of this, particular machines can be characterised as able to 'accept' or to 'generate' certain classes of strings (the expression is 'grammatical' according to the grammar for the particular class of machine). Moreover, different classes of machines are able to compute, i.e. accept or generate strings, only of a certain type.

For example, FSMs, as a class, compute only *regular* expressions (Kleene's theorem).

Because that is so, the otherwise arbitrary idea of an output gains substance. A non-trivial output occurs if, and only if, a machine has recognised (in other words accepted) a string or expression belonging to some class; in which case it is a recogniser of that class of expressions. Further, the irredundant internal states of an FSM are not just mechanically ordained configurations but correspond to histories, i.e. finite expressions, that the FSM is able to recognise.

The serious reader should refer to Minsky (1967). Chapters 1 to 4 contain the most suitable treatment of the concepts I have just mentioned. The development of machine theory in the field of psycholinguistics is brilliant but has questionable relevance.[1] One compact account of machines as grammars and of the expressions they can generate or accept is in a series of papers by Chomsky and Miller (in Luce, Bush and Gallanter, 1963). It is well worth reading these articles and also the shorter account in Bach (1964) of what expressions can be produced/accepted by what machines.

1.8 One very important extension of finite state machines, Von Foerster's (1971) class of finite-function machines, is readily expressed using the notations of Figures 13 and 14 together with two additional symbols; namely '/' to stand for a parametric operation upon the internal state set of a machine and '⊗' to stand for a description of the condition of any machine that is operated upon parametrically (since, in the present case, the machine is a finite state machine, a description can be a record of its input/output history). A finite function machine is shown in Fig. 15

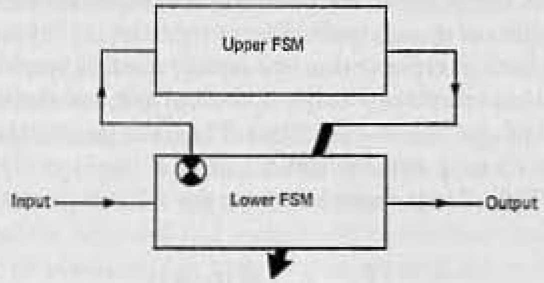


Figure 15 A finite-function machine.

1. There is no doubt about the validity or elegance of this work but I am inclined to the view that knowledge of the syntactic forms admissible for certain machines has little bearing upon the meaningful utterances made either by human beings or by machines. The reader should judge the matter for himself.

and derives its name from the fact (the irredundant internal state theorem) that all irredundant or relevant internal states can be represented as recursively expanded sequences of inputs and outputs, i.e. as state-history functions. The upper machine in Fig. 15, which takes these values as an input and computes an operation that transforms the internal state set (history function) of the lower machine is a function class or *functional*.

We can add comments to this. (a) Certain finite function machines reconstruct themselves in the way that has been described; they correspond to a subset of the self-reproducing automata to be introduced in section 1.10. (b) There is no reason why the lower box of Fig. 15 should not contain many, accumulated, finite-state machines generated, as variants, by this method, though, under the restrictions currently in force, they are synchronous and equivalent to a very large machine built up in stages.

1.9 A finite state machine is 'open' in so far as it does not control its environment. Unless the environment is already expressed as a tape with unique symbols inscribed on it, moved one step for each τ, it must decode the environment so that any input satisfies these constraints.

Now consider a finite-state machine that *can* control its environment, constrained as before, by moving in it or by printing symbols onto it. This environment is a series of linearly arranged storage locations, the 'machine tape', which is used for input and output and to which the machine has access by reading or printing one symbol at any one instant. The tape(s) may have an indefinite length(s); no size or storage limit is imposed. It is decreed, however, there is only one FSM in the environment, or if there are several, in practice they are coalesced to act synchronously as one FSM. On the other hand it does not matter a great deal in the abstract how many tapes there are. For example, the FSM might have an input and an output tape or a collection of storage tapes. These properties are important because they are used both in representing any serially clocked synchronous computer as an abstract machine which is equivalent to it, and also in developing the theory of self-reproductive machines. The entire (abstract) contraption; the FSM and its tape environment taken as a whole entity is a *Turing Machine*, or TM, if it is represented by the following set of specifying functions

$$\text{TM}\left\{\begin{aligned} z_{\tau+1} &= f(z_\tau,u_\tau) \\ v_{\tau+1} &= g(z_\tau,u_\tau) \\ \text{Move}_{\tau+1} &= h(z_\tau,u_\tau) \end{aligned}\right\}$$

where 'move' means 'move on tape' or 'move tape' and has the possible values 'right' and 'left' and *null* or '+ 1 step' and '− 1 step', and *null*. The TM can also be represented by a list of ordered states or symbols standing

for them, for example quintuples like

$$\langle z_\tau, u_\tau, f(u_\tau,z_\tau), g(u_\tau,z_\tau), h(u_\tau,z_\tau)\rangle$$
$$\text{or} \quad \langle z_\tau, u_\tau, z_{\tau+1}, v_{\tau+1}, \text{Move}_{\tau+1}\rangle$$

But a TM is represented more conveniently by a picture (Fig. 16). The ratchet-like object shown there is a device either for moving the tape or the locus of the reading input and the writing output on the tape.

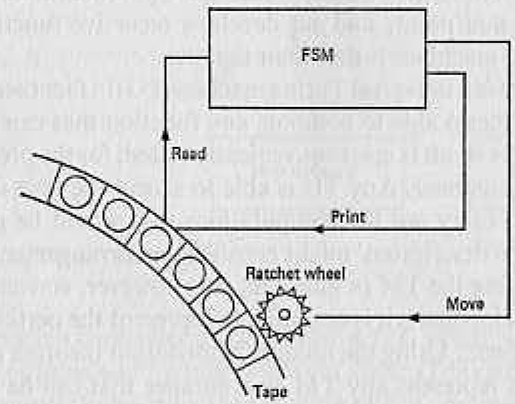


Figure 16 A Turing machine.

1.10 At this point, the serious reader should refer to Minsky (1967), Chapters 5 to 8, and Chapter 10; the other chapters are fascinating but optional. If he thinks, as I do, like a philosophically minded mechanic, this is by far the best and most lucid introduction, but it is also necessary to take up the references to Davis (1958) and it is an advantage (at least for people with an aptitude for pure and symbolic argument) to read Davis in his entirety. Perhaps this is a project to be undertaken at leisure; it appears on my postgraduate's reading list as a 'book to be mulled over'. But the mandate is crucial because the next few sections do nothing more than assert some results and properties. The detailed proofs of these results are relatively unimportant to the arguments in the sequel although they can be appreciated as beautiful and memorised by mathematicians. But it is illuminating to construct (on paper or as an artifact) entities with the required properties; for example, some finite-state machines and at least one universal Turing machine of your own design.

(a) Regarded as a grammar, a Turing machine permits the generation or acceptance of more expressions than the regular expressions of a finite-state machine. Regarded as a productive device a TM is an unrestricted rewriting device.

(b) Similar comments apply to a TM *qua* computing machine and its greater power may be ascribed to its greater storage capacity; for example an FSM cannot multiply arbitrary numbers. But of course, it is possible to simulate a finitary form of a TM in an FSM, capable of multiplying specified numbers. 'Add', for example, only entails the degree of (one stage) retention needed to realise a 'carry' operation. 'Multiply' may involve any number of 'retention stages'. In particular, there is a TM able to compute any general recursive function; Davis (1958) bases his approach to recursive computation upon this point, and he develops recursive function theory in terms of Turing machines rather than the reverse.

(c) Next, there is a universal Turing machine TU (in fact there are indefinite numbers of them) able to compute any function that can be computed by some TM. The result is most conveniently stated, for the present purpose, in the following manner. Any TU is able to accept the tape description of any TM and to carry out the computations that would be performed by this TM. A 'tape description' might consist in an arrangement of the quintuples representing the TM in question. It is however, convenient, and for subsequent developments it is essential, to represent the pertinent TM in an arithmeticised form. Using the unique factorisation theorem of arithmetic, it is possible to represent any TM as a number that can be unpacked by successive factoring to uniquely specify the TM. This numbering trick is essentially the trick employed by Godel in proving his famous theorem;[2] the number in question is often called the code number of the TM.

(d) Without *logical* innovation it is possible to specify a constructing machine which, given the tape description of any machine *TM*, call it *Descr* (*TM*), will produce a (chain-like, or tape-like) copy of *TM*. Further, the constructive automaton (let us call it *Build*) may form part of a larger automaton in which its operations are started and to which their completion is signalled. That is

$$Build\,(Descr\,(TM)) \rightarrow TM$$

(the original description being demolished in the process).

Next, there is, from the existence of *TU*, a further machine (called *Copy* as a tag word) that will copy any description such as *Descr* (*TM*) to produce copies of it. That is

$$Copy\,(Descr\,(TM)) = \langle Descr\,(TM),\ Descr\,(TM)\rangle$$

(the original description being demolished in the process).

2. In this respect Nagel and Neumann (1959) is an extremely readable book and provides a discussion, which is useful for the argument in the next volume, of the classical paradoxes and antinomies; notably Russell's paradox and the Richard paradox.

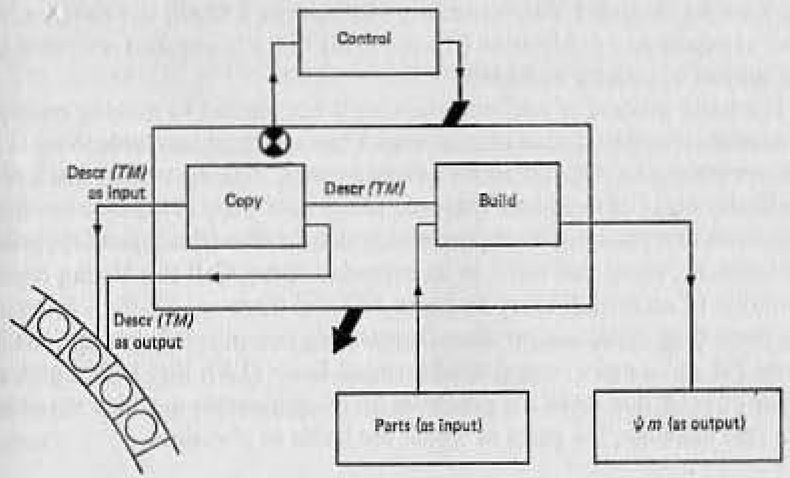


Figure 17 A universal constructor.

Finally there is a controller machine, also a *TM*, which is able to organise the activities of *Build* and *Copy* in respect of *Descr* (*TM*) which can institute the following operation (using the notation of Fig. 16, in Fig. 17).

$$Copy\,(Descr\,(TM)) \rightarrow \langle Descr\,(TM),\ Descr\,(TM)\rangle$$
$$Build\,(Descr\,(TM)) \rightarrow \langle TM,\ Descr\,(TM)\rangle$$

at the end of which *TM* and its description are cut free by punctuation. Call the entire machine *TR*; that is

$$TR = \langle Copy,\ Control,\ Build\rangle$$

so that $TR\,(Descr\,(TM)) \rightarrow \langle TM,\ Descr\,(TM)\rangle$

and *TR* is preserved.

There is a description of *TR*, *TM* and *TU*. Hence, it is possible to represent the process

$$TR\,(Descr\,(TR)) \rightarrow \langle TR,\ Descr\,(TR)\rangle$$

with the original *TR* being preserved.

This argument is a sketch of the demonstration given in Von Neumann (1966) lecture 5 and this reference (not only lecture 5 but all, at least, of part 1 of the book) should be consulted. The entirety *TR* is a Von Neumann 'self-reproducing' machine and it may be formulated in various ways that remove the 'linearity' (Von Neumann's word) of the tape descriptions *Descr* (*TM*). One of these ways, the Tesselation model (introduced in Part 2 of Von Neumann, 1966), is discussed briefly in the next chapter.

Burks, who compiled Von Neumann's lectures as a book, has contributed several papers to a publication (Burks, 1970) which is a crucial reference on the subject of cellular automata.

The basic process of self-reproduction is not limited to making replicas of a machine or population of machines. There are machines, able to execute the operations of *TR* and to do others as well. Amongst them there is a particular class of machines (Myhill, 1970; Loefgren, 1972) that produce sequences of replicating machines (which also do other things) and these are *evolutionary*, using this word in its everyday sense. Call the Turing representation of an evolutionary machine *TE*: and there are *TE* that, given all the parts they need, can produce increasingly complicated descriptions of novel *TR* and a more complicated form of *Descr* (*TE*). Fig. 18 is a picture either of evolution or of the preservation of an abstract machine structure in a real machine, the parts of which are liable to abrasion.

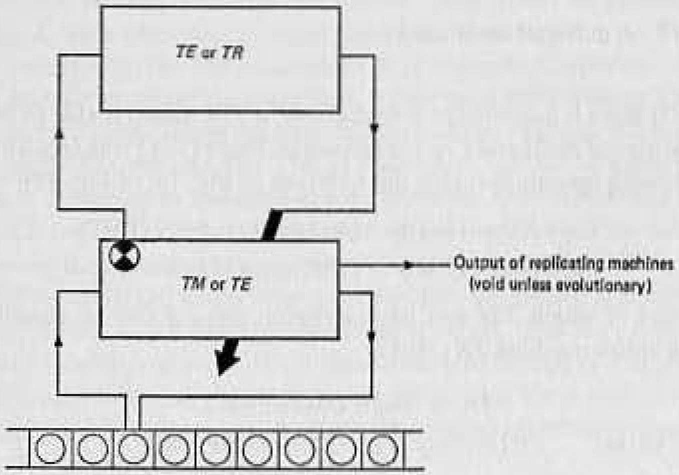


Figure 18 An evolutionary machine.

2 Fuzzy Abstract Machines

The theory of abstract machines is founded upon the concepts of set theory in so far as any such automaton has an internal state set, an input state set, and so on. In contrast, there exists a discipline known as fuzzy set theory, due to Zadeh (1968, 1971, 1973) and his colleagues.

The elements or members of a classical set definitely belong to it or not. The set is equivalently called a 'property'; any object either does or does not 'have' this property and if the property is *designated* by an adjective or predicate in a suitable language it may be *said* to have it or not. In contrast the elements or members of a *Fuzzy* set have a certain *grade of membership*.

This fact can be expressed numerically (as stated, very briefly, in Appendix C).

The concepts of immediate consequence are as follows.

2.1 An automaton with fuzzy input or state sets is a fuzzy automaton, and the computations it performs lead to grades of membership assignments. In general, the automaton computes a fuzzy relation; that is, it produces members of a (fuzzy) subset of the cartesian product of two or more fuzzy sets.

2.2 However, this automaton is strictly and serially clocked (its steps in operation are *not* fuzzy; the index set $t = 0,1, \ldots$ is not fuzzy, nor is the *index* set of the Cartesian product of which a fuzzy relation is a fuzzy subset).

2.3 In general, a step in the computation gives rise to a set of elements with varying grades of membership in each of the property values (alias sets) called the state variables. However, this *fuzzy outcome* can be resolved by various ranking operations involving the maximum values of the resulting cluster of grades of membership. If so, the operation of the automaton is said to be *numericised*, and the outcome at any step is resolved to yield a unique value, say the highest ranking according to a given scheme.

2.4 If the automaton is conceived as having one *state* (not one fuzzy state) at once, i.e. if it is strictly serial, then it can only operate in a numericised fashion. For example, a probabilistic automaton is a special case of a numericised fuzzy automaton.

2.5 If there are several independent copy automata; that is, automata running in parallel but still synchronously clocked by $t = 0,1, \ldots$ then those, in aggregate, constitute a fuzzy automaton and its operation need not be (though it *may* be) *numericised*.

3 Clocking and Programming

We should now recall the distinction, made right at the outset, between real machines and abstract machines or automata. Without qualification, the abstract machine mirrors the operation of an already *given* real machine. Moreover, a real machine can be designed using an abstract machine as a blueprint. But if so it is necessary to augment the blueprint by adding (apart from storage capabilities) a sequencing and clocking equipment to make the real machine change state (to 'be executed'). For example, the machine can be furnished with a clock and a shift register, and these, as in section 1.1, correspond to abstract automata.

But, regardless of the representation used to depict it, the clock and the shift register *are* very definite entites indeed; those mooted in Chapter 1, section 10. The clock automation is executed by dint of a local supply of negentropy which converts a pattern (its design) into *information*. In turn, the clock controls the excursion of a local source of negentropy to scan an *ordered* (usually spatially ordered) storage medium and to convert inscribed patterns (as laid down by abstract design, or in other ways) into information; some of the information modifying (or addressing) the scanning locus. This arrangement (often abbreviated as 'clock') is a very primitive processor; a *real* not just an abstract, 'machine'. In general, the processor may consist in a number of clocks, driving counters and shift registers with ordered storage; these may or may not be synchronised. Finally, the storage may itself be dynamic, insofar as retention is accomplished by local sources of negentropy or by real machines that recompute patterns by retrieving information from them and converting it into further patterns at other addresses or positions in an ordering scheme. The existence of these more bizarre machine organisations is the main reason for the mild dissidence expressed in Section 1.2.

The notion of a processor is especially important if real (computing) machines are not so much *designed* as constructed by an *instruction* giving act, i.e. introducing a program and some data for it to operate upon as an initial state. The object into which the program is introduced is called a *processor* and it must have two basic capabilities: (a) interpretation; so that it can recognise and execute instructions; and (b) an ordering or clocking arrangement that allows it to execute interpreted instructions.

3.1 For a *serial* processor (Fig. 19(*a*)) a sequence of instructions, to set up the internal and input states of a real machine from an abstract machine described by the instructions, is an algorithm or program. Of several possible representations the most convenient is a series of conditional imperative statements ('If . . . then . . . else' statements) combined with the assignment statements (giving 'values to variables' or 'inscribing values on addressed location'). Thus a program is imaged by an ordered series of assignments and statements like

'If A then B else C'

For example ('If $x > y$ then $p \Rightarrow q$ else *to* statements n' or 'If $x = y$ then *to* statement m else *to* statement n'), in toto, computes a specific function.

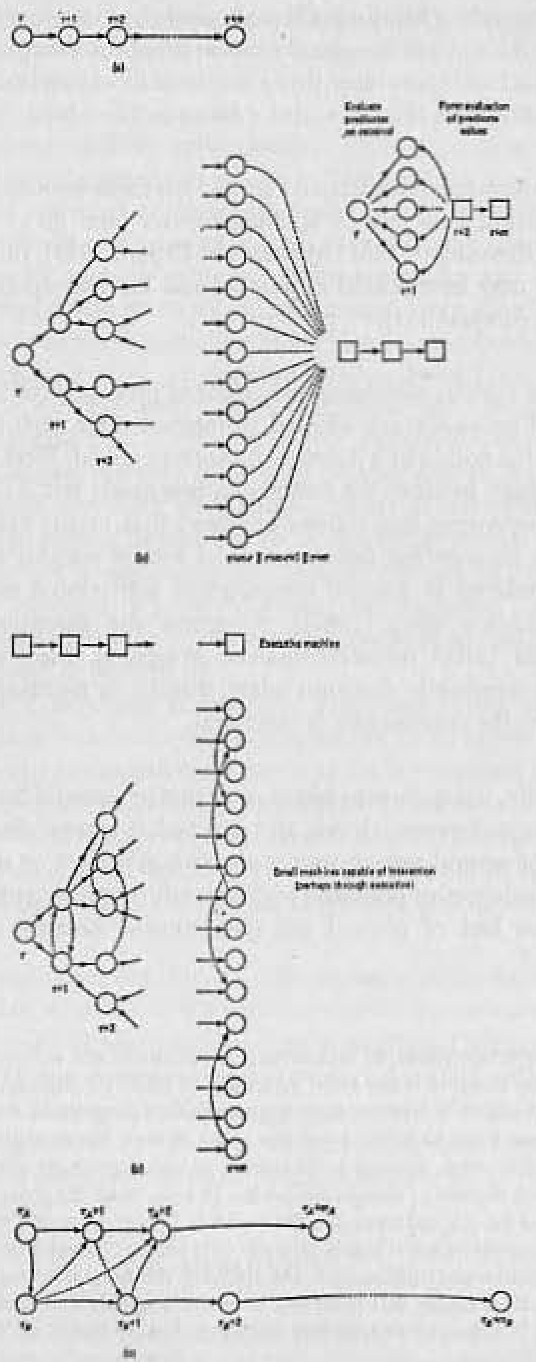
The corresponding fuzzy conditional imperative statements are outlined in Appendix C and are necessarily *numericised* for execution by a serial processor) and compute any fuzzy relation in numericised form. The program, in this case, is often called a *numerical heuristic*.

3.2 A processor is parallel (see Fig. 19(*b*)) if there are many synchronous clocks executing instructions simultaneously. One good example is a perceptron (Rosenblatt, 1961; Minsky and Papert, 1969). In this case fuzzy conditionals may be executed in parallel and the corresponding program is a *heuristic* (unqualified).

3.3 If the various simultaneously clocked processes (or loci of *control*) in a parallel processor are allowed to interact after each execution step (Fig. 19(*c*)), the notion of a *heuristic* is far from trivial. First, it is necessary to avoid conflict between the interacting processes; but it is also possible to obtain *cooperative* interactions provided that means are provided for *ordering* the transactions between several loci of control as they would have been ordered in a serial computation with stored results (see, for example, Simon's essay (1967)). Programs for execution by such a processor are called non-deterministic programs (Manna, 1970). The term 'non-deterministic' does not relate, directly or necessarily, to 'probabilistic'; here, the determinism is structural.

3.4 Finally, it is quite possible to construct processors that are provided with many asynchronous clocks, and in which it is possible to execute the operations of several asynchronous abstract machines or automata (Fig. 19(*d*)). This category of processors is inherently interesting since the initially asynchronous loci of control are synchronised because of *cooperative* interactions.[3]

3. Useful approximations to concurrent computation are achievable using serial processors. One example is the serial execution of SRETIC (Kilmer, McCulloch and Blum, 1969). Another is a recent language recognition programme due to Stavrinides. This programme takes as initial input the order of text; for example, of a children's story. It uses this order, and the juxtaposition of words (without other syntactic constraints) to form clusters of recognition nodes. In turn, these are grouped under further nodes. Whereas the original nodes are dealt with in a *given* order, the higher level nodes, to which they appear as novel inputs on a par with text, act 'simultaneously' in the sense that order is deliberately obfuscated. On applying the same construction heuristic to obtain higher level nodes that overlook both the originals and those nodes derived from them, it is necessary to establish discrimination by means of 'faster' processing (Stavrinides, 1973).



3.5 (a) The program for such a device is a fuzzy algorithm in which the index set (of well-ordered steps) is itself fuzzy; it is also a heuristic.

(b) The cooperative interactions between initially asynchronous loci of control, which synchronise them, are *information transfer* in the sense of Petri (1965) and Holt (1968), a concept quite distinct from the notion of information in Chapter 1, i.e. the interaction may be *cooperative* only if the asynchronous components become ordered by synchronisation (retrieving order, as in section 3.3).

(c) As was the case in section 3.3 as well, conflicting computations must either be avoided or tolerated as the price paid for a richer organisation.

Figure 19 Types of computation execution: (*a*) serial computation, from τ(start) to $\tau + m$(finish). At any point, locus of control may return to pick up stored data. Processor and program are one FSM and it has one state at once (the specification of which includes stored data). This FSM has no input; both initial state and computing inputs are read into processor, before execution starts, as program and data. (*b*) Parallel computation as a tree. Each locus of control '○' may be regarded as belonging to a small FSM but all of these FSMs are synchronous (so may be represented as one large FSM, i.e. their product); furthermore, they are independent and have no input. If the computation entails no conflict, the simultaneously submitted results are equivalent. Otherwise, as shown, they must be evaluated or assimilated by a further, serial FSM, with loci of control marked '□' to which the results are submitted as its input (signified by a dashed line). The loci of control '□' are at a higher level in an hierarchy of control than the '○' loci. Hence, a system with potential conflict is hierarchical. Minsky and Papert's analysis of a one layer perceptron is shown as a special and interesting case on the right. (*c*) Parallel computation with conflict-resolving interaction at each step (input to small FSMs corresponding to '○' loci of control from other small FSMs). If this cooperative interaction is organised by an executive machine at a higher level in hierarchy of control, input and output may take place through the executive (control loci '□') which prevents incompatible results and eliminates a need for an evaluating machine since all the FSMs are synchronous. It is possible to regard the entire processor and program as one large product FSM but, if so, the product (a peculiar one) has components at both levels of control in the system. (*d*) Concurrent computation by processor with initially asynchronous clocks (FSMs, loci of control) labelled $_{A}$, $_{B}$ that become synchronised in conflict-resolving interactions, shown as dashed lines (these represent inputs, as before, but include generalised synchronising or 'change state' input). There are as many clocks as there are loci of control, though the effective number is reduced by partial synchronisation. The need to stipulate an hierarchy of control is eliminated but the operation of the system (in contrast to its 'physics') cannot be represented as a product machine with one, and only one, state at once. Also, there is not *necessarily* a *finish* to the process (we do not necessarily require that $\tau_A + m_A = \tau_B + m_B$); the same condition applies to *start*.

3 Aspects of Evolution and Reproduction

Computer theorists make a distinction between the 'hardware' and 'software' aspects of a system. The hardware includes the physical constituents (whether the machine is really made in the flesh or the metal) and also the structures either permanently or characteristically attached to the *processor*, such as compiler routines and the organisation of data storage in words, blocks, etc. The software is the program. To preserve the realisability of arbitrary TM and TU in equivalent programs/tape descriptions we insist that the storage *capicity* is potentially unlimited.

1 Hardware–Software Equivalence

For any machine, it is possible to trade off hardware for software and/or vice versa provided the restrictions do not prohibit the existence of a general abstract machine. Various authors refer to somewhat different general abstract machines; for example Burks (1970, essay 2) calls it a 'control automaton' while Chiaraviglio, Hagen and Roehkasse (1972) call it an 'iterative abstract computer'. One important point is that such a thing can be given an indefinite number of hardware realisations, so that the trade off is very general. In this sense, which is much stronger than talking about dependence upon physical bits and pieces, abstract machines can be said to be hardware independent.

1.1 Equivalences in Respect of Reproduction The theory of self reproducing abstract machines is *hardware independent*. For example, the self reproduction outlined in the last chapter is kinematic. It is argued, for example, in Burks (1970), that such a formulation runs into undue levels of complexity when it comes to specifying the *parts* and how they are arranged to be 'generally available'. It is possible to rephrase the theory without losing any of the properties discussed (*not*, however, as the reader will note in the reference, without loss of *some* possibilities) in terms of *tessellation automata*, or, in general, *cellular automata* which include, for example, Holland's iterative circuit computers (Burks, 1970, last three essays). These

are all hardware revisions more fundamental than increasing the number of tapes so that descriptions are no longer *linear* (permitted already in Chapter 2). It should be emphasised that these are not the only rephrasings of the theory in current use; others are due to Baricelli (1962), Glushkov's associate Letichevsky (cited in Glushkov, 1966), Toda (1962), Apter (1966), Penrose (1962), Codd (1968), Kauffmann (1971), and my own group (reported below and in Pask, 1969a).

1.2 Tessellation Models The tessellation model, in particular, is very interesting as a curiously aseptic and 'simple' form of representation, and it has recently been shown to be possible (Vityani, 1973) to represent a wide variety of reproductive systems in this manner (various kinds of sexual reproduction, immune response systems, and so on). A very lucid account of self-reproduction is given by Moore (in Burks, 1970, essay 6). The following notes form an overview, not an explanation.

A tessellation is an infinite plane of cells, each containing a finite-state machine with a distinguished 'quiescent' state. The planar configuration is inessential (a 'cube' or an n-fold figure would do as well, or better if it could be as easily depicted); it is essential, however, that the cells are neighbour-indexed in a plane by rows and columns. The finite-state machines are synchronously clocked, each changing state at fixed intervals $t = 1,2, \ldots$ and the state of any one of these machines (i.e. these cells) at $t + 1$ depends upon the state of its neighbours and itself at t. Thus the neighbours provide an *input*. The detailed assignment of neighbours is not critical; for example all adjacent cells may be neighbours of any one cell (as in Moore's essay) or a more limited set of cells may be neighbours. However, the neighbours must be adjacent in the tessellation indexing scheme. Further, the entire tessellation must be homogeneous in the sense that if a transition rule is specified to show how the state of a cell depends upon the states of its neighbours, then this has the same meaning for all cells (Fig. 20). As a result of executing the process determined by the constraints of the tessellation, the initial condition and the chosen transition rule, certain configurations of active and inactive cells are produced, one configuration is matched to another if the two can be superimposed, state for state, by translation on the tessellation plane. In so far as this condition applies, the two configurations are *copies*. A configuration is self-reproducing, if some interval $t + \tau$ can be chosen for any value of $t > 0$, such that it generates copies and these copies are contained as disjoint within it.

The *configuration* is the entity which reproduces. The finite-state machines and their organisation in the tessellation plane is the hardware *in which* this piece of software *reproduces*. Of course, the picture can be inverted (to say that certain state sequences, for example, are reproduced in the software equivalent of the program <initial states, transition rule> generating the configuration). However, the intended interpretation and the convenient one is as specified.

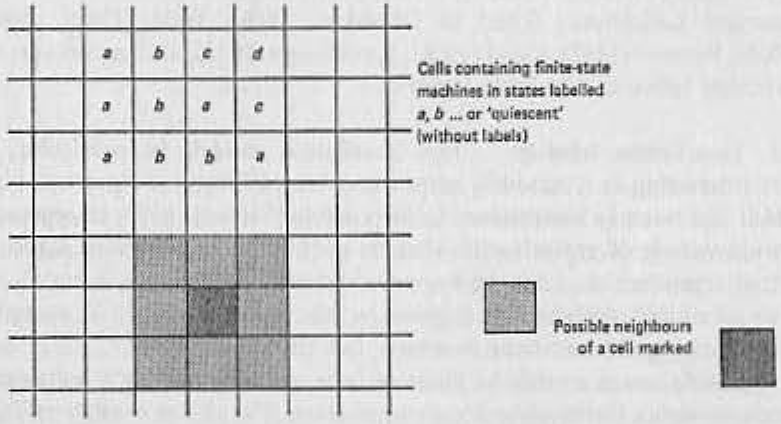


Figure 20 Features of a tessellation plane.

1.3 Primitive Evolutionary Machines Our own work on reproduction and evolution was instigated by the curious dendritic computing systems reported in my book (Pask, 1961a) and in Pask (1959). They make use of a model which superficially resembles the tessellation model (for example, there is an infinite plane and there are configurations). The structures to be described, in outline, are more closely related to Toda's models or some of Apter's. In common with Toda and Apter, the work was undertaken in order to elucidate certain aggregation and exclusion processes manifest both at a social and a cellular level.

1.4 Some Interpretations For example, Wilkins (1963) postulated a form of social homeostasis; inherently stable *roles* differentiate and are stabilised due to mass aggregation and group pressure against deviance from the norms that go along with these *roles* (e.g. common occupations such as 'labourer' or 'squire' and orientations such as 'self-centred' or 'keeping up with the Joneses'). Moreover, if these roles are viewed as self-replicating *organisations* in a social *group*, it should be possible to adjust norms by the introduction of appropriately selected deviant individuals, thus either establishing a fresh norm or switching between potentially available, though not necessarily manifest, old norms. At the cellular level similar effects are manifest in connection with differentiation and immune reactions.

More specific simulations were also performed as part of the project which extended over six years or more, on and off (Pask, 1961b). For example, it was possible, by introducing signalling mechanisms that mirrored the biological signalling substance, acrasin, to mimic the behaviour of populations of cellular slime moulds. These are colonial amoebae with a life-cycle involving (1) a phase of unicellularity; (2) a phase of aggregation of the unicells along chemical gradients to form a partly differentiated 'slug'; (3) a phase of motility as the slug; (4) a phase in which certain cells in the 'slug' form a stalk and others spores; and (5) rupture of the spores to produce further unicellulars.

Only the more general results are used as illustrations in the present discussion.

1.5 Consideration of Pure Machine Evolution The models consisted in machines. To some extent, they were viewed as populations of machines with no particular interpretation attached to them. However, when conceiving them in this way our attention was focussed upon the evolutionary, rather than the replicative, activity of the machine population. This attitude, combined with the requirement to simulate some social/biological systems, determined a much more complex and less tractable design than the tessellation model. The design is thoroughly inelegant (for our interest was not primarily mathematical) but the models exhibit most of the additional features set aside on moving from the *kinematic* to the *tessellation* model for self-reproduction.

With certain qualifications, the work will be described in a way that seemed appropriate when it was carried out; many of the notes are culled from contemporary reprints and publications. But, with rather small modifications, it is possible to convert the evolutionary models of those days into systems that are similar to those that image strategy and concept learning, viewed as a kind of symbolic evolution which takes place in the brain as a processor. These developments are surveyed in the present volume but not properly discussed until the next.

2 Background for Abstract Models

A model for evolution involves the environment in which evolution occurs, as well as that which evolves. With respect to organisms that evolve, or a species or single organism, the environment will be a physical structure in which there is food to consume and form to perceive and the companionship of other organisms. When one considers brain activities that evolve, the environment will be sequences of messages in various languages and various modalities. To machines, the environment is composed of other machines, parts for them, and a supply of negentropy. These environments, and, for that matter, many others, can be represented as abstract constraints (called 'food' or 'neighbourhood', but only as convenient ways of talking about abstractions). The abstract constraints can be identified with a variety of different environments or the attributes of different evolving entities. But these interpretations will appear more or less plausible, to the extent that some may become impractical and others become imperative, at different stages in the evolutionary process. As a consequence, there is a structural uncertainty regarding what it is that does evolve—the organism, an aggregate of organisms, or the process of development of each individual.

2.1 Conditions for Evolution 1. In the real world, evolution occurs when there are distinct elements each of which can survive in certain conditions of the environment and not in others. The issue of whether or not a particular element does survive in conditions that permit its survival depends upon its behaviour, and in any interesting process the behaviour of the elements must be such that they tend to remain in existence. Survival of the *structured* physical material that constitutes the element is a prerequisite of the stability of the organisation that maintains the element, commonly by resynthesis of structural components from raw material ('food') in the environment. Thus, conversely, survival depends upon stability. Cells are typical elements in the biological environment (though it would be possible to cite the reaction centres of some autocatalytic reactions or regions of activity in a network of artificial neurones as perfectly legitimate evolving elements in different environments). For cells, the survival of the energy-transforming mechanism is a prerequisite for maintaining the nucleic acids that chiefly determine the cellular organisation.

2. Survival is conditional. The simplest conditionality occurs if 'food' is available in short supply. In this case 'food' is a source of available negentropy and it can be read as 'money' or 'electric current' or any other conservable commodity without altering the essential condition that, if it is restricted there will be competition between the elements for whatever is available.

3. Either the elements must be capable of reproduction on their own account, given success in the 'food competition', or there must be a locally specified state of the environment such that one element is created when the local 'food concentration' is high (a 'nucleation' process) or both.

4. The reproductive mechanism may take many different forms and act at many different levels, for characteristically, it evolves also. In biology this is Bonner's (1958) thesis, the 'evolution of development'; in genetic evolution it is part of Waddington's (1957) thesis. So it would perhaps be

more accurate, to assert a 'principle of reproduction' as the requirement. 'Reproduction', in any case, means a little more than 'replication', in the sense of creating accurate images of the ancestor. The process of reproduction (at whatever level it is realised) may be imperfect due to ambiguity and the resulting offspring may include variants upon the original.

It is customary to consider an active source of variation as a prerequisite of evolution; for example, genetic mutation. Here the only variation is due to the resolution of ambiguity.

5. Unlimited development of a population is restricted by the rate of influx of the locally conservable commodity, given the code name 'food'. To make sense of this statement there must be a *principle* (in the present arrangement it is manifest in various *mechanisms*) that gives an advantage to some types and configurations of automata over other types.

6. The environment of the automata is so designed that if there are many of them in an aggregate then an aggregate in which the actions of automata are correlated is at an advantage, and such an aggregate (a coalition of cooperatively interacting automata) is replicating in the environment. It plays the same role as a *configuration* on the tessellation plane, but is not the same *thing* as a configuration. It may or may not be the case that the co-existing *coalitions* cooperate as well.

7. For the sake of uniformity we were anxious to embed a similar property in the most elementary constituents, i.e. the automata. One method, due chiefly to D. J. Feldman (and used in model I, below) is a recognition criterion. An automata can recognise what it *does*, retain a record of this activity and, if it encounters an ambiguous situation that requires choice, it resolves the ambiguity by preferring the same activity. Another method (used in model II below) is to assign the cooperative property to complementary *pairs* of automata that are of different *types* and have different possible actions built into their repertoire.

8. Given these conditions inflow of 'food' into the 'environment' initiates a process of evolution wherein elements, once created, tend to form increasingly organised aggregates and successful variants tend to be selected and reproduced.

2.2 Specific Evolutionary Models During the project, many different models were tried out. Two classes of model (I and II) are described and outline flow charted in Appendix D. In model I, there is one complex type of automaton; in model II, there are several simple types which may be composed according to combination and mating rules. In model I the recognition criteria of section 2.1 (number 7) are embodied as part of the automaton and constitutes a mutual cooperation or replication principle. In model II the cooperation is a distributed property of types. Both classes of model satisfy the requirements of section 2.1 (number 3) by mating and copying (a form of reproduction) though other models incorporated nucleation as well.

The following comments on behaviour are based upon between seventy and eighty simulations. The work is dated by the fact that the simulations were run on an ICT 1202 programmed in machine language. Feldman showed great ingenuity in augmenting the limited storage capacity by externally sorting cards representing the automata and machine checking the card-sorter output.

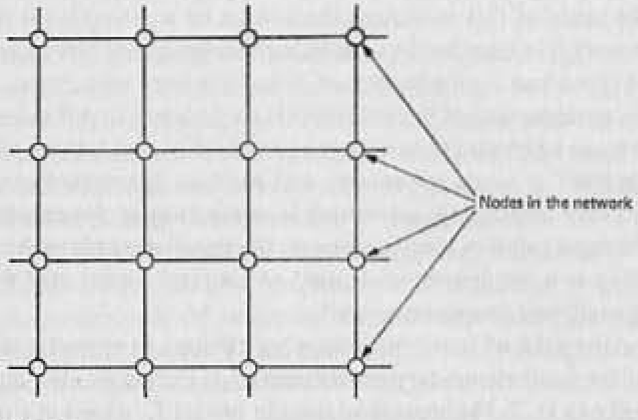


Figure 21 Nodes.

2.3 A Specific Evolutionary Model The environment is a lattice of the kind shown in Fig. 21. Other topologies (torroidal, infinite plane and so on) are possible. Each node in Fig. 21 is a point at which 'food' becomes available and where one of the evolving elements or (as it is convenient to call them) automata, can sit and feed. The connections in the lattice are pathways along which automata can move from one node to another. The possible movements of the automata are shown in Fig. 22 (type * only is relevant to model I, all other types to model II).

The structure imposed upon the environment is a set of pathways or nodal connections and a rule for delivery of food which is conveniently visualised through Fig. 23. Each node (indexed k) is associated with a 'node bucket' (level u_k) filled with food through a constricted aperture. When an automaton (indexed i) rests at a node (indexed k) it eats food from the node bucket and stores it in a bucket of its own (a 'stomach') with level θ_i. It eats food faster than the speed that this commodity is replaced through the aperture. As a result, the automaton is bound to move. The most elementary linear motions for any type of automaton are shown in Fig. 24.

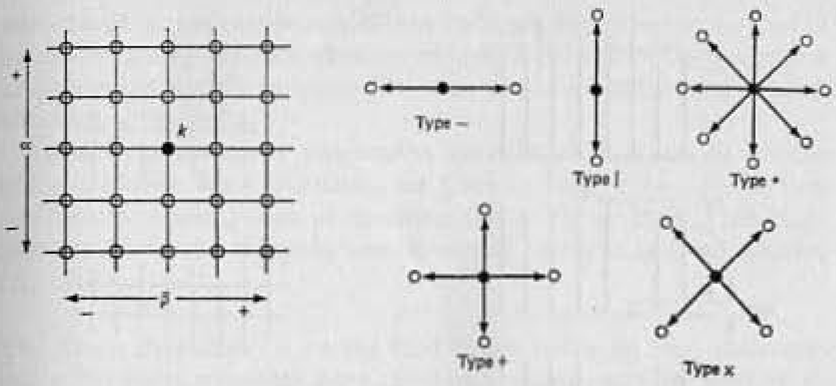


Figure 22 Indexing of node k in environment (far left) and possible motions of automata of different types situated at this node.

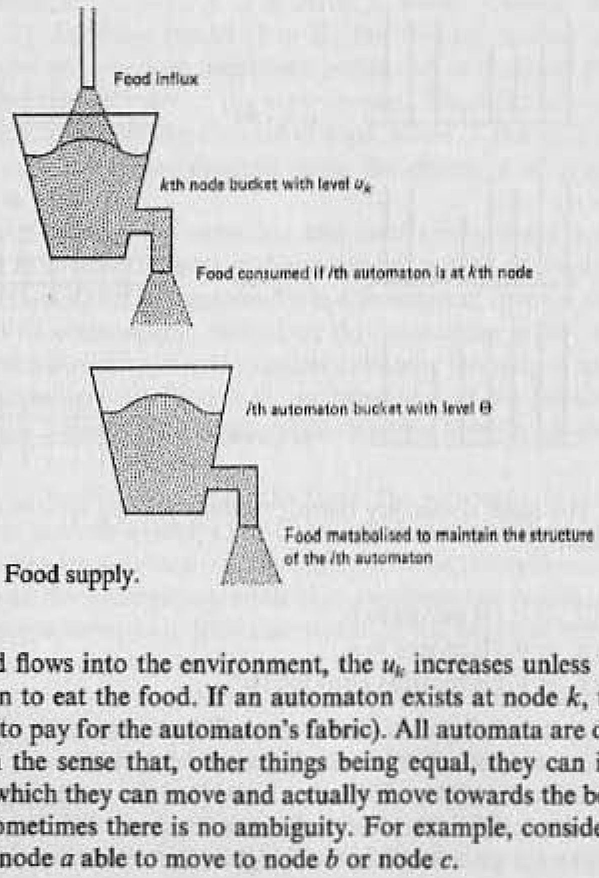


Figure 23 Food supply.

As food flows into the environment, the u_k increases unless there is an automaton to eat the food. If an automaton exists at node k, the food is depleted (to pay for the automaton's fabric). All automata are designed to survive in the sense that, other things being equal, they can inspect the nodes to which they can move and actually move towards the best stocked region. Sometimes there is no ambiguity. For example, consider an automaton at node a able to move to node b or node c.

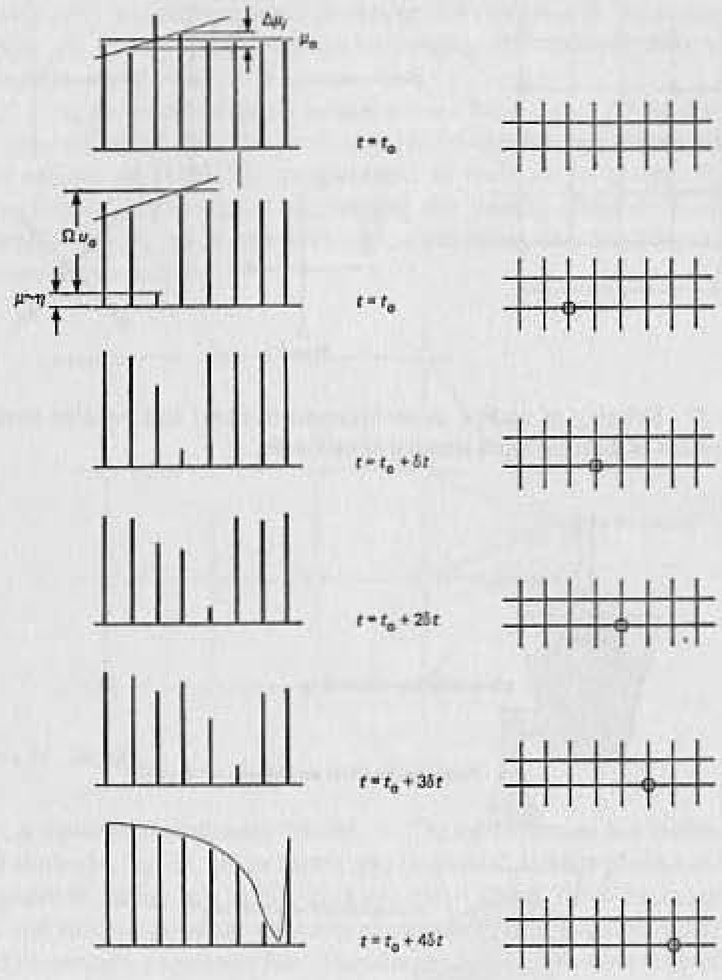


Figure 24 The most elementary (linear) motions for any type of automaton (models I and II).

If $u_b > (u_a$ or $u_c)$ it moves to b,
If $u_c > (u_a$ or $u_b)$ it moves to c,
If $u_a > (u_b$ or $u_c)$ it remains motionless.

If $u_b = u_c$ the situation is ambiguous. Either $u_b = u_c > u_a$ when the automaton *should* move but has no direction determined or $u_a > u_b = u_c$ when it remains motionless. Though a move into the same place is apparently determined, the rate at which the stomach content, θ, must be metabolised to pay for an automaton is set higher than the influx rate; so here, also, there is an impasse. In model I the ambiguity is resolved at the level of the automaton by invoking the recognition criterion (the learned preference weights in the model). In model II it is only resolved at the level of combinations of automata.

Apart from parameter assignments (considered in Appendix D) the initial conditions for a simulation are given (at step O) by $u_k(0)$ (always uniform) by a distribution of automata (either ten or twelve, randomly disposed about the thirty-six central nodes) and for each automaton, i by $\theta_i(0)$.

2.4. Gross Properties As the food inflow builds up food concentration at the nodes, automata move about and combine with one another. In model II there is, on average, an advantage in the combination $A \bigcirc B$, but this rule may be contradicted in regions that are populated by a particular species of automaton (for example, in a region populated by the combination $A \bigcirc A$, $B \bigcirc B$ have a better chance of survival than $A \bigcirc B$). In either model (I or II) the feeding pattern of the prevalent species induces a characteristic pattern upon the food distribution over the nodes in this part of the environment. The mean population size initially depends only upon the rate of food inflow. Later in development the mean population size depends upon the efficiency of aggregates or coalitions as well.

Interaction between the automata and their environment is due to the fact that a behaviour induces a characteristic pattern of food depletion. Now the behaviour of any automaton is a function of its own state and of the state of its environment. But where this interaction is very strong due to the concerted activity of many similar automata the state of the environment is increasingly determined by the behaviour of the population, and as a result, the automata form groups in which the individuals play specialised roles.

Even at an earlier stage, the activity of the automata is cooperative, since pairs of automata behave in a fashion that increases their joint chance of survival (the behaviour would be impossible for the individuals alone). The action of the isolated automata that constitute the pair is correlated. The mechanism sensitive to food concentration has begun to serve a different function, which it reasonably can in these conditions, namely that of a communication mechanism whereby one automaton senses the presence of the other in terms of food depletion. The 'storage' capacity needed to rationalise this statement resides in the inertial characteristics of the environment.

Successful coalitions of automata, linked in this fashion, are reproduced here (Fig. 25, for model I, Fig. 26 for model II). But a population may or

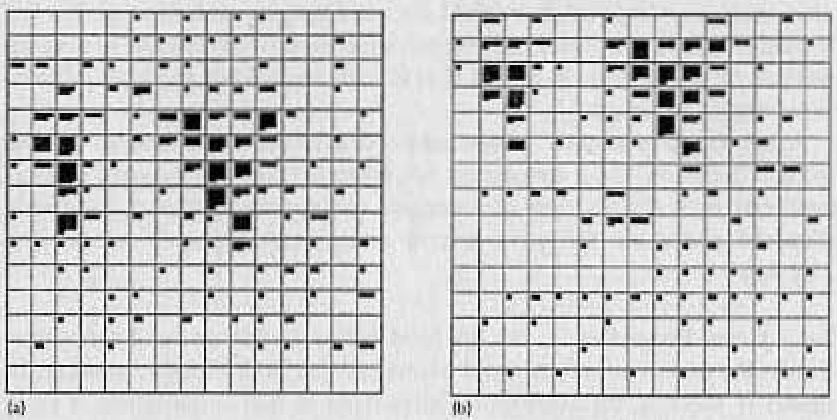


Figure 25 Distribution of type • automata in environment: (*a*) step fifty in simulation; (*b*) step sixty in simulation (ten steps later).

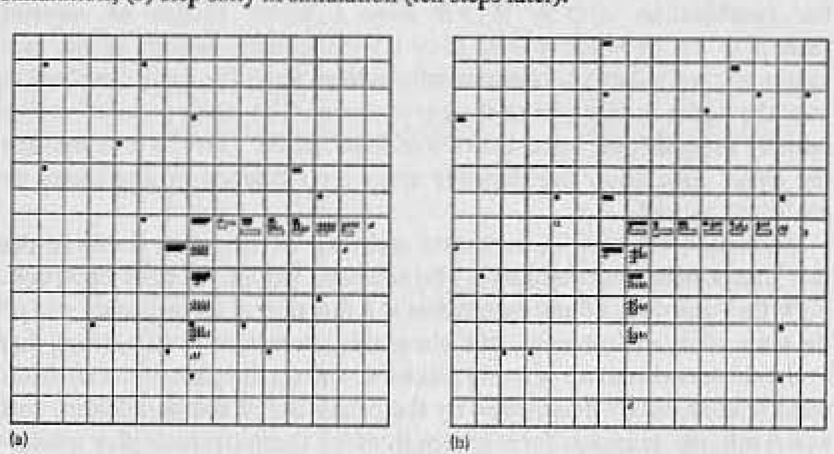


Figure 26 Distribution of automata in environment: (*a*) step fifty in simulation; Key to types: |, □; −, △; +, ▲; ×, ■; (*b*) step sixty (ten steps later).

may not survive; this depends upon the food influx rate and upon an inherent stability (measured as sensitivity to decrements in food inflow).

The development of *norms* can be followed in macroscopic terms by examining clusters of values of vital statistics (for example, the mean of motion weights in Fig. 27) which indicate homeostasis and mass action upon deviants (section 1.4). But salient clusters reflect an underlying configuration of the cooperative and replicating type.

If the density of a stable configuration of automata becomes high enough there is a very interesting discontinuity in its structure, reminiscent of crystallisation. Although the group moves about as a whole and modifies

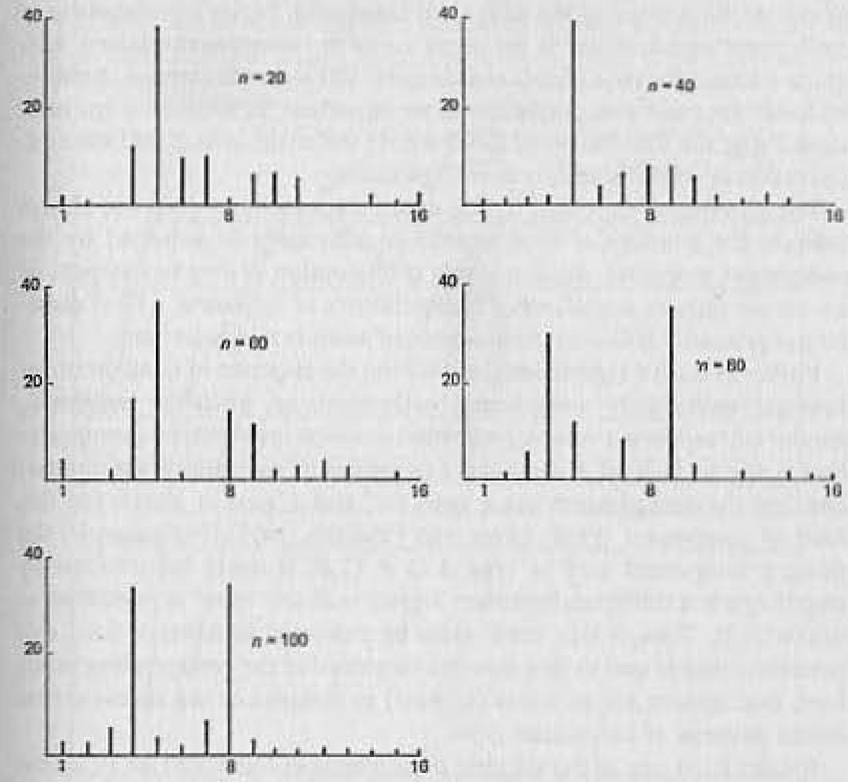


Figure 27 Histograms showing the evocation of a novel norm. The abscissa represents the ratio of horizontal to vertical movement weights and the ordinate the number of automata characterised by each ratio value.

the environment where it resides, any individual has a behaviour that is invariant relative to its neighbours. Some individuals, for example, behave as units in a transmission line composed of a chain of automata, such that movement of one induces food depletion that induces a completely determined movement of the next automaton. Transmission lines have a critical role in the stability of the group since they act as elements in a 'nervous system' that determines a direction of movement for the group as a function of the food available in other parts of the environment (sensed by the relatively unrestricted activity of the outermost automata), and the food available internally.

Regarded as a stable entity, which it is, the *configuration* is aiming to survive in the context of other configurations in a common environment. It makes 'decisions' with this object. It is thus reasonable to ask what is maximised by the group 'decision rule' (by analogy with the θ maximisation of the 'decision rules' of the individual automata). Surely a prerequisite of any group 'decision rule' is maintenance of the components, hence, adequate θ values for the individual automata. But a simple average 'quantity of food' does not seem to be the most important factor, and it has been shown that the *distribution* of food, within the environment that the configuration inhabits, is usually more significant.

The *distribution* as sensed by the outermost automata does not always indicate the quantity of food which can ultimately be collected by the component automata. Since a stable configuration is able to compete or cooperate with, or engulf, other configurations of automata, a food distribution primarily indicates the existence of another configuration.

Rather extensive experiments, examining the response of configurations towards deliberately introduced collections of unviable automata, empirically supported the contention that incipient instability in a configuration is due to lack of a component (a species of elementary automaton) and that the configuration has a 'need for' and a 'goal to search for' this kind of component (Pask, Lewis and Feldman, 1965). For example, the missing component may be type $A \bigcirc A \bigcirc B$; it could be obtained by engulfing a less stable configuration replete with this type[1] or by cooperation with it. Though this 'need' *could* be expressed in terms of food distributions that is not, in fact how it is *described* at the configuration level; here, descriptions are in terms (at least) as complex as the characteristic action patterns of automaton types.

Results from one of the simplest experiments of this kind (the *induction* of a specific change) are shown in Figures 28 and 29 (a control experiment) using the type II model. In a free running simulation, mechanisms of the kind proposed appear to be responsible for a number of ubiquitous phenomena. One of them is shown in Fig. 30. It may either be interpreted as a predator prey interaction, the population phenomena analysed by Kerner (1959), or as the explicit symptom of a primitive immune response, notably in the sense of Bell (1973). Certainly, the oscillatory interaction can be suppressed by actions that are designed to density regulate the reproduction of the configuration *constituents* counting as prey *or* the *configuration* counting as an alien organisation.

2.5 Retrospective Identification Suppose that the stable configurations of automata are identified as the reproducing/evolving entities. (As pointed out earlier, this is a matter for choice; several options are open to the experimenter.)

1. In other models by engulfing a configuration having a more complex type, say $(A \bigcirc A \bigcirc B) \bigcirc (A \bigcirc B)$ that is decomposable into the required type.

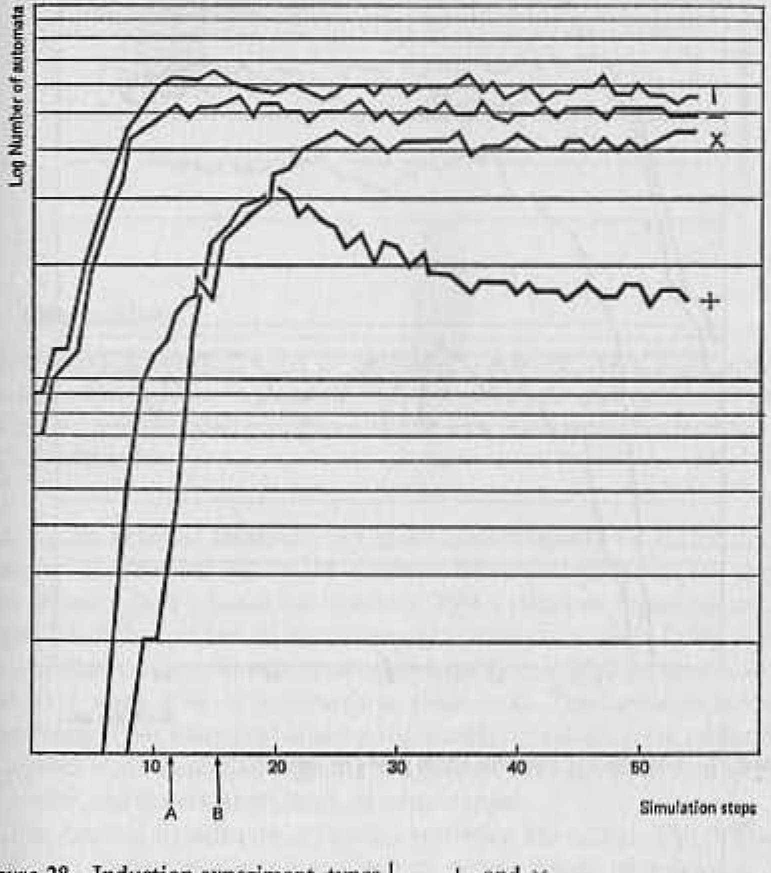


Figure 28 Induction experiment, types |, —, +, and ×.

Suppose the configurations *do* correspond, in this model, to configurations on a tessellation plane. If so, the environment is *not*, under that correspondence, analogous to the tessellation plane. The processor that executes the configuration is a population of automata (adapted automata in model I, or automata of different types in model II). Hence, the analogous 'tessellation plane' is neither homogeneous nor invariant. Moreover the characteristics of this processor are prone to modification by the process undergoing execution.

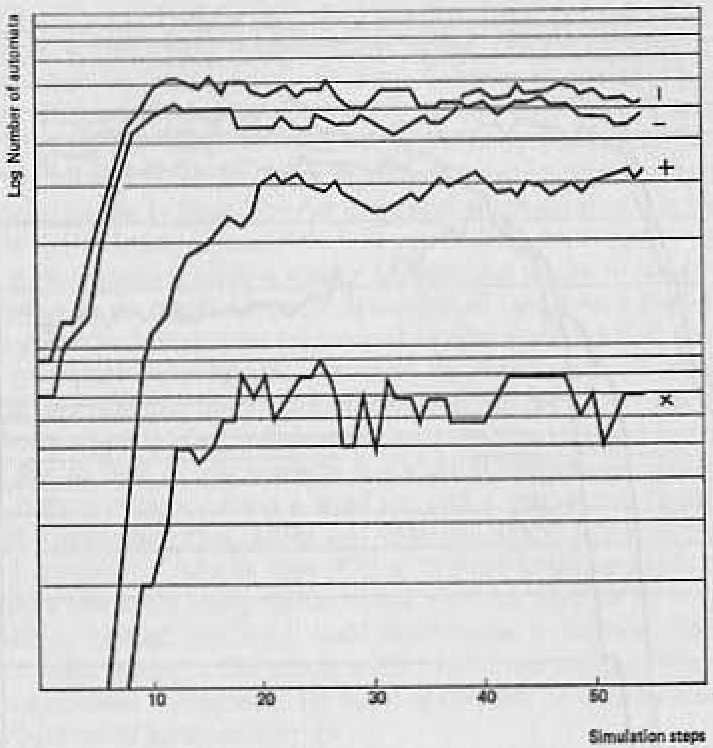


Figure 29 Control experiment, types |, –, + and ×.

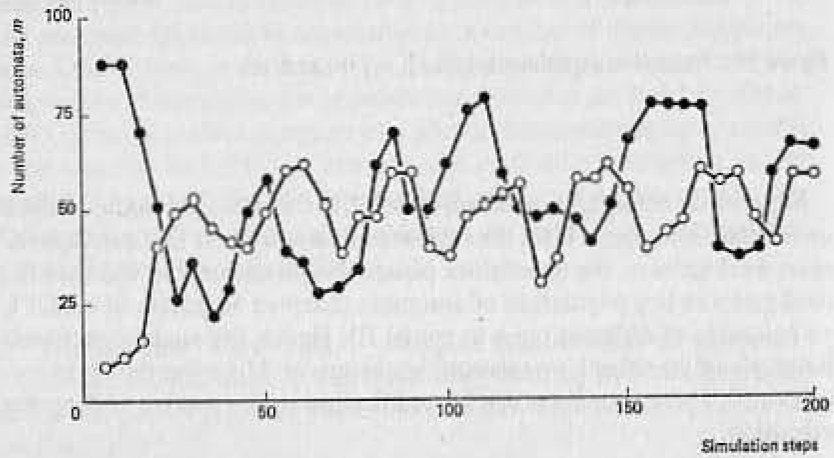


Figure 30 Oscillatory interaction between complex (+ or ×) and simple (| or –) automata: ○, + or × automata; ●, | or –.

4 Relativism

1 Observation

Most scientific writing, even in behavioural science, takes it for granted that an observer can, in principle, act as a numinous and unbiassed entity called an external observer. This point of view is embedded in part but not all of system theory.

1.1 An external observer may make measurements, to determine the state of an observed system for example. Should he partition the system (for instance, into a 'black box' (Ashby, 1964c) called an organism and another 'black box' called its environment) he comes, by dint of observation or auxillary data, to entertain detailed hypotheses. Though of course, he did have some kind of hypothesis to begin with. The organism is worth observing; it has some goal or some characteristic behaviour; it, rather than a myriad other candidate organisms, has been distinguished from the flux of events and chosen apart from its environment.

The detailed hypotheses ultimately arrived at are causal. The organism and the environment are conceived as certain kinds of machine. The observer believes, with certainty or just statistically, that each output was *caused* by some input, or some input/output history. Conversely, he may act upon the organism in a special manner or he may build the environment, as an experimental regulator, in order to do so. In this case, he entertains the predicitive hypothesis that an input will *cause* a certain output or output sequence.

1.2 Moreover, the external observer is causally related to the system under observation; he necessarily conceives it impersonally and refers to it as *it*. One useful consequence of this fact is that he can consistently entertain the notion that the system has, in principle, a state and that one, and only one, state occurs at once. The state transitions are ordered, and this order is interpreted as temporal ordering and can be determined by an observational clock. *Its* internal clock may be synchronised with this clock in principle at least, in the following sense; any process serially executed or not, is *equivalent* to a serial model within this frame of reference.

1.3 As part and parcel of the same issue, only the external observer is allowed to make distinctions and to engage in certain other operations which are noted below. The distinction between organism and environment, for example, is of his making; the organism is deemed incapable of such an activity. If an event of this kind occurred it would either remain unobserved or be blamed upon a chance process; this follows from the scheme in which an external observer can preserve his demarcation and impartial stance; it is not a *truth*.

1.4 Many consequences of choosing to be an external observer are advantageous; most of these have been cited so far. They characterise a theoretical orientation and an experimental method which will be called *classical*. In some ways classical *models* are quite powerful, also. For instance, it is possible, under these conditions, to impute the activities of deductive and inductive inference to the system or its components; sufficiently complex models erected within the classical framework are able to act in this way. Hence, they might, if clever enough, be held to reflect some mental operations of the observer himself were he immersed in the system and observed by another observer impartially.

But other consequences of the classical concept are not as desirable. Consider, for example, the observer's basic mental activity in constructing an hypothesis (section 1.1). It is neither deductive nor inductive but a mix of abduction (as McCulloch (1965), insisted); of distinction, predication, or of cleaving out from a flux of events (the logic of which is due to Spencer-Brown, 1969), and of rule invention to collect the distinguished fragments under the abduced principle. In this connection the reader is also referred to Beer's (1966) commentary on Charles Pierce and 'Fixing Belief'.

Of course, this dogma can be qualified; some hypotheses are deduced from others and some appear as inductive generalisations from previously unfalsified smaller hypotheses. But the initial theme was abduced/distinguished/invented for all that, and even the relatively pedestrian activity of deriving subsidiary hypotheses is necessarily guided by an acumen that stems from this source. Here the inventiveness is manifest as the application of aesthetic or 'know how' criteria used to winnow out 'profitable' lines of development from really or virtually limitless possibilities.

These aspects of mentation, or the corresponding aspects of other than mental processes, cannot be credited to the participants in a classical system because the modelling rules and methods prohibit it.

1.5 The difficulty is amply supported by the attempts that are made to represent such interesting aspects of cognition within a classical scheme. For example, when Wittgenstein (1953) speaks of a language game, he means a game in which the participants have certain roles and in which question-forms are interpreted *as* questions. When Harrah (1966) speaks, in a similar vein, of communication, of debate or interrogation, he means a process involving question and answer forms that are interpreted by participants with roles. In each case the cogency and existence of a question is asserted; there is no denying that questions exist and are, in fact, properly interpreted (as semantic and pragmatic entities, not just as question forms). But the roles responsible for giving this interpretation are stated extra-theoretically for the simple reason that they cannot be accommodated *within* a classical framework. If someone *really* asks a question it may lead him or whoever the addressee may be, to abduce/distinguish/invent. In any case a questioning transaction is not causal and if the roles and the reasons for adopting these roles were expressed within a theory, then the participants could not be regarded as *it*; rather as personally pronominalised entities.

Though the argument sounds less dramatic, these comments about models for sentient interaction (with the role, psyche or interpretation unformulated in the theory) run closely parallel to similar comments about machine cognition. The acts of abduction/distinction/invention are outside the theoretical specification of the model and only *simulated* within it; this is also true of those heuristics that seek out profitably executable productions and entail abduction/distinction/invention only indirectly.

1.6 If we are anxious (and I am) to theorise about such matters as abduction/distinction/invention then it is necessary to set the classical concept of modelling and experimentation aside and to resort to a *reflective* theory (like many rather 'sloppy' theories in psychology and sociology). There is nothing counterfactual about doing so, though much convenience is lost. For example, in a reflective theory, the external observer's special position is lost; with minor caveats he becomes a participant. Further, any reflective theory is prone to paradoxical situations engendered by the possibility of self reference. In *general*, such a theory does not tally with a classical theory; though in special but important cases it does so. (The trouble is that there is a basic indeterminancy in specifying the special cases in advance, or picking them at will.) The roots and beginning of this development are stated in the present volume but the main and serious argument is deferred until the next.

1.7 Is there a true dichotomy, that insists that the theory must be classical or reflective? The distinction is valid; but fortunately there is a continuum of theoretical types and experimental methods that are laid out between the classical and the reflective extreme.

1.8 One route of approach to a reflective theory, the route to be adopted in the sequel, goes by way of relativistic theories and methods in which the distinction between organism and environment is systematically eliminated (whilst the external observer's posture is retained) until, at last, the external observer's position is undermined.

2 Development of Relativism[1]

The gap between the classical concept of a theory and a reflective theory, and the experimental methods that also go along with them, can be bridged by a series of intermediary cases of theories, models and methods shown graphically in Figures 31 to 35.

The figures employ the following symbolism or shorthand. In each figure there is a modelling universe for an observer. The models may be abstract and intellectual, for example, mathematical constructs of one sort or another, or artifacts, for example, physical devices and computer programs like the learning models described in Chapters 5 and 6. Readers who prefer tangible instances to begin with should scan Chapters 5 and 6 before starting the (next) section 2.1. The basic requirement is that a model shall be communicable to other observers using an observational language (usually scientific English together with formal schemes like algebra to augment it locally). It is a metalanguage for talking about models as a class and also for talking about real structures and laboratory experiments. In each figure there is a real or experimental universe. It contains an experiment, meaning any constructible or coherently observable situation. For brevity, the discussion is confined to experiments involving an organism or subject. The experiment comprises a subject/organism and a regulator which in Fig. 31 is a programmed environment and in Figs. 32 to 35 is one of the null-point/steady-state mechanisms described in Chapter 7 or an adaptive regulator (Chapter 8). I suggest a scan of Chapters 7 and 8 before starting the (next) section 2.1, and readers who prefer tangible instances will find it essential. The modelling universe and the experimental universe are shown as dotted outline boxes.

In all cases it is assumed that an observer has a description of his model and a possibly imperfect description of the experiment. Though the form and scope of the description could be detailed, this refinement is avoided

1. Relativism is not without exponents. For example, Helson's *adaptation* level theory is explicitly relativistic and Helson (1964) discusses the philosophical implications in the introduction to his book on the subject. However, in that case, the relativistic frame of reference is an adaptation level for the organism or subject, rather than an observer or an experimental situation.

and the description is indicated by a specific symbol ⊗. Further, the external observer can operate in various ways upon the models and/or experiments under scrutiny. An operation that is unrestricted, apart from the rules of the observational language, is represented by a filled-in parametric arrow ↙ penetrating whatever is operated upon. To limit the scope of an operation, the arrow never penetrates an entire dotted box but only some part of it, a subsystem demarcated by a plain line box. An operation, of whatever scope, that *is* restricted; namely, an operation which would jeopardise the systemic structures, asserted by a figure or its text description if they were carried out in an untramelled fashion, is symbolised by a blank parametric arrow ⇙ ; generically the restricted operations are called 'tuning' operations. This is a commonly used jargon in the field of large population modelling and indicates adjusting parameters so that the model fits the facts or adjusting an experimental design so that it satisfies a model.

Finally '→' is a unidirectional (information flow) coupling and the double arrow notations '↔' and '↔' stand for bidirectional flows.

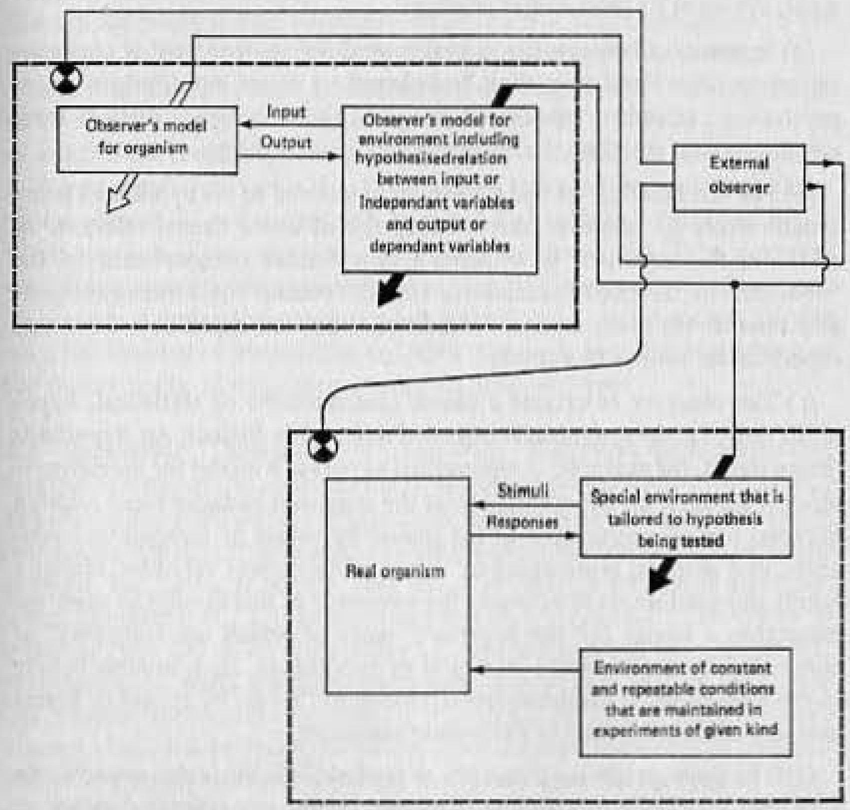


Figure 31 Classical method.

2.1 Figure 31 depicts the classical experimental method.

(a) The observer is an *external* observer in so far as he can view the experiment and any participant (here dubbed the 'organism') impersonally as an *it* which is studied.

(b) Because he is in this position he can use a clock or his sense of time to order events. In particular, time (mapped onto the integers) imposes a successor-ordering upon universes (both modelling and experimental) so that either universe is constant and there is equivalence between temporally distinct universes. $U(t) = U(t + 1)$.

(c) It is always possible, on that account, to distinguish between the *organism* and its *environment* in the following very definitive sense: the grounds for making the distinction in the experimental universe are the same as the grounds for making it in the universe of a model for the experiment. Hence the distinction appears in the model as well and it represents the same thing; for example, the boundary of the organism with the environment, drawn at its sensors and effectors.

(d) Interaction between the organism and the environment is also conceived as causal and may thus be referred to as an input/output or, in psychology, stimulus/response exchange. This exchange is distinct from any interaction entailing the observational metalanguage.

(e) The box labelled 'special environment tailored to the hypothesis being tested' refers to whatever surrounding the observer deems relevant. In addition, the organism is embedded in a further compartment of the 'environment' designed to exclude or annul or swamp out extraneous inputs and thus to maintain constant conditions that are repeatable from one experimental session to another.

(f) The observer entertains a *causal* (deterministic or statistical) hypothesis only, i.e. the organism is regarded as *it* and is distinct. An hypothesis (being tested, for example) is represented as part of a model for the environment. Typically, an hypothesis about the organism includes some relation between inputs, represented in the model by values of independent variables, and outputs, represented by values of dependent variables. Under a family of hypotheses (or acting as the generator of this family) an observer entertains a model for the organism, parts of which are supported or eliminated by confirmation or denial of hypotheses. Such models belong to the class of serial machines (recall comment (b)) or, by appeal to Bremmerman's limit, the class of finite-state machines.

(g) The appropriate macroscopic or probabilistic variables are *selective* uncertainty/information indices (Chapter 1) and are estimated either as

statistical indices or indices of selective work. In particular, the universe is constant so that $U(t) = U(t + 1)$.

A theory, within which an external observer can make inferences, is a metaphor and is expressed in the observational metalanguage, which asserts a strict analogy, that is to say, it is either an isomorphism (one-to-one correspondence) or a homomorphism (many-to-one relation—preserving correspondence) between a *relation* coupling 'observer's model for organism' with 'observer's model for environment' and a further *relation* coupling 'real organism' and 'special environment'. Tenure of the analogy is contingent upon the existence of its domain of interpretation which (as below) is an identification between the 'model' universe and the 'experimental' universe, i.e. it holds if a metalinguistic proposition 'there is an identification', is true.

An identification, usually obtained by specifying measurements and the like, is a stable condition in which the boxes or subsystem boundaries enclosed in 'model' and 'experiment' retain their integrity and in which those in 'model' are in register with some or all of those in 'experiment'. The crucial point is that variations which do not explicitly appear in the model are 'constant conditions' in the experiment (i.e. the constant condition 'Box' has a unidirectional coupling directed towards the organism). In other words, the material analogy underlying the theory has a universe in which these conditions *are* held constant as its domain of interpretation.

The external observer in Fig. 31 carries out two kinds of operation: (1) by means of unrestricted operations, he poses and tests hypotheses which, if valid, permit him to make inferences within the *theory* and (2) by means of the 'tuning' operations, he adjusts his model for the organism so that it can be identified under the measurement procedures at his disposal. In this way, the contingent proposition becomes true, and the material analogy of the theory holds, in its proper domain of interpretation.

2.2 Relativistic Theories A *relativistic* theory/model/experimental method is necessary in so far as the organism (more plausibly, the subject) engages in activities such as exploration, attention direction or non-trivial learning in which he exerts control over his environment and consequently changes the universe (either real or modelled) in which he operates. Thus, in general, $U_\alpha \neq U$ (a constant) but $U_\alpha \rightarrow U_\beta$ and $U(t) \rightarrow U(t + 1)$ where $U(t) \neq U(t + 1)$. If the subject is *designed* to act in this manner and it is desired to study the phenomena (learning or exploration) that cause the trouble, then a classical model is inapplicable. In order to achieve closure at all, the subject's excursions must be compensated and founded within a (larger) universe ($\mathscr{U}$ say) such that $U_\alpha \subset \mathscr{U}$ and $U_\beta \subset \mathscr{U}$ (or $U(t) \subset \mathscr{U}$ and $U(t + 1) \subset \mathscr{U}$). Under these circumstances, the constancy required for any kind of external observation is maintained by a compensating regulator and the 'constant condition' is a dynamic but stable *interaction* between the subject and the regulator.

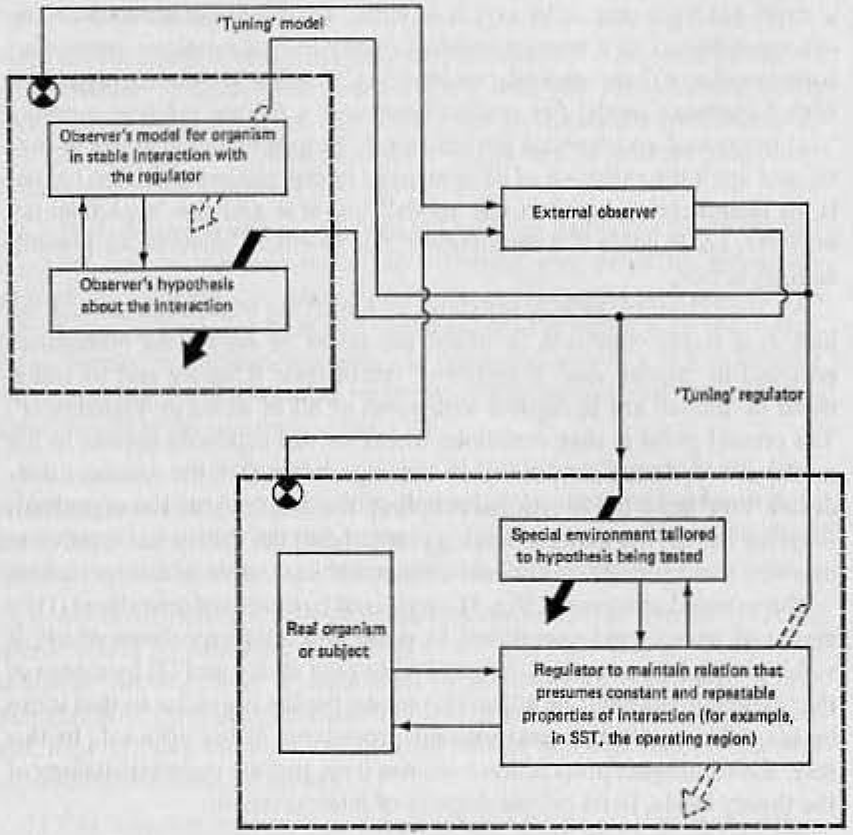


Figure 32 The relativistic method using a null-point or steady-state technique. 'Tuning' connection may also be interpreted as a design process for the regulator in which a computer programmed model (for instance of the type in Chapter 6) is run predictively in order to obtain effective regulator design.

Figure 32 depicts a simple *relativistic* method, detailed in Chapter 7. The constant condition is maintained by a regulator, a device that holds the organism in an 'operating region' where he is able to perform a task and to persist in its performance. The appropriate form of model represents the *steady-state system* of subject and regulator in interaction, and is instanced in Chapter 7. 'Tuning' operations increase the size of $\mathscr{U}$ and may be refined by executing the *joint system* model as a predictive device (under various

parameter values or hypotheses) and by using the parametric adjustments that stabilise the steady-state model to improve the design of the *real* regulator in the experiment.

The salient features of this relativistic theory/method/model are listed in the following previously established order.

1.(a) The observer is an external observer.

(b) He can order events, as before, *except* those within the joint system.

(c) If he is to maintain the fundamental constancy, the external observer can no longer make a clear demarcation between that compartment of the environment responsible for constancy of conditions and the subject (unlike Fig. 31 the coupling is bidirectional; it is only possible to observe a stable regulator/subject interaction).

(d) The events *within* the joint system are not *observed* as causal but *are* distinct from any transactions in the observational metalanguage.

(e) The environment that embodies the observer's experimental hypothesis *can* be distinguished from the real regulator/subject system, but *not* from the subject in isolation.

(f) The observer entertains causal hypotheses *about stable modes of interaction* in the joint system and his model is a model for this entity taken as a whole, as in Chapter 5. It belongs to the class of finite function machines.

(g) The appropriate macroscopic or probabilistic variables are indices of self-organisation, derived from selective uncertainty/information indices, but which are *not* constrained by the requirement that $U(t) = U(t + 1)$; in fact, this condition is rarely satisfied.

2. A theory, *within* which the external observer can make inferences, is a metaphor expressed in the observational metalanguage that asserts a strict analogy between a relation coupling 'observers model for interaction' to 'observers hypothesis about interaction' and a further relation coupling 'real interaction (both components)' to 'special environment'. Tenure of the analogy depends upon the existence of its domain of interpretation, which is an identification between the 'model' universe and the 'experiment' universe.

3. Identification is secured by the stable operation of the regulator.

4. The external observer carries out two kinds of operation. (a) By means of unrestricted operations he poses and tests hypotheses about a stable interaction. (b) By means of 'tuning' operations he adjusts the real regulator so that it secures stable interaction and brings his model for the interaction into register with the reality.

5. The arrangement shown in Fig. 33 is a special case of some practical importance. All 'tuning' operations are built mechanically into an adaptive

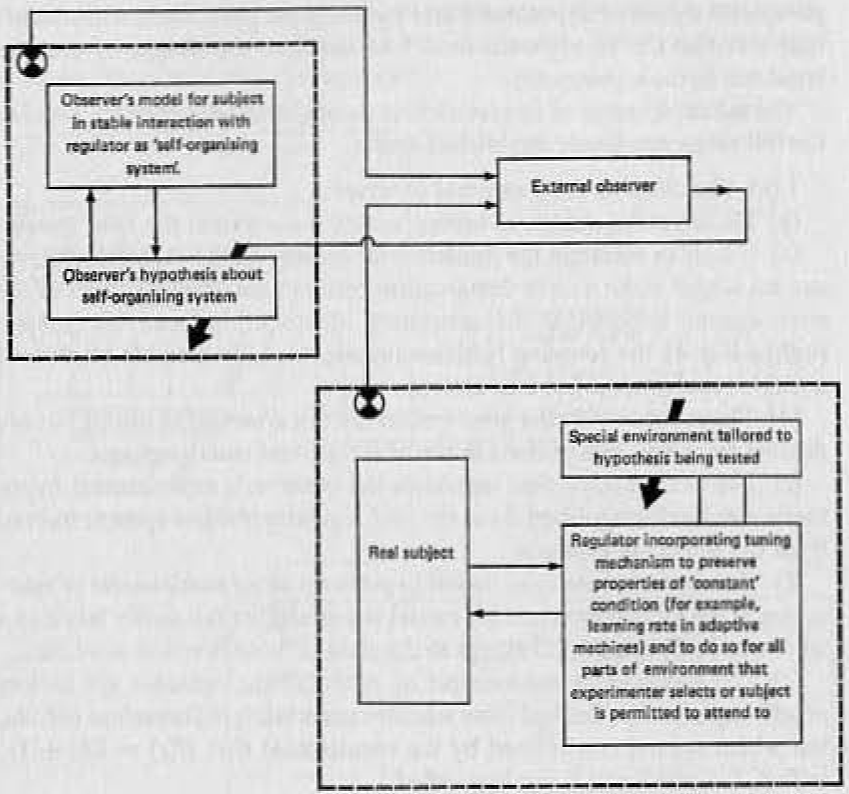


Figure 33 A special case of Figure 32 in which the 'tuning' mechanism is specified as a part of a regulator and its model. This is the adaptive system of Chapter 8.

component of the regulator (as in Chapter 8) and are represented, isomorphically, in the model (as in Chapter 6). The construction also indicates the restrictions imposed upon 'tuning' in general; namely, it must be open to instrumentation and representation within the calculus of finite-function machines. In turn, this limitation imposes restrictions upon the size and structure of the (large) universe $\mathscr{U}$ in which the phenomena of interest can be observed.

6. The point can be usefully rephrased, using a distinction due to Hesse (1963) discussed in Pask (1963, 1971a). The distinction demarcates analogies *with* and *without* 'neutral' analogical properties; that is, properties of undetermined relevance, at some instant.

A standard analogy is a relation, in the simplest case, of the form 'A is to B as C is to D' where A and B belong to one universe (U_X say) but C and

D belong to another universe U_Y. There is a function, or in general a relation, F, carrying elements (such as A) in its domain, into elements B in its co-domain; likewise a relation, G, carrying elements C in its domain, into elements D in its co-domain. The analogy exists in so far as there is a relation between F (in the universe U_X) and the relation G (in the universe U_Y). The analogy is *strict* if this relation is an *isomorphism* or a homomorphism. The *analogical* universe is specified by choosing an indefinite sequence of properties P_X of U_X and P_Y of U_Y; the analogy has no *neutral* properties and is *closed* if for all P_X or P_Y that are chosen, each one either is or is not relevant to the analogy. In the former case a particular pair P_{X_0}, P_{Y_0} is characteristic of the analogical universe. If this is not possible at a given instant (though it may become possible to effect the discrimination later) F and G holds only in part of the analogical universe (there are regions of fuzzy specification); the analogy is only *locally* strict and not always so.

Let U_X and U_Y represent the 'model' universe and the 'experimental' universe so that $\mathscr{U}$ is the pair $\langle U_X, U_Y\rangle$ related by the analogy underlying the theory. The limitation upon 'tuning' is that this analogy remains always strict, or 'tuning' is carried out to secure this result.

2.3 The Hinterland between Relativistic and Reflective Theories The difficulties caused (from the classical standpoint) by activities like attention directing and learning can be ascribed to a misbegotten choice of the unit for observation. Instead of regarding the subject as a system (such as a system in classical physics), with input and output it would be possible to regard *him* (now *him* quite seriously though) as a unit of interpretation; a symbolic structure which is usually executed in a non-localised processor. Really, it is a matter of approach. For, in either case, the sentient character of a human being (and other systems as well) must be acknowledged. Either his integrity as a symbolic (interpretive, innovative, strategy constructing) entity can be taken as *fundamental* and the processor responsible recognised after that; or else, using the approach so far adopted, the processor can be regarded as the fundamental unit and the symbolic processes it executes can be ladened onto it.

All the relativistic expedients may be viewed quite legitimately, as means to isolate an *observable* core of mentation within a capsule of processor. In each case, the processor has expanded (to keep symbolic events observable, the processor is distributed across the components of an interactive system bounded by $\mathscr{U}$). But the limits imposed upon $\mathscr{U}$ in section 2.2(5) are, for many purposes, crippling. They disallow the investigation of lengthy or educationally interesting stretches of learning, for example, or any kind of non-trivial creativity. So, on balance, it is provident to accept the obvious (that is, the obvious to someone primarily concerned with these psychological events rather than someone concerned with human neurophysiology) and to recognise that man is first and foremost a symbolic entity.

Something is gained by doing so. Under this interpretation it is clear that the constant conditions of an experiment rest upon certain norms which the subject accepts if he engages in an experimental contract (to take part in the experimental situation at all and to obey its rules). Moreover, in such a situation he adopts a role and this role (student, respondent in an interview, decision-maker or whatever) *may* be taken as the fundamental unit under scrutiny. On the last point it is possible, at this stage, to remain uncommitted. But if the investigation is to proceed beyond the limits shown under number 5 of section 2.2, then the following characterisations are essential.

(a) The subject speaks a language: either he is formally given a language, the object language of the experiment, or he uses the symbol system actually provided *as though it is a language*, a fact that is empirically manifest, for example, as the 'participant interaction' of Chapter 7. The object language has a genuine interpretation and it is a command and question language, not just a formal language.

(b) The special environment is symbolic and constitutes a description, in terms of this experimental object language, of relations or topics that are knowable. It may be extremely complex (Chapter 11).

(c) The experimental constancy is maintained by a normative scheme, the experimental contract. As one party to this contract the subject accepts and interprets a *role* which the regulator must also interpret. For its part, the regulator cooperates with the subject (through the experimental language) and makes it *possible* for him to keep the contract he has agreed to keep. The notion of 'experimental construct' is exemplified in Chapter 7, section 7 (notably in subsection (e) on p. 190 and p. 191).

It follows, of course, that an observer's hypotheses are no longer necessarily causal and that though the observer may refer to the subject/regulator system as *it* he must recognise, with increasing cogency as more interesting hypotheses are tested, that he is looking at a conversation in which the subject and the regulator regard one another as 'you' and 'I' (not, as before, in which the subject regards the regulator as *it*). In the most restrictive forms of theory, these points are not too obtrusive. One restricted form is shown in Fig. 34; the features of which are listed below.

1. (a) The observer is an external observer.

(b) He can *only* order events at the cost of restricting his enquiry and he is not in a position to order all events taking place in the joint system.

(c) The special environment is symbolic and is interpretable both in the object language of the experiment and the observational metalanguage.

(d) Events (transactions) in the system are only distinct from transactions

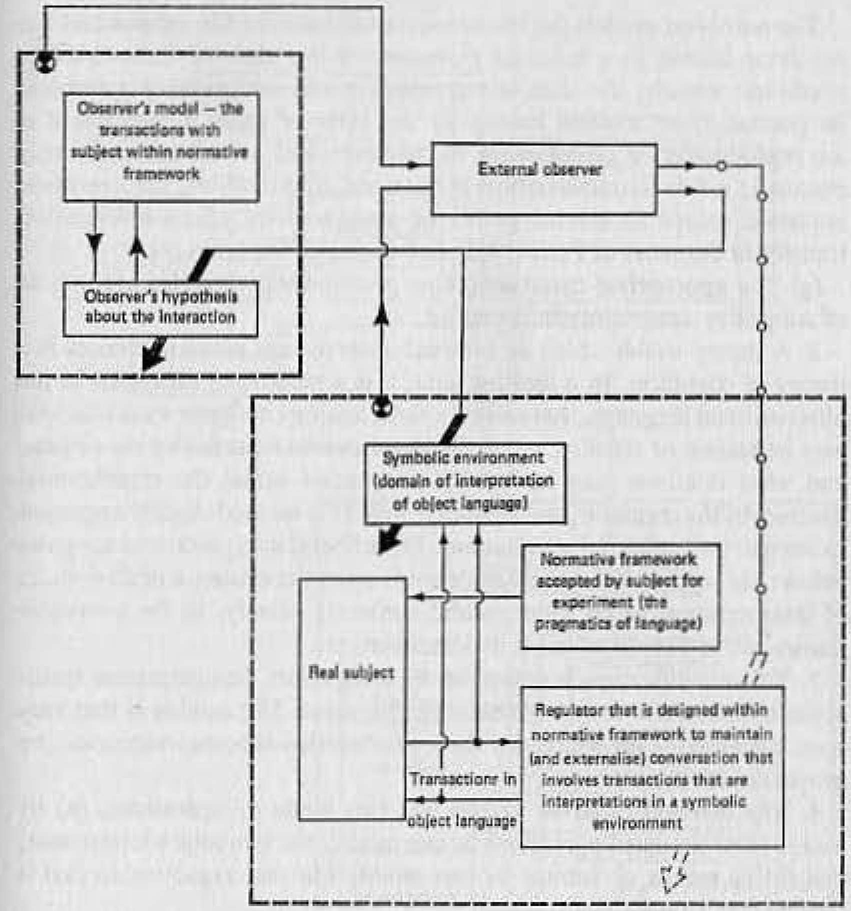


Figure 34 Model identified in a normative framework. Only very limited 'tuning' operations are permissible if the observer is to retain his external status.

in the observational metalanguage in so far as usage of the object language is restricted for this purpose.

(e) The special (symbolic) environment that embodies the observer's experimental hypothesis is distinguished as the topic of a conversation which the regulator and the subject discuss.

(f) The observer entertains hypotheses about relations that may be brought about by the execution of procedures and the topics that may be known by dint of executing other procedures. Some hypotheses concern knowable relations and others concern how relations are known, reconstructed, and satisfied.

The *narrowest* models for the conversation between the subject and the regulator belong to a minimal extension of the class of finite-function machines; namely, the class of reproductive and evolutionary automata. In *general*, these models belong to the class of fuzzy algorithms that are reproductive or evolutionary in character and are also concurrently executed; a full characterisation is obtained by fuzzifying the execution sequence except at special points of synchronicity where information transfer in the sense of Petri (1965) and Pask (1973b) takes place.

(g) The appropriate macroscopic or probabilistic variables are indices of subjective uncertainty/information.

2. A theory *within* which an external observer can make inferences is a theory of cognition. In a limiting case, it is a metaphor, expressed in the observational language, that asserts a strict analogy between what relations *may* be known or satisfied *within* the experimental contract by the subject, and what relations may be known or satisfied *within* the experimental contract by the regulator, and similarly, any of the methods legally employed to learn or bring about these relations. More liberal interpretations are given below; but tenure of *this* analogy depends upon the existence of its domain of interpretation as an experimental contract; namely, as the *normative framework* in Fig. 34 which is its identification.

3. This identification is preserved by a regulator that maintains transactions which refer to the symbolic environment. The trouble is that very soon the analogy on which the theory is founded becomes vaccuous; no properties are relevant.

4. The external observer carries out two kinds of operations, (a) by means of unrestricted operations he can modify the symbolic environment, and (b) by means of 'tuning' he can modify the real regulator so that it secures stable interaction.

The 'tuning' operations are very restricted if the conditions of section 2 above are to be preserved. A special case is shown in Fig. 35 in which the model with components $E(n)$ and $I(n)$ are those discussed in Chapter 6; similarly, the real regulator is characterised by an 'internal model' $E(n)$. If used as a process for designing effective regulators (for example, to maximise learning rate or some other property of the interaction) and if confined to the trivial symbolic environments of Chapter 6 (one coding relation $\mathscr{R}$ or two alternating relations $\mathscr{R}_1$ and $\mathscr{R}_2$) the arrangement in Fig. 35 degenerates into Fig. 34 even though the system is given a symbolic interpretation. However, if the learning model is equipped, as shown, with a list of learning strategies employed by subjects when exploring a larger symbolic environment (the *data table* in Fig. 35) the system, though still highly restricted, is not positively degenerate.

5. If more liberal 'tuning' operations are permitted in Fig. 34 (and if,

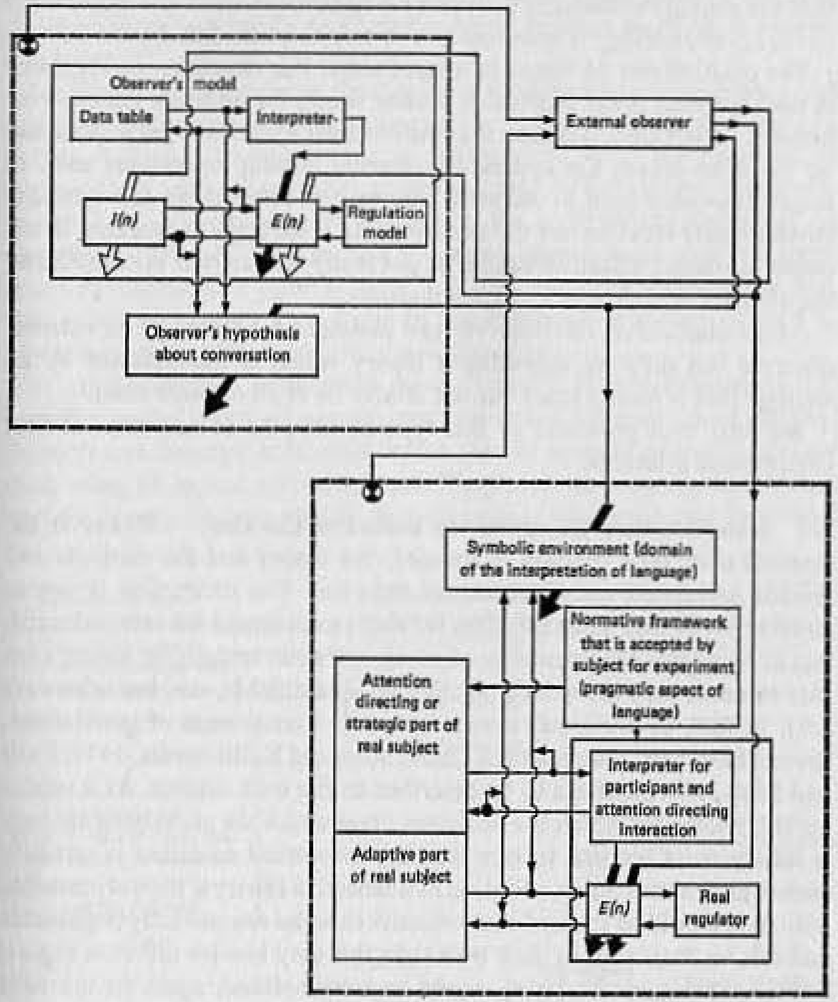


Figure 35 A special case of Figure 34 detailing one legitimate 'tuning' operation, the adjustment of $E(n)$ so that learning rate is maximised. A computer program model for $I(n)$ (of the type described in Chapter 6) is run predictively to find an effective design for $E(n)$, which is then introduced into the real regulator. The data table is a list of observed strategies.

correspondingly, the *data table* is replaced by a competent strategy producer) then either the normative framework is disrupted (the experimental contract is broken) or the transactions extend into the symbolic environment, which must be defined so that it accommodates them, in such a way that the analogy underlying the theory is not *always* strict but only *locally* strict, i.e. the analogy is *open* and not closed, as under 6 in section 2.2.

The position can be stated in several ways. For example, the regulator in such a system could be another human being; for instance, the observer himself. In that case, it is clear that the observer would be a *participant*. But by the same token, the apparently external 'tuning' operations used to secure constancy (and to maintain transactions focused on the symbolic environment) also convert the observer into a participant speaking in the object language. Effective 'tuning' is not really an *external* operation and the observer is no longer an *external* observer.

As an alternative, the observer can maintain his status as an *external* observer but only by tolerating a theory which is underpinned by an analogy that is *locally* strict but not *always* or at all instants strict.

We deal with problems of this type in Chapter 11 and offer several compromise solutions.

2.4 Transition from Relativistic to Reflective Theories Whenever the external observer's position is abraded, the theory and the methods and models associated with it, becomes *reflective*. The distinction is not as clearcut in the case when the observer elects to maintain his external status but to relinquish the possibility of an always strict underlying analogy (so that certain events are strictly measurable, quantifiable, etc., but others are not). In fact, as previously asserted, there is a continuum of possibilities. Several have been instrumented (Pask, Scott and Kallikourdis, 1973; Pask and Scott, 1973) and are to be described in the next volume. As a rule, a point is reached at which the occasions upon which the underlying analogy is locally strict become so rare that the theoretical construct is virtually useless and, at this point, adoption of a reflective theory is the only sensible option. It should be stressed that reflective theories are perfectly respectable and can be dealt with in their own right but they involve different experimental techniques (to be discussed and exemplified, again in the next volume) and have little resemblance to classical theories.

For future reference, it is worth noting here that a significant transition takes place at the point when the structure of the symbolic environment (Fig. 34) comes under the control of transactions in the joint system, i.e. when the dialogue is no longer confined to a topic (however broad) but the topic evolves, under the influence of the dialogue.

2.5 Other Examples The following examples are culled from Pask (1973a) where they are much more thoroughly discussed. They serve a dual purpose; to give some impression of the effects of the 'significant transition' mentioned in the last section upon models of repeatable real life events and

to dispel the impression (which may easily have been given) that these considerations only apply to human subjects, human learning, cognition and so on.

In psychology, the bias is shifting from behaviouristic to symbolic and cognitive interpretations; for example, the trend exhibited in Scandura (1970). A similar transformation is taking place in ethology and social anthropology. One manifestation of the trend is that events previously regarded as epiphenomena thrown up by the complex gurgitation of an underlying particulate process are conceived nowadays as symbolic regulatory mechanisms; a point of view that goes from the whole to the part (like *relativism*) rather than going in the traditional direction of part to whole (one salient feature of the classical theory).

In his discussion of population density control, Wynne Edwards (1963) puts forward a convincing argument that many phenomena of display, mimicry and directive behaviour have a specific signalling function, rather than being biological epiphenomena. They mediate communication in a specific density control system. For example, the singing and territorial

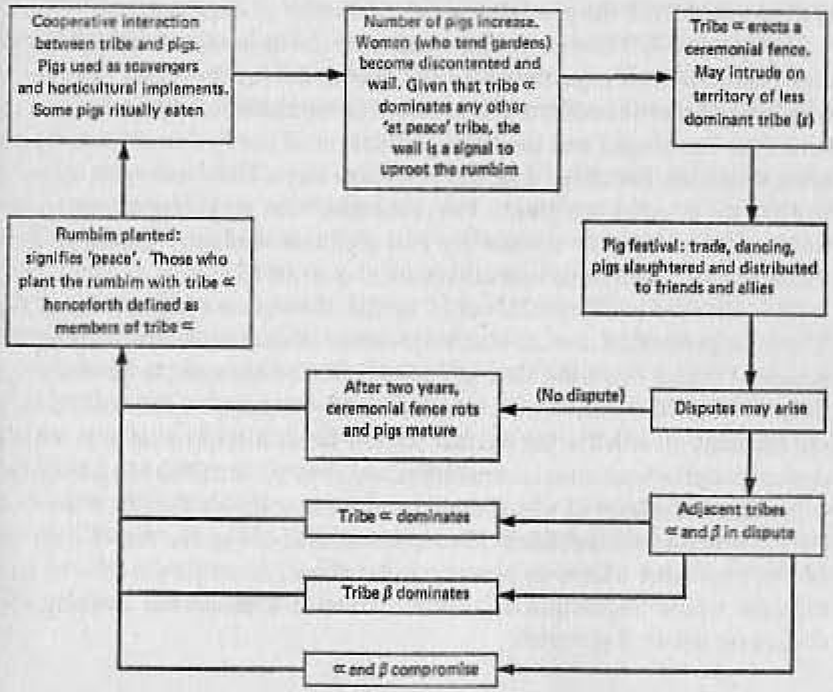


Figure 36 The ritual cycle of interaction between the Tsembega tribe and the local population of partly domesticated pigs.

excursion of male birds prior to mating, serves (a) to indicate male density to other male birds, and (b) to provide information about the resources available in a given habitat. From (a) and (b), an individual male is provided with a difference-signal relevant to density control. Notably, this difference-signal is a property of the local group of animals and is sometimes generated by a ritual colligation or 'epideictic' display.

Within a group, there is an established dominance hierarchy, wherein any male individual has a level. The hierarchy is established by convention in some cases, and by stress-mediated hormonal mechanisms in others. The lowly individuals, 'excluded males', are not allowed to mate and breed (for the hierarchy sets up a prescriptive norm in this respect). According to Wynne Edwards' hypothesis, the relative number of 'excluded males' in the local group is determined by the intensity of the difference-signal which thereby serves as a feedback to establish the group goal of securing a reproduction rate (hence, a subsequent population number) that fits the resources of the environment. The group goal is, in many cases, contrary to the innate goals of the individuals; for example, reduction of various sex drives. Thus, the model involves a logical hierarchy of goals to be sharply distinguished from the partial-ordering hierarchy of dominance.

Rapaport (1967) makes similar comments about human density control and exemplifies his argument by a specific model for the ritual regulation which maintains a condition of mutualism between the local population of a tribe (the Tsembega) and the local population of partly domesticated pigs upon which the Tsembega depend, in various ways. The ritual cycle is easily written as a serial program. For example, one very elegant program was written as a term exercise by two graduate students, Cartledge and Rezac (1970).[2] The basic operations are shown in Fig. 36.

As in the case of Wynne Edwards' model, it involves a logical hierarchy. There is a perceptual level at which conditions of the environment are interpreted as acting upon the state of individuals (for example, to produce the signal of female dissatisfaction which is chiefly responsible for determining the moment at which a pig festival starts). Next, there is a level at which conventional or traditional norms are established in the milieu of a particular tribe (this is the level at which the flow chart is written). Finally, there is a level at which a tribe is defined. The terse comment of Fig. 36, 'A man belongs to the tribe with whom he plants the Rumbim', conceals the crucial fact that the whole 'regulation algorithm' contains a clause for defining the domain in which it operates.

2. Robinson (1973) is responsible for a more elaborate program which simulates the ritual regulation process responsible for maintaining a deviant group in urban society.

Bateson (1956, 1972) represents the cultural 'double bind' (a trapping state induced by misperception and misinterpretation at a higher or meta level in a conversation) as the mechanism whereby the conventionally encoded organisation of a culture is modified. It is argued that the maintenance of homeostasis in a social group depends upon the existence of a suitably encoded organisation, such as the Tsembega ritual, in which conflicting tendencies trigger off compensatory reactions. This organisation personifies the culture and is essential for its physical integrity. If a pair of cultures having incompatible organisations are juxtaposed, a 'double bind' situation occurs, as a result of which the cultural organisation pattern is necessarily modified. Bateson has also considered an analogous process operating in the genesis of familial schizophrenia. Once again, the model is accommodated in an hierarchical structure and this structure is mandatory.

These organisations are studied in an essentially relativistic manner by ethologists, anthropologists and psychiatrists; all of whom act (overtly or not) as regulators compensating for their intrusion. The observers in question use sociological variates, locally tenable political or conventional structures, or themselves, pure and simple, to secure this end. In pursuit of regulation, liberal use is made of models; either intellectual, or mathematical, or computer programmed. To a large extent it is possible for a professional observer to maintain the posture of an external observer; that is, to segregate a mental function of external observer from the mental function of regulator.

This orientation becomes impracticable at the moment when the ritual structure is modified, autonomously; for example, in the case of density control, by genetic or mass aggregation effects; in the ritual cycle, by resolving which tribe a human being belongs to, and in a double bind situation, throughout. The concommitant event of this transition in the abstract, is an instruction from within the model, to rewrite the model or to change its domain. In other words, execution of this instruction, in a stable manner, places the model fairly and squarely in the category of reproductive automata (section 2.3(f)) and usually in the class of evolutionary automata executed concurrently rather than serially.

Attempts to match the evolution to reality (or vice versa, to prescribe the mode of evolution at the outset), entail participation, without which the process looks merely random. By virtue of participation, however, the theory entertained by the observer becomes reflective.

5 Learning Models

The next two chapters review two categories of learning model. The first category, discussed in the body of the present chapter, consists of models that were made when behaviourism, of rather a strict and narrow variety, was rampant. As a result, they are coloured by this philosophy. For all that, it was necessary to introduce mechanisms responsible, non-trivially, for anticipation, expectation, and also for the type of cooperative interaction that Grey Walter (1953) so delightfully imaged with a gaggle of adaptive 'tortoises' and that the ethologists, like Tinbergen (1953) and Lorenz (1952), adopted as an essential part of the scientific stock in hand.

As a result, the models did not behave in a narrowly behaviouristic fashion. In fact, they exhibited many of the tricks (fuzzy computation, for example) that are introduced more deliberately in our later work. But an unfortunate consequence of the tradition in which they were spawned was that they did not contain the hierarchical structures necessary to expose these tricks very clearly to view.

The category of learning models described in the next chapter do have an hierarchical structure. Although they represent rather simple operations, compared to more recent models made by our group, they are cognitive systems. However, they were built in the spirit of symbol processing, a more modern dogma beset by quite different constraints; for example, list-search paradigms and serial processing. Because of that and also their smallish size, they are unable to image many of the processes which, today, seem most important.

Neither category of model embodies my current view of mentation, though both contributed to it. The reader is asked to accept the fact that it is possible to model very different and more interesting facets of cognition; either in the metier of artificial intelligence (Winston, 1970; Winograd, 1972) or in the more experimental (man-machine interaction) terms pursued in my own laboratory. He is also asked to leave this aspect of modelling as pending; it receives due attention, analysis, and discussion in the next volume.

1 The EUCRATES Machines

Inspired by Wittgenstein's insistence upon the primacy of relations, Bailey, McKinnon Wood and I built a machine to learn the relationships between events in its environment. In parallel with this project, we devoted some effort to teaching the machine a subset of useful or desirable relationships and concluded that a further mechanism was needed for that purpose. This mechanism was embodied in another piece of hardware and C. E. G. Bailey christened the learning system 'EUCRATES' (the original sorcerer's apprentice).

There were several EUCRATES systems. The first of these was commissioned by the Solartron Electronic Group and was exhibited in the Physical Society Exhibition of 1955 in London. The finally engineered version is an analogue computer for use in designing adaptive teaching machines by examining their interaction with a simulated student, and was completed by Solartron in 1960. These systems were mentioned in an earlier book (Pask, 1961a) where both of them are illustrated (Plate 1(i) is the production machine; Plate 1(iii) the original machine catalysing the growth of a system of metallic dendrites), but the account of them is terse and incomplete.

1.1 Philosophy Basically, EUCRATES is a reflex 'learner' which is sensitive to reward and brings about rewarded relations between the stimuli it receives and the responses it produces. In this respect, its design owes a great deal to the pioneering work of Uttley (1959) with conditional probability machines. In addition, however, EUCRATES contains a mechanism interpretable as an expectancy mechanism and another which simulates response anticipation (or, at any rate, something akin to Thorpe's (1956) specific action potential). Further, EUCRATES is designed around concepts to do with 'curiosity' and 'attention' (it has a primitive curiosity system and there is a sense in which its attention must be occupied). Briefly, EUCRATES is a machine which looks for problems and, having found one, is impelled to learn how to solve it. This property is so basic to the design of the system that if the machine is placed in a situation that prevents the exercise of 'curiosity', then it becomes functionless.

The action of the machine can be analysed in several ways. We might, for example, adopt the elegant techniques developed by Steinbuch (1961) in connection with his learning matrices (which are special types of conditional probability machine). For the present purpose, however, the activity can be analysed quite satisfactorily by saying what the machine does and how it does it, without mathematics.

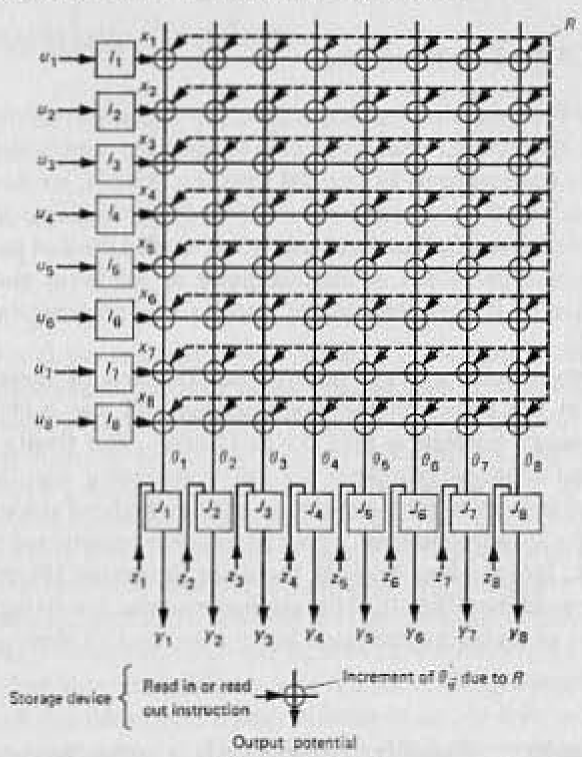


Figure 37 The plan of the machine. Each storage device retains the value of a biassing variable θ_{ij}.

1.2 Overall Plan The layout of the machine is shown in Fig. 37. It consists of a stimulus unit with components $I_1, \ldots, I_8$, an output or response unit with components $J_1, \ldots, J_8$ and an array of 8 × 8 analogue storage devices (retaining the values of variables θ_{ij}, $i = 1,2, \ldots, 8$ and $j = 1,2, \ldots, 8$). In the first machine to be constructed, this 'array' consisted of sixty-four capacitors with read-in, read-out and lock facilities provided by clamp diodes. (Read-out, even into a high impedance, appreciably modifies the state of an individual store.) In later machines, this rather crude arrangement was replaced by sixty-four capacitors, each of which was associated with a Miller Integrator, amplifier and impulse circuits for read-in, so that the state of a store is substantially unchanged by an interrogation.

The circuitry of the stimulus unit is shown in Fig. 38. The output of the unit is a vector, X, of eight binary variables x_i, each of which is associated with one of the components I_i. Any of the I_i consists of a local storage element, an averaging circuit, a current limiter and a lock circuit. The current limiters are cross coupled so that an increase in the current passing

through the ith circuit inhibits the action of all the remaining circuits. If a stimulus 'search' instruction is delivered (as a positive potential) to the common cathode valve, then a decision process is instituted as a result of which the current passing through one or more of the I_i exceeds the predetermined value built into the limiter. When the limit is exceeded by the ith circuit, $x_i = 1$. The lock circuit holds this condition (irrespective of the state of the machine) until a responsive event $y_j = 1$ takes place (the process whereby $y_j = 1$ is discussed below) and it delivers a 'response search' instruction to the response unit. Next, the event $x_i = 1$ delivers a negative, inhibitory charge to the ith local storage capacitor and a small positive charge to each of the remaining local storage capacitors. Then, after a short delay of δt, this event cancels the stimulus search instruction so that the decision process terminates. Finally, the event $x_i = 1$ 'opens' a row of the 8 × 8 storage array so that a vector θ_j of values θ_{ij} is delivered as part of the input to the response unit.

The outcome of the decision process depends upon the values assumed by the variables u_i in a vector U (the external input to the I) and the values assumed by eight variables μ_i (derived, as an internal input, from the storage circuits). If all of the $u_i = 0$, the action of the stimulus unit is fairly straightforward. The inhibitory cross coupling between the I_i fosters (though it does not strictly enforce) the condition that no more than one $x_i = 1$ at once. Suppose that the stimulus search instruction, cancelled by the event $x_i = 1$ is again instituted (by a mechanism to be described) and that x_i is set equal to 0. The decision process now searches for and produces the least 'likely' or the most 'unusual' event.

These conditions no longer hold true if the external input is energised; for example, it is possible to make several of the u_i positive valued and to secure several x_i conjointly equal to 1. Even so, the external input, U, still acts upon an underlying search and memory process of the sort described. In particular, because of the local storage feedback signal, the machine is prone to reject repeated and familiar inputs. Conversely, the value of u_i required to elicit the event $x_i = 1$, increases with repetition. In other words, the stimulus unit exhibits habituation.

The response unit, also shown in Fig. 38, resembles the stimulus unit. It differs in the following two respects: (a) in addition to the external input Z (analogous to U), and the internal inputs η_j (analogous to the μ_i), the J_j averager receives an input θ_{ij}, and (b) the lock circuit of a component I_i is replaced, in a component J_j, by a circuit that holds the condition $y_j = 1$, (analogous to $x_i = 1$) for an interval Δt. With these comments it will be evident that on receipt of a response search instruction (analogous to the stimulus search instruction) a decision process is set up in the response unit which leads to one or more events of the form $y_j = 1$. However, even if

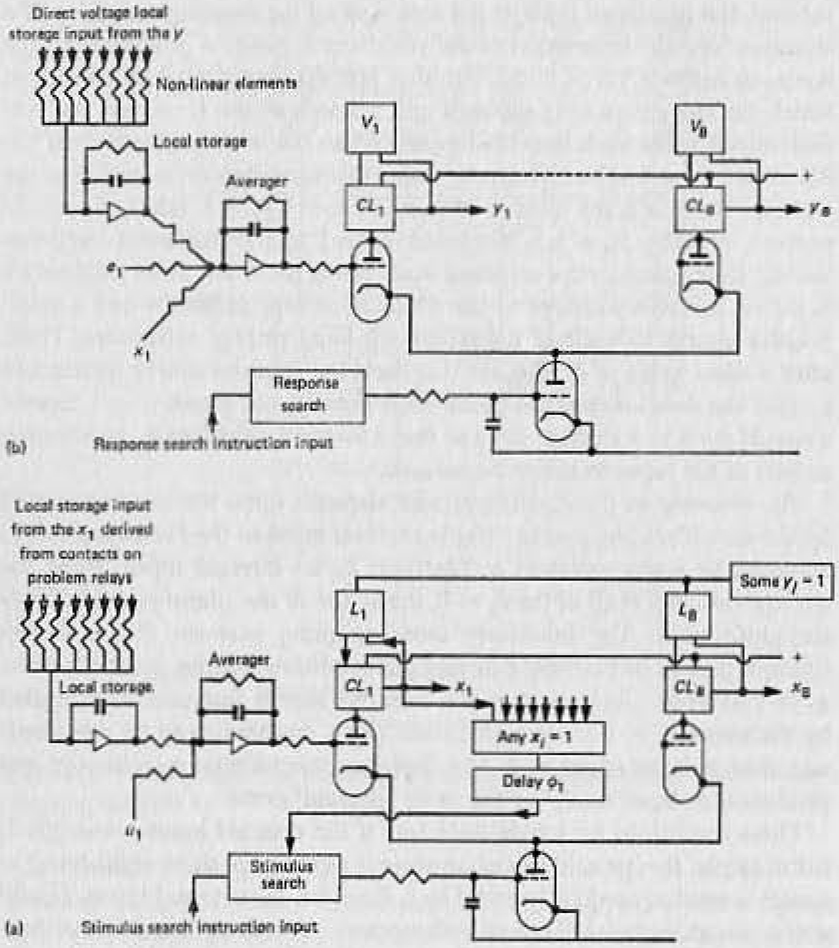


Figure 38 Circuits of the stimulus and response units: (*a*) stimulus unit. Component 1 shown in full. *CL*, current limiter; *L*, lock relays. (*b*) Response unit. Component 1 shown in full. *CL*, current limiter; *V*, relay circuit for retaining event, $y_j = 1$, over an interval of Δt.

the variables z_j in Z are zero valued, this process is biassed by an input from the 8 × 8 array of Fig. 37.

In considering the form of the bias, notice that the feedback to the local storage elements in the response unit is derived directly from the current limiting circuits through non-linear components. The time constants of these circuits are short compared with the time constant of the stimulus unit storage elements, and the feedback acts within the compass of a single

decision process. The effect of a high valued input θ_j at the *j*th component is to drive this circuit into the state $y_j = 1$. However, if the input does not do so, within a given interval, its action is inhibited by an increasingly negative η_j from the local storage circuit output. As a result of this, the θ_{ij} biassing effect is not unique, for example, the event $y_j = 1$ may occur either because θ_{ij} is higher than the other entries in the biassing vector θ_j, or because there is an ordering of the form $\theta_{il} = \theta_{im} > \theta_{ij}, \ldots$ when the tendencies to produce $y_l = 1$ and $y_m = 1$ compete and are inhibited by increasing η_l and η_m so that ultimately $y_j = 1$. A large number of competitive impasses are resolved in this fashion.

The response search instruction is delivered whenever $x_i = 1$; further, the particular row of the 8 × 8 array presented as a biassing input to the response unit depends upon the particular event x_i (this event 'opens' the row of storage devices and presents a vector θ_i to the response unit); hence, the array serves to couple the stimulus and the response units. To close the cycle, note that the stimulus search instruction is delivered at the termination of any event $y_j = 1$ (that is, Δt seconds after such an event has occurred).[1]

The cyclic action gives rise to sequences of events $x_i = 1, y_j = 1$, which overlap by Δt seconds to form conjoint events $\langle x_i, y_j \rangle = \langle 1,1 \rangle$. The values θ_{ij} retained by the 8 × 8 storage device in the 8 × 8 array depend upon the value assumed by a variable, R, at the moment when a relevant conjoint event occurs. Thus, at trial n in a sequence, if a single event $\langle x_i, y_j \rangle = \langle 1,1 \rangle$ occurs, the value of θ_{ij} is incremented or decremented by an amount proportional to Δt multiplied by the difference between R and θ_{ij}.

Conversely, the values of all of the entries in the vector θ_i are decremented towards 0 by an amount proportional to the values of these entries and the interval between the event $x_i = 1$ and event $y_j = 1$. This interval is later called the 'anticipation latency'. The variable R (which is later interpreted as a 'reinforcement variable') can assume values in the interval $-1, 0, +1$ and a null value $\emptyset$, if $R(n) = \emptyset$, then $\Delta\theta(n) = 0$.

The incrementation process is more complicated if a pair of events $x_i = 1$, $x_k = 1$, occur simultaneously or if they overlap before some $y_j = 1$. Suppose, to begin with, that $x_i = 1$ and $x_k = 1$ and $y_j = 1$ at the *n*th trial and that $R(n) = \emptyset$. In this case, if $\theta_{ij} > \theta_{kj}$ then the entry θ_{ij} receives an increment $\Delta_1\theta(n) \approx c(\theta_{ij} - \theta_{kj})\Delta t$. If $R(n)$ is not equal to $\emptyset$, then the total incrementation is complex and consists of a component $\Delta_2\theta(n)$, the 'extrinsic reinforcement' and a component $\Delta_1\theta(n)$, the 'intrinsic reinforcement'. So far as decrementation is concerned, if $x_i = 1$ and

1. If the θ_{ij} are constant valued, this coupling is such that the machine acts as a stochastic matrix multiplier. When, as below, the θ_{ij} are variables, the entries in the stochastic matrix are functions of the performance of the machine.

$x_k = 1$ all of the entries in θ_i and all of the entries in θ_k, including θ_{ij} and θ_{ik}, are reduced in the direction of zero in proportion to their several values and the length of the anticipation latency. Because of this, and because of the active characteristics of the stimulus and response units, the maintenance of a pattern of θ_{ij} values entails some sort of dynamic organisation.

1.3 The System Let us talk about the system anthropomorphically. On receipt of a stimulus search instruction, the stimulus unit 'expects' to receive a 'problem' (some non-zero value of X). The problem it does receive will be biassed by 'evidence' U, from its environment but, even if there is no evidence, some problem will ultimately be posed (as a result of the stimulus unit decision process). This state of 'expectancy' gives rise (when the problem is posed) to a state of 'anticipation' regarding a solution to this problem. A solution is *denoted by* some non-zero value of Y, and it is achieved by the response unit decision process, instituted through the response search instruction by some $x_i = 1$. The solution depends, even in the absence of the external input Z, upon the θ_{ij} values in those rows of the 8×8 array that are selected by the particular problem. The state of 'anticipation' terminates after a solution is *designated* and its termination gives rise to a further state of problem 'expectancy'.

In all this, the θ_{ij} values play a crucial part by determining what solutions will be given to various problems. An assumption that values of R represent values of some reward or desirable commodity, and that the machine would like to solve problems in such a way as to maximise the average value of R, is effectively built into the design. For, as a result of the process just described, the machine 'learns' to solve problems in a manner that is compatible with maximising the average value of R (the θ_{ij} assume a pattern that leads to this result, providing that specific and consistent values of R are associated with specific problem solution pairs).

1.4 Modification of Habituation Mechanism The habituation of the machine requires further comment since it is, in some respects, unrealistic. If a stimulus is repeated, the strength of stimulus required to elicit the problem state $x_i = 1$ increases. This is more or less as it should be, provided that $x_i = 1$ does not lead to a reinforced response (a significant event). If the result of a stimulus is significant (if $\langle x_i, y_j \rangle = \langle 1, 1 \rangle$ is reinforced) then the present machine performs as it would do if the stimulus were not significant (at least, it does so with respect to the expectancy process; the anticipation latency, the interval occupied by the response unit decision process, is reduced). This is not altogether satisfactory from a biological point of view because in a real organism habituation with reference to significant stimuli is suppressed or nullified. The model works the right way

round for the anticipation latency; but it is perverse in respect of the expectancy latency for significant events.

The right characteristic can be secured in several ways. It was in fact, obtained by modifying the local storage feedback to the stimulus unit so that, instead of a simple inhibitory signal, this feedback became, at the nth trial and for the event $\langle x_i, y_j \rangle = \langle 1, 1 \rangle$ a quantity $1 - \theta_{ij}(n)$ where $\theta_{ij}(n)$ is the quantity delivered as the internal input to the jth component of the response unit. If the problem event induced by a stimulus is not significant, $1 - \theta_{ij}(n)$ is high (and habituation takes place as before). If the problem event is significant, then $1 - \theta_{ij}(n)$ is low valued and habituation is suppressed. This arrangement gives the machine the required characteristics, but it does so 'unfairly'. Additions of this sort, constitute an attempt to represent, in a primarily single level model, the organisation of a many level, and hierarchical structure.

1.5 Experimental Facilities For experimental purposes the EUCRATES 'learner' is connected to a console from which the experimenter can act upon and observe the main variables as well as receiving an 'illicit' (from one stance) view of the interior via a θ_{ij} display. As shown in Fig. 39 the learner may either be manipulated directly or coupled to a EUCRATES teacher (not yet described, but essentially the same as the learner apart from a full representation of the task relation to be learned and circuits for computing criteria of performance, i.e. of the extent to which behaviours satisfy the task relation).

In addition, the teacher can be coupled to a real-life student so that comparative observation is possible (student/EUCRATES teaching machine in contrast to EUCRATES learner/EUCRATES teaching machine). Though important for the main application (design of smaller, degenerate, teaching machines) this facility is not discussed.

2 Ways of Looking at the System

The EUCRATES learner can be viewed in several ways. During the late 1950s it was fashionable to take a strongly behaviouristic point of view and it will be instructive to start with this image in mind and to modify it as needs be.

2.1 The Machine Regarded as a 'Black Box' Consider the machine as a 'black box', in Ashby's sense. It has an input U, which may be interpreted as a stimulus, perhaps complex in form, which gives rise, internally (and beyond the behaviourist image) to a problem vector X. It has an output, Y, which may be interpreted as a response vector designating an hypothetical

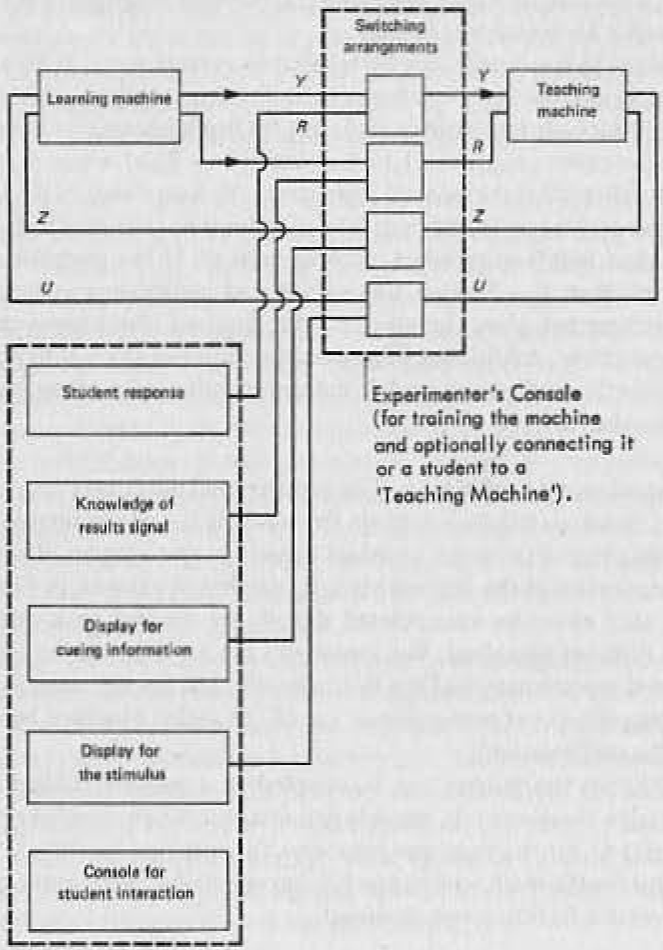


Figure 39 Switching arrangements for connecting either the learning machine or a real-life student into the teaching machine.

solution naming vector, if all of the y_j are brought out to external connections. But, in many arrangements of the system, only some of the y_j are brought out. In any case, there is a great deal of difference between a solution (the response *process*) and a response which may select in context amongst possible solution *names*.

The machine has an input R, which, given a correspondence between stimuli U and responses Y, acts as a reinforcement variable. The machine

has an input Z which manifestly guides it in selecting its response (we can prevent the machines from making response j by holding the jth component of Z negative-valued and force it to make response j by holding the jth component of Z positive-valued). At least this is true in *most* conditions; to be *certain* of doing so, it would be necessary to delve inside the black box. Hence, Z is identified with a cueing vector.

If this identification is adopted, it is convenient to relabel a vector U, in which one component is positive and the rest are zero valued by a *letter* (calling such a vector 'stimulus a' = $U = \langle 1,0,0,0,0,0,0, \rangle$ or 'stimulus b' = $U = \langle 0,1,0,0,0,0,0 \rangle$; there are eight of them in all).

Similarly, the response events $y_i = 1$ are named 'response A' = $Y = \langle 1,0,0,0,0,0,0,0 \rangle$, or 'response B' = $Y = \langle 0,1,0,0,0,0,0,0 \rangle$. We may, of course, apply more than one stimulus at once, in which case the vector U has more than one non-zero component. (Thus, the conjunction of 'stimulus a' and 'stimulus b' is the vector $U = \langle 1,1,0,0,0,0,0,0 \rangle$.) It is also possible for the machine to make more than one response at once.

At this point, it is apt to comment upon an ambiguity in the use of words like 'stimulus' and 'response'.

'Stimulus' can mean.

1. The state of the environment that is appreciable by the machine at a given trial.

2. An appreciable and separable event (the state of some part of the total environment).

If 'stimulus' is interpreted according to (1), then the several components, u_i in U, are properties of the stimulus, designating appreciable attributes. In this case, it is possible to talk about stimulus discrimination (that the machine comes to select certain properties as important and to disregard others as unimportant). However, it would not be reasonable to say that a pair of stimuli occurred simultaneously or that the machine rejected one of this pair and respected the other. (This situation might be described by saying that the machine attends to one property of the total stimulus and disregards the others.) Conversely, if definition (2) is adopted it is entirely possible to have coincident stimuli. But stimulus discrimination effects will be suppressed and will only be manifest in connection with phenomena such as habituation that are predicates of sequences of trials.

The 'stimulus a' and 'stimulus b' nomenclature opts in favour of definition (2) rather than (1). It is important to realise that this decision is not forced upon us by the machine; it is made to suit the experimental context in which the machine's behaviour is examined. The machine's behaviour could be equally well described within the framework provided by (1) or by (2). Many phenomena are the consequences of the *nomenclature* that has been selected.

Similar comments apply, more obviously perhaps, to 'response'. According to a definition that parallels (1), the machine, at any trial, makes a response which is a vector Y (to be more specific, adjoin latency information, stating when the various components y_j of Y undergo a transition $0 \rightarrow 1$). If, on the other hand, a definition akin to (2) is chosen, then the machine may make several 'responses' at a given trial and specificity is achieved by saying how and when these several responses are made. The word 'response' is itself slightly misleading (it would be more usual to say 'the machine's response has several parts—these parts of the total response are made in such and such a fashion'). As a result we shall take the liberty (commonly taken in the literature) of using the word 'response' to denote either the total configuration or the parts of this configuration, providing that the practice does not give rise to confusion.

2.2 Interpretation of the Learning Model Is this model a finite automaton, albeit a probabilistic one, i.e. is the artifact a finite-state machine? It certainly is, if regarded as a rather simple and straightforward physical object. It is, under interpretation, too; provided we subscribe to one or other of the definitions of 'stimulus' and of 'response' explained in the last section. But, however useful these constructs may be as approximations, there are good reasons for rejecting all of them.

The trouble is that the machine has several asynchronous loci of control or several asynchronous automata that act as parts; for example, the 'horse race' of the response resolution unit or even one component of it (each 'horse' in the race). Hence, strictly speaking, we cannot, in the capacity of users or model builders, regard aggregates of these asynchronous components as *having* states to begin with, just because the units *are* asynchronous. As conceded in the last paragraph, the physicist who regards the machine as a chunk of material is in a somewhat different position.

The crucial issue is independent of degree of resolution. Clearly the input and output 'states' posited earlier in section 2 are of low resolution; so are the internal states that might go with them. If the machine *could* be regarded as a finite-state machine the state specification could be refined; either by partitioning or by using behavioural histories to delineate irredundant (algebraic) internal states. But, as it is, the notion of state is undermined (deliberately so, in order to secure the anticipation and expectation properties) by the existence of asynchronous parts.

It should be emphasised that the machine is composed of asynchronous or autonomously clocked parts and is not a parallel computer such as a perceptron (where the ongoing computations, though occurring simultaneously, are centrally clocked). The crucial feature of the machine is that otherwise asynchronous computations *become* synchronous (or partially

so) by virtue of information transfer due to the global computation that the machine is currently executing; very similar entrainment phenomena occur in Kilmer and McCulloch's model for the mammalian attention directing system, SRETIC (Kilmer, McCulloch and Blum, 1969). Further the same comments apply if the model is realised as an actual machine (it has been, using the PSV modules of Barron (1968) and Gilstrap as the units for computation) rather than simulated in a digital computer with the housekeeping needed to make a serial processor mimic a concurrent process. Similar remarks apply to various mesh-like collections of components, derived from Beurle's original model (Beurle, 1954, 1959), noted in Pask (1961a), provided there is no degeneracy due to simulating concurrency as a randomisation of serial events. Here, the activity consists in the propagation of wave-like modes of activity. There is an interesting correspondence between initially asynchronous operation and incoherent (or 'random-phase' wave propagation) and partial synchronicity and coherent (or 'in-phase' wave propagation) which is tacitly employed, for example, in Longuet-Higgins and Michie's (1970) holographic model for memory.

So far as EUCRATES is concerned, the preferred interpretation is as follows. The machine was probably the first device to realise a fuzzy algorithm in the sense of Zadeh (1973) and for this reason is rescued from present-day triviality. In any fuzzy computation (Chapter 2, section 2) it is possible to insist that the (generally) fuzzy output is made to select one element from amongst a set of alternatives (using any convenient rule such as choosing a maximal or a minimal element) and, as a matter of fact, this kind of expedient is used in coupling the machine to its environment. But, as will be shown, it is neither necessary nor usual to convert fuzzy outputs into 'alternative selections' *within* the machine; one part can, and often does, accept (as a fuzzy input) the fuzzy output of another part.

Consider the response-resolution process (much the same comments apply to the stimulus-resolution process). If the machine is constrained to select one and only one response, which is counted as *the* output then the computation does select amongst alternatives (the response set). But, left on its own the machine generally selects *many* responses in various orders so that the entire output is fuzzy. This statement can be interpreted in several ways; for example, by noting that the entire output determines an element with grades of membership in several sets, which might be numericised by noting the order of the selections and the moments at which they occur. But the output can also be passed on to another part of the machine as a fuzzy output *simpliciter*, without numericisation. This occurs, for example, if the response resolution unit is back coupled to a stimulus unit when the machine interpretation of the fuzzy output (*as* a fuzzy input) depends upon the state and the clocking of the stimulus units. Concurrent computation occurs in so far as the otherwise asynchronous clocks in distinct parts of the machine are synchronised because of such an interaction.

In the experiments to be described, pains are taken to make the machine act *as though* it was a finite-state machine; for example, by external constraints that ensure that (or maximise the probability that) one and only one response will be selected. Similar precautions are used to secure a 'begin and end' ordering; that a response *follows* a stimulus. This part of the design is introduced *simply* so that an observer can synchronise events in the machine with his own stopwatch. It is important to bear in mind the fact that all such gambits involve constructs that are imposed upon the machine in order to make it behave *as though* it was a finite-state machine or (at most) a fuzzy-state automaton with numericised input and output. Nearly all of the interesting or emergent behaviours are due to the fact that the machine is not *actually* a finite-state machine and because the efforts to make it act as though it was such a thing are not and *cannot* be altogether successful.

Experimenters use similar tricks to make real organisms (human beings included) act as though they were finite-state machines, so that they can be coupled to a further finite-state machine representing the environment and described within the strict behavioural format. If these tricks were successful the experiment would be of small consequence (a point taken up in section 6). The interesting behaviours appear in so far as these tricks do not always work for real organisms any more than they do for the EUCRATES machine.

3 Learning Experiments

In a suitable environment, the machine is able to mimic many features of classical conditioning and instrumental conditioning. To make it play these tricks, we tacitly identify the device with a laboratory animal such as a pigeon or a rat.

A few preliminary remarks are needed. In the context of classical conditioning, a stimulus is partially causative (it is something that gives rise to, or helps to give rise to, an observable response). Responses would not occur without stimuli of some sort (though these may be internal stimuli or stimuli that are concealed from the observer). By way of contrast, the field of instrumental conditioning is built around the concept of an operant, that is, a response emitted by the organism autonomously. Stimuli are chiefly discriminating stimuli; events which, in Skinner's words, provide the occasion for a response but do not give rise to it. They influence and bias an autonomous response process.

The distinction, in experimental psychology, between a stimulus and a discriminating stimulus is tenuous (but the philosophical orientations of

these different fields are distinct enough to deserve respect). So far as the machine is concerned, the distinction between a stimulus and a discriminating stimulus is a matter of degree – some stimuli are more causative than others; there is an underlying autonomous activity; the hidden stimuli that produce it are not made explicit but could be invented.

3.1 The Classical Conditioning Paradigm If a stimulus is applied (say stimulus *a*) and, if this gives rise to the problem state $x_i = 1$, then the machine will emit one, or usually one, response (say response *A*). The sequence of events 'stimulus *a*, response *A*' can be interpreted as a reflex.

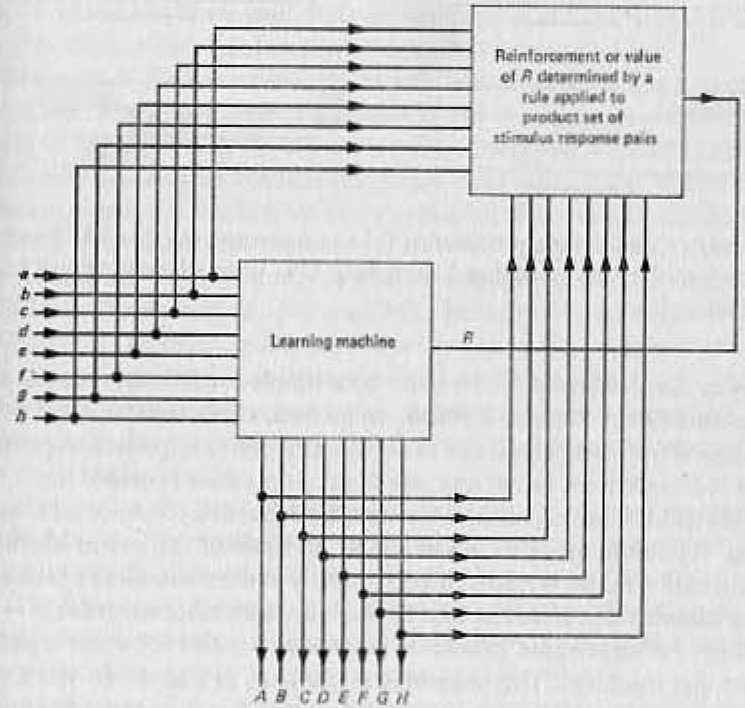


Figure 40 Reinforcement procedure.

To prepare the machine for experiments in classical conditioning, it is first necessary to embed in it certain unconditional reflexes – by hypothesis, innate and inherently reinforcing reflexes. The arrangement shown in Fig. 40 is used for this purpose. A subset *r* of outcomes or stimulus response pairs is associated with a high reinforcement value, the remaining outcomes being associated with a low value reinforcement. After a number of trials in the conditions of Fig. 40 the machine adapts so that it has a number of ingrained unconditional reflexes, the members of r. We shall assume that $a \rightarrow A$ is a member of r and will mention any other reflexes that belong to r, explicitly.

In its adapted condition, the machine demonstrates some of the simple features of classical conditioning. Although this point is not developed, the changes in response latency with conditioning also replicate the form of the latency changes that are observable in animal conditioning.

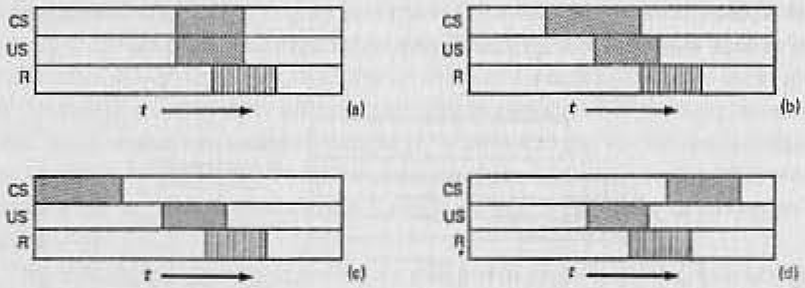


Figure 41 Conditioning procedures: (*a*) simultaneous; (*b*) delayed; (*c*) trace; (*d*) backwards. CS, conditional stimulus; US, unconditional stimulus; R, response.

Reflex Establishment To build up a simple conditional reflex choose an unconditional stimulus b which, on its own, elicits some unconditional response B (or which elicits one of several alternative responses, depending upon the conditions, but at any rate does not produce response A).

The conditioning procedures for simultaneous, delayed trace and 'backwards' conditioning are shown in Fig. 41, in terms of the way in which the conditional stimulus is related experimentally to the unconditional stimulus a (the stimulus that gives rise to A through the unconditional reflex $a \rightarrow A$). Of these, the 'backwards' procedure is ineffective (either for a real organism or for this machine). The remaining procedures in Fig. 41 do work. For organisms, both simultaneous and delayed conditioning are readily instrumented but delayed conditioning is more effective than simultaneous.

So far as the machine is concerned, trace conditioning is impossible. This occurs because there is no mechanism for retaining the 'trace' of the conditional stimulus which, in this procedure, is presented in the past and terminates before the unconditional stimulus. However, 'trace' conditioning is possible in certain more elaborate versions of the machine which contain loops capable of retaining a representation of the past stimulus. Both simultaneous and delayed conditioning work well, as they do for the organism, and delayed conditioning is the more effective. Thus, if the

unconditional stimulus b is paired (before and over-lapping with) the conditional stimulus a, and if this pairing is repeated, the following effects may be obtained: (1) response B either drops out or is modified; (2) stimulus b comes to elicit response A (or, occasionally a modified form of A) in the same way that $a \rightarrow A$; (3) the connection $b \rightarrow A$ is associated with a decreasing b, A, latency; (4) the connection $b \rightarrow A$ is eventually established, even in the absence of a. Hence, (5) $b \rightarrow A$ is a simple conditional reflex. Further, once $b \rightarrow A$ is well established, it may be used as though it were an unconditional reflex to produce another conditional reflex. Thus, if $c \rightarrow C$ is associated with $b \rightarrow A$, we can establish the reflex connection $c \rightarrow A$ and so on. Whereas the primary mechanism at work in establishing $b \rightarrow A$ is the 'extrinsic reinforcement', the only mechanism at work in establishing $c \rightarrow A$ is the 'intrinsic reinforcement' described in section 1.2.

Since each of these conditioning experiments is beset by the caveat 'or a modified form of response' a sequence of experiments can give rise to a variety of behaviours, some of which will be modified responses that are not manifest within the conditioning experiment itself. Thus, before conditioning, perhaps $d \rightarrow D$; after having established $b \rightarrow A$, it may be that $d \rightarrow E$. Bear in mind that the experimenter is grappling with a dynamic, restless machine that does something on its own accord if he fails to stimulate it in a suitable fashion. The machine's behaviour has a richness comparable to that reported by Pavlov and his school in their original writings (though its conditioned activity is more or less deterministic).

The chief distinctions between the behaviour of an animal and the behaviour of the machine occurs at a level of complexity stressed by Konorski (1962), namely:

1. Only certain features of a stimulus are relevant for conditioning (the animal has an innate structure, resembling a releaser organisation, in respect to aspects of its environment that may become significant).

2. The features of the stimulus that are relevant often depend upon the form of conditional response (certain features are significant in respect to some responses and other features in respect to others).

The deficiency of the machine as a model compatible with (1) is due entirely to the chosen identification of the stimuli as unitary entities (without this constraint it will mimic an organism). To achieve a performance compatible with (2), it is necessary to introduce a hierarchical structure.

Reflex Extinction If a conditional reflex such as $b \rightarrow A$ is elicited repeatedly in the absence of a it becomes extinguished. The b, A latency increases and ultimately b fails to produce A. The phenomenon of extinction is not a forgetting process (which may also occur due to the decay of the θ_{ij} values). Unlike forgetting, it depends upon the repetition of stimulus b.

Similarly, the phenomenon of 'spontaneous recovery' (which used to be pointed out as a demonstration that extinction is not merely forgetting) can occasionally be observed as part of the machines' behaviour. If $b \rightarrow A$ is extinguished to a long latency L_2, and if (after a rest interval) $b \rightarrow A$ is tested by applying b, then, if the rest interval is long enough, $b \rightarrow A$ will again be manifest at a shorter latency L_1 where $L_2 > L_1$. Spontaneous recovery in the machine is due to the fact that the response latency is the sum of the expectancy latency and the anticipation latency. Extinction increases the length of each latency period. After a rest interval the expectancy latency is likely to be reduced by the normal feedback to the stimulus unit local storage circuits so that the total response latency is L_1 rather than L_2.

3.2 The Instrumental Conditioning Paradigm For experiments in instrumental conditioning, the reinforcement variable is under the experimenter's control and is continually manipulated. (This is in direct contrast with classical conditioning, where reinforcement is a function of the organism's state or of an innately determined relationship between the organism and its environment.) The reinforcement may be interpreted as primary, examples of which are the delivery of food, events closely related to the reduction of a basic drive or, even better, the electrical stimulation, through an embedded electrode, of a positive centre in the brain. (Flood uses this technique in adaptation studies and the results are dramatically consistent. As a philosophical premium, this is what 'reinforcement' really is.) Alternatively, the reinforcement may be interpreted as *symbolic*; really, a misnomer, for the condition would be better stated as confirmation of an internal hypothesis.

Primary reinforcement will be considered first, in particular, reinforcement of an operant response. An operant is a response produced autonomously by an organism or by the machine, and the instrumental mode of conditioning depends upon the fact that autonomous responses are emitted to meet with reward or punishment. (The machine contains no analogue for punishment, i.e. a negative centre in the organism's brain. Thus it is only possible to achieve positive reinforcement or negative reinforcement of a response.) Discriminating stimuli are introduced to modify the autonomous activity or are provided, by the experimenter, in response to the activity of the system. For instrumental conditioning experiments the initial condition of the machine is a *tabula rasa* (apart from any residual charges that assign values to the θ_{ij}). In this condition, some response, say A, will be fortuitously rare and the experimenter selects A as the 'operant response' with which he is going to deal. In particular, he concentrates on response A and disregards other responses.

Interval Reinforcement Interval reinforcement consists of assigning a positive value (say of $R = +1$) to the reinforcement variable on the first occasion that A occurs after a fixed interval from the last occurrence of A. With respect to this procedure, the machine simulates an organism such as a pigeon in so far as the rate of response A is inversely proportional to the interval chosen. The reinforced response A is, of course, gradually extinguished if the reinforcement is withheld. Extinction is reflected in the response A latency and consequently the graph of the numbei of responses A per minute shows the short-term slowing down fluctuation that is characteristic of the animal subject. The fluctuations may be reduced by using randomly chosen reinforcement intervals around a fixed mean value (in place of a fixed interval). Although this procedure leads to higher response rates, it does not lead to a greater resistance to extinction of the operant conditioning (as it does in the pigeon).

Ratio Reinforcement In ratio reinforcement, the response A, is reinforced after a fixed number of occurrences, regardless of when they occur; the 'ratio' concerned being the ratio of unreinforced to reinforced responses. The experimental finding for animal subjects is that the larger the ratio, the more rapidly are responses emitted.

The machine parallels this behaviour within limits, and this is, at first sight, surprising. On closer inspection (or an illegitimate peep inside the black box) the reason is fairly straightforward. With larger ratios, the response A is reinforced in connection with a greater diversity of problem events $x_i = 1$. The latency is compounded from a stimulus unit latency and a response unit latency and the mean rate of response A depends inversely upon the average latency. But the average latency is less, as the problem events in connection with which the response A has been reinforced are made more diverse.

As with interval reinforcement, the behaviour shows the characteristic slowing down fluctuation which, once again, may be reduced by choosing a ratio at random (around some fixed average) in place of the fixed ratio. This procedure leads (as might be expected in view of the glimpse at the internal mechanism) to rather high response rates.

3.3 Discriminating Modes In the psychological laboratory, an animal may be trained according to several alternative reinforcement schedules, each of which produces a characteristic pattern of behaviour. If each reinforcement schedule is associated with a discriminating stimulus, such as different lights α and β, the appearance of a particular discriminating stimulus elicits the characteristic behaviour in the fully trained animal. The machine can be used to simulate this sort of experimental result if we choose a subset of the 'stimuli' as representative of one light and another subset of the 'stimuli' as representative of the other. Thus, light α is simulated by the vector $U = \langle +1,+1,+1,+1,-1,-1,-1,-1 \rangle$, and light β is simulated by the vector $U = \langle -1,-1,-1,-1,+1,+1,+1,+1 \rangle$. In this case, the machine is segregated, functionally, into a pair of subsystems α^* and β^*, one or the other of which is trained in connection with a particular stimulus α, β. Subsequently, the appearance of α will elicit one behaviour (α^*) and the appearance of β will elicit the other behaviour β^*). So far, the machine is fairly lifelike. Unfortunately, the machine cannot react as a real organism and test the reinforcement conditions of its environment to determine how it should perform. To simulate this 'choice' or 'test', it is necessary to use an hierarchically structured machine of a sort described later.

Next consider an extremely important characteristic of stimuli that act as discriminating stimuli; namely, that they become secondary reinforcers. The paradigm experiment consists of reinforcing a response, say response A, and associating the reinforced occurrence of response A with a stimulus b which would not, on its own, elicit response A. After the conditioning is established, the response is extinguished by withholding stimulus b and allowing the organism to respond with A for trials that are not reinforced. If the reintroduction of stimulus b without any reinforcement leads to the production of response A (or to a decrease in the response A latency) then stimulus b is a secondary reinforcer (in the initial pairing process it acquired the property of acting like a reinforcement).

Obviously, if this experiment is performed with the machine it will yield a positive result. Stimulus b, having been reinforced in connection with response A, is part of a conditional reflex $b \rightarrow A$. Hence, by previous definition, stimulus b does act like a positive value of R. Further, as in the classical conditioning experiments, it is possible to build up conditional reflexes such as $c \rightarrow A$ by association with $b \rightarrow A$. Finally, because a response is not determined uniquely by the problem state evoked in the machine, it is possible to manipulate and modify the mode of response by using secondary reinforcing stimuli.

Unfortunately, in the machine as described, the secondary reinforcing stimulus will not play the crucial trick of reinforcing the emission of some response other than A, say response B. Secondary reinforcers are not generalised as they are in animal learning. It is a comparatively simple matter to modify the machine (by introducing a further mode of 'internal' reinforcement) so that 'generalisation' of this kind does take place. The modification works. But, as it probably works for the wrong functional reason, we shall not dwell upon it. (To make it work for the right functional reason entails building a machine that can accept several sorts of reward, rather than the single reward of value R.)

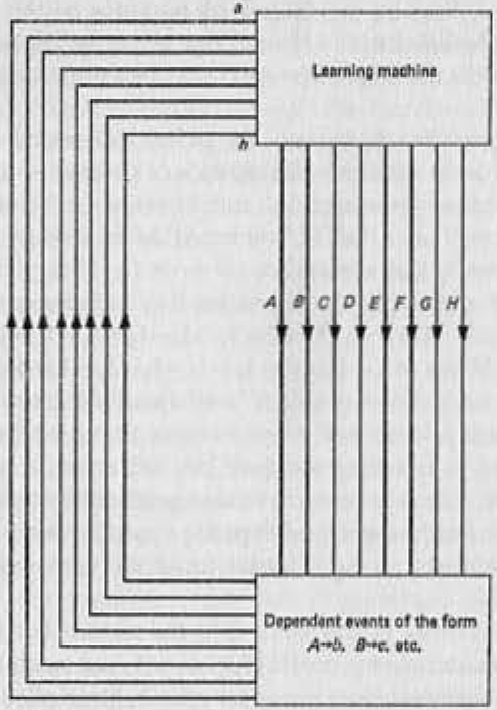


Figure 42 Contingent environment.

3.4 Chain Responses It is easy enough to establish chains of reflex actions. If the production of response A gives rise (in the environment of Fig. 42) to stimulus b and if $b \rightarrow C$ is a conditional reflex, then the chain $A \rightarrow b \rightarrow C$ is built up. The process may be repeated up to a chain length of eight as a maximum; for example, to produce chains such as $A \rightarrow b \rightarrow C \rightarrow c \rightarrow D \rightarrow d \rightarrow E \rightarrow R \rightarrow F \rightarrow f$.

As in the case of animal conditioning, the production of chain reflexes relies upon the initial establishment of the component reflexes and a later assembly of these components into the composite reflex entity.

3.5 'Symbolic' Experiments The experimental environment can be given a quasi-symbolic interpretation, as follows. In the case under discussion, some relationship, $\mathscr{R}$, is defined between the stimulus set and the response set. One $\mathscr{R}$ is a permutation of the numbering of the stimuli with reference to the numbering of the responses; but a 'many-to-one' relationship is equally valid. Given a stimulus the machine is required to produce the correct response, that is, the $\mathscr{R}$ related response to this stimulus. The variable, R, is identified with a 'knowledge of results' signal that assumes the value $+1$ if the machine response is a correct response and the value 0 if it is a mistake.

Within this framework, stimuli are to be interpreted as events that definitely occur or do not occur; the absence of a stimulus negates an event, i.e. complementation is presupposed, and, in a very weak sense, the machine entertains an hypothesis that is 'confirmed or denied' by 'knowledge of results'. To embody this constraint, we write '-1' in place of '0' in the stimulus vector and apply the corresponding negative potentials to the machine; thus 'stimulus a' is $U = \langle +1,-1,-1,-1,-1,-1,-1,-1 \rangle$, and 'stimulus b' is $U = \langle -1,+1,-1,-1,-1,-1,-1,-1 \rangle$. Responses designate solutions to a relational problem. 'Anticipation' becomes the characteristic of a selective process and it makes sense to regard the response unit decision process as resolving more or less 'uncertainty' regarding an $\mathscr{R}$ related response, given the problem state engendered by a particular stimulus. (The amount of 'uncertainty' depends upon the form of the decision process at a given trial; the greater the uncertainty, the longer the anticipation latency.)

The cueing variables z_j now come into the picture as constraints upon the response unit decision process. If $z_j = -1$, the machine is instructed that the decision process must not select $y_j = 1$. Generally, the assignment of negative values to some or all of the z_j for some or all of the anticipation interval at a given trial, will reduce the machine's selective uncertainty at that trial. (In conditioning terminology, this would be response differentiation.) Conversely, of course, if $z_j = +1$ the machine is instructed to select $y_j = 1$. Either mode of cueing is possible but it is sufficient to consider the case in which the z_j are assigned negative values.

The machine can be trained to use a skill entailing knowledge of $\mathscr{R}$. The training procedure amounts to repeatedly presenting each stimulus, constraining the machine to make a correct response by the use of the cueing variables, and positively reinforcing the correct response when it occurs. Further, the machine will continue to perform the $\mathscr{R}$ based skill as long as the environment provides it with a sufficiently varied and rapid sequence of stimuli to replace the variety of the stimulus search process. (Recall that the stimuli are associated with definite values of $u_j = 1$.) In particular, the machine performs adequately as a chain conditioned device where $\mathscr{R}$ is of the form $a \rightarrow A$, $b \rightarrow B$, and the environment completes the chain by consequential events of the form $A \rightarrow b$ and $B \rightarrow c$.

At first sight, these operations amount to a rather stupid renaming of entities; in particular, inputs have been constrained so that the operation that used to be called 'reinforcement' is plausibly relabelled 'knowledge of

results'. But, on closer examination, something more has been gained. The concurrent modes of operation and the synchronisation phenomenon that are latent in the machine's activity (and which give it non-trivial attention and expectancy properties) break through the framework of constraints imposed to *mimic* problem solving or symbolic transformations in such a way that certain conditions *cannot* hold and certain types of training are rejected. The machine, constrained to this extent, demonstrates a number of 'abhorrences' some of which are very lifelike.

The machine cannot learn to sit quietly until a stimulus occurs (perhaps for an indefinite interval). The condition in which all stimuli are negated is contrary to the tenets of the model. Hence, it cannot be trained to 'learn $\mathscr{R}$' in the commonsense meaning of the phrase, if $\mathscr{R}$ applies to the entire repertoire of stimuli and responses. The abhorred condition is 'all $u_i = \ldots 1$' (so that the stimulus search process is effectively inhibited and no problems are produced). The electronic consequence of this condition is that no 'response search' instruction is delivered, and, ultimately, all of the responses assume states $y_j = 1$. The philosophical consequence is that we have disobeyed the (tacitly stated) rules of a machine that should actively search for problems and, given problems, search for their solutions.

To avoid breaking the rules whilst giving a commonsense meaning to the phrase 'the machine learns $\mathscr{R}$', it is necessary to restrict the stimuli and responses related by $\mathscr{R}$ to a subset of the entire repertoire, for example, by saying that $\mathscr{R}$ relates stimuli, a, b, c, d, to responses A, B, C, D; that stimuli e, f, g, h, feature as 'internal stimuli' with $u_e = 0$, $u_f = 0$, $u_g = 0$, $u_h = 0$; and that responses E, F, G, H, are 'irrelevant responses'. Problems corresponding to e, f, g, and h, may thus occur without external stimulation, and responses, E, F, G, and H may be made without being counted as relevant responses. By this expedient the machine is given a field of attention on which it can concentrate even if the relevant (skill orientated) field of attention is held void by the experimenter.

4 Teaching and Training

A machine that has been partitioned in this fashion to mimic problem solving, cannot be trained or conditioned by the simple extrapolation of techniques that *would* work for any similarly dressed-up reflex making device. There are some broad, but from the foregoing comments, obvious principles of effective teaching. These will be stated. After that, the principles will be embodied in a suitable teaching machine. Although these comments apply to a machine in which the 8×8 array is a *tabula rasa*, it is much more interesting to consider the case of a machine with previous experience (so that the 8 × 8 array contains definite entries θ_{ij}). In particular, a part of this previous experience is likely to interfere with the acquisition of the $\mathcal{R}$ skill.

The principles are as follows:

1. The teaching procedure must retain the machine's attention. If it does not, the stimulus search process will give rise to irrelevant problems. Conversely, the rate of presentation of stimuli must not exceed the point at which the machine becomes overloaded.

An instrumentation of this principle involves a feedback loop, through which the instructor is informed whether or not the machine's attention is occupied. As a result of this information, he (or it) may adjust the rate and variety of the stimulating input.

2. Correct responses must be reinforced. If the knowledge of results were genuine it would be sufficient to say that a certain response is correct. Here, assigning a positive value to R is, of course, a trick standing in place of real confirmation. But another feedback loop involving the machine and the instructor is required for this purpose.

3. Effective teaching should allocate more effort to problems that give rise to individual difficulties (the instructor should present the machine more often with stimuli that produce mistaken responses, so that these aspects of the skill are well rehearsed). Once again, this principle involves a feedback loop. The instructor must determine which stimuli are associated with mistaken responses, and present a controlled sequence of stimuli to the machine.

4. The instructor should co-operate with the machine by introducing cues that reduce the response selection uncertainty. This will reduce the number of mistakes and avoid unduly long anticipation latencies.

5. There is, however, a requisite variety (in Ashby's (1964c) sense) built into the response search process. Hence, the response selection uncertainty must not be reduced beyond a point that would inhibit the response search process.

6. Again, cueing should be minimised on the grounds that if the anticipation latency is cut down (by excessively rigid cueing) the machine will be unable to get rid of mistaken tendencies (or θ_j entries) that interfere with the performance of the skill. The machine does, quite genuinely, learn best from 'near misses' (this is the argument used by Winston (1970) in a slightly different context). The required degree of cueing depends upon the form of the decrementation process. In general, mistakes can only be rectified if the machine is allowed to exhibit its mistaken tendencies and it is also generally

true that the way a skill is taught and learned should be adapted to the existing internal organisation of the machine.

7. Clauses (4), (5), and (6), can be satisfied by a delayed cueing procedure. Cues are delayed to a variable extent after the delivery of the stimulus. The delay in each cue, which may be so great that the cue never appears, is modulated as a function of the machine's activity to minimise the amount of cooperation needed to maintain learning.

One simple but effective instructional algorithm starts by delivering all the cue information with every stimulus. As the learning machine responds correctly, the cue information is delayed until a mistake is made. The cueing required to correct this mistake in respect of a particular stimulus is presented earlier after *that* stimulus until the mistake has been eliminated (with reference to *that* stimulus). This operation is repeated for all of the stimuli. Ultimately, when the machine responds correctly and with the required minimum latency to all of the stimuli, cooperation is withdrawn entirely since all the cues are so much delayed that none of them appear. In summary, this is a cueing algorithm that selectively increases the difficulty of the task (by selectively withholding cooperation) as the machine's proficiency increases. Its execution necessarily involves yet a further feedback loop.

8. The delay of a cue is a 'delay relative to the mean latency for a correct response' (that is, relative to the expected value of the sum of the expectation latency and the anticipation latency for a correct response) rather than an 'absolute delay'. A final feedback loop is needed to estimate the mean latency.

5. The EUCRATES Teaching Machine

It would be quite possible for a nimble-fingered, real-life instructor to use these principles to teach the learning machine a relational skill. But four separate feedback loops are involved, each of them requiring a certain amount of calculation. In practice, therefore, the instructors' task would be rather difficult. Further, his difficulties are unnecessary, for the entire teaching process can be automated using a machine that engages in partially cooperative interaction with the learning machine (or, alternatively, with a real-life student).

This automated system (outlined in Fig. 43) was realised as the EUCRATES teacher. The block outline of the teaching component is refined in Fig. 44, from which it will be clear that the EUCRATES teacher is an image of the EUCRATES learner constrained by the requirement that $\mathcal{R}$ is satisfied and by the rules listed below.

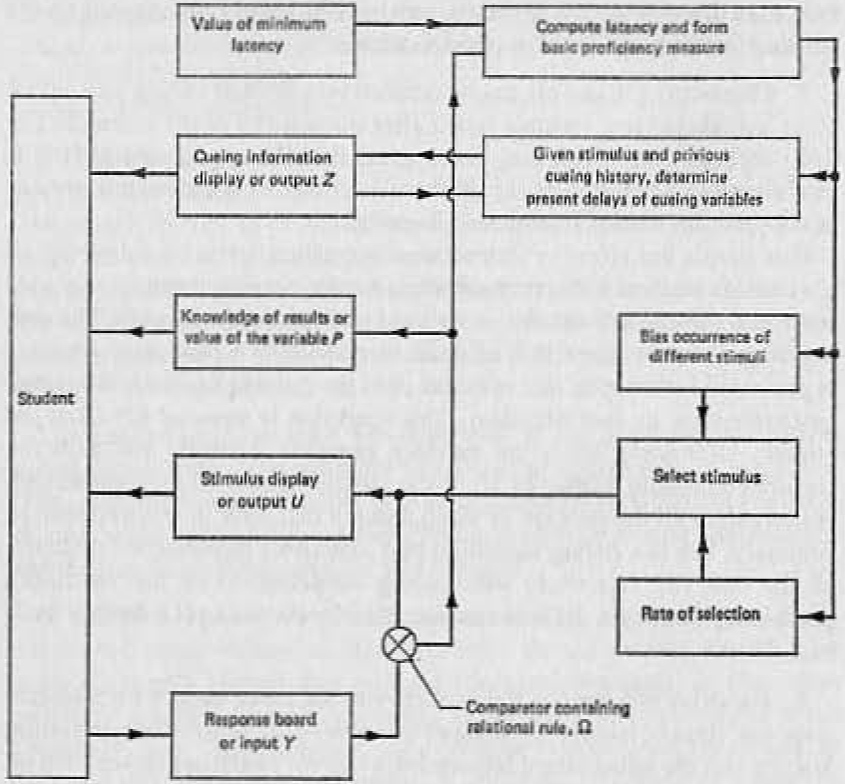


Figure 43 Plan of a teaching machine.

The student's or machine's response is first compared with the $\mathcal{R}$ transform of the stimulus delivered, to determine its rectitude and its latency. Both terms enter into the four feedback loops (Fig. 43) and the data are collated to give a basic measure of performance.

(a) The teaching machine specified in Fig. 43 forms an estimate of the sum of the expectation latency and the anticipation latency from an estimate of the mean correct response latency.

(b) Biasses a random selection of possible stimuli so that stimuli yielding correct responses are less often selected.

(c) Provides a knowledge of results signal which, in the case of the machine (rather than a human student), is returned as the value of variable *R*.

(d) Determines the degree of simplification of the problems posed under the selected stimuli by assigning a variable delay to the cueing information

(in the case of the machine by assigning a value $z_j = -1$ to each of the cueing variables after a variable delay interval).

The last process requires further comment.

(i) The delay introduced is relative to the estimate of latency as calculated in (a) above.

(ii) The delay associated with variable z_j must be computed separately for each stimulus, *i*. Hence, the basic measures from the comparator must be averaged with reference to an array of 8×4 contingencies; eight cueing variables combined with four stimuli.

The requirements of (i) and (ii) are instrumented (Fig. 44) by an inverse of the learning machine. The biassed random selector is replaced by a stimulus search process having a basic repetition rate determined by the mean expected latency. Values of 8×4 averages of the performance measure are compiled in the 8×4 analogue storage array of Fig. 44 (as indices θ_{ij}' analogous to the θ_{ij} in the learning machine), and the rows of this storage array (selected by stimuli) provide a slope determining input to a set of four sawtooth generating circuits. These circuits, which are analogous to the response circuits in the learning machine, contain voltage limiters. When the *j*th sawtooth voltage exceeds a predetermined value, the cueing variable z_j is set equal to -1. The sawtooth voltage is returned to 0 when the next stimulus is displayed.

6 Learner/Teacher Distinctions

Adopting the behaviouristic and causal stance, the learner/teacher system in Fig. 44 may be partitioned into two finite-state machines, namely a 'learner' and a 'teacher'. This picture is plausible if the actual mechanism of EUCRATES is constrained by the various tricks already described: for synchronising the separate machines, for ensuring that a stimulus *begins* a selective process which gives rise to a response that *ends* the process, and for delineating stimuli that are input states and responses that are output states of the machines in question. Although the tricks are not easy to play they can often, though not always, be played successfully and, under these circumstances, it is possible to construct contingency tables representing the frequency of events, joint events, etc.; for example, stimulus/response/cueing tables. If the frequencies in question are interpreted as probability estimates, it is also easy to plot statistical learning curves, for instance, in terms of correct response probability.

By adjusting the parameters of the learning machine and restricting the teaching machine to act as a simple environment (a degenerate finite-state machine) it is possible to replicate the results of statistical learning theory. By removing the restriction upon the teaching machine it is generally possible

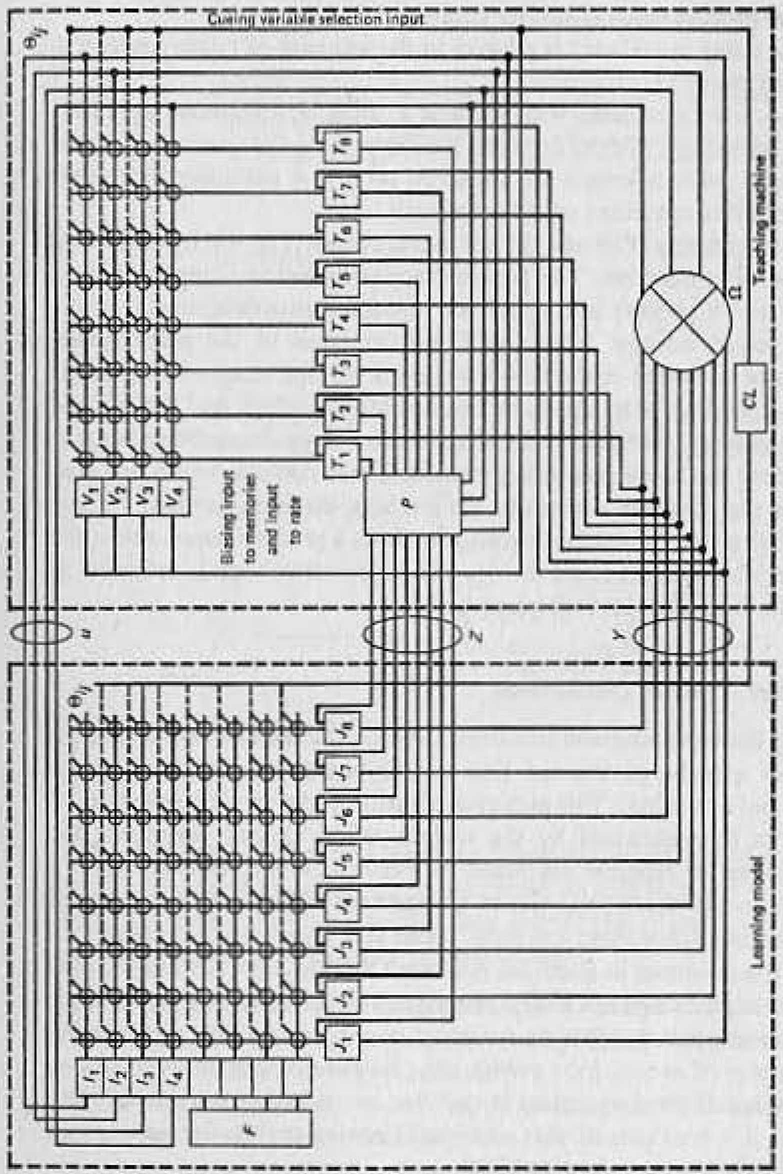


Figure 44 Learning and teaching system: E, inputs of $u = 0$ in learning model; V, stimulus selection generating units; T, sawtooth cue generating units; P, rule based on Ω determining which response should not be inhibited by the sawtooth cueing output, given a particular stimulus; CL, compute latency and form the basic proficiency measure.

to show that an adaptively modulated teaching process leads to more rapid learning than any of the commonly advocated stimulus, cueing and reinforcement schedules. Reams of these curves exist, but are of chiefly parochial interest in solving specific design problems. The general result (that the learning system converges in various ways and at various rates to an asymptotic condition) is better shown by the mathematical formulae of Bush, Estes, Mosteller and others in the field of statistical learning theory (*see* Luce, Bush, and Galanter, 1963) or by work on the statistical theory of optimal teaching operations due, for example, to Matheson (1964) and Smallwood (1962). It is easy to convert the response probabilities, etc., into indices of selective information and consequently to represent the data in information theoretic terms. There are some advantages in doing so, as pointed out by Wattanabe (1963).

7 Von Foerster's *Reductio ad Vacuum*

The formulae of statistical learning theory are unquestionably elegant but unavoidably correct. That is, any pair of finite-state machines coupled to one another (Fig. 45) is bound to form a large finite-state machine with a deterministically or probabilistically convergent behaviour. If the organism

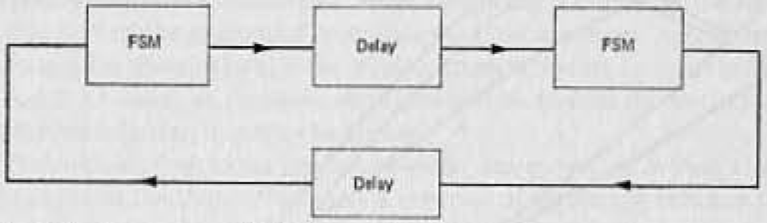


Figure 45 A pair of coupled finite-state machines.

or the learning machine is restricted to act in this fashion the result is inescapable; as pointed out by Von Forester (1971) in his discussion of finite-function machines. Of course, it is possible to obtain a large number of statistical behaviours within the compass of this framework, depending upon the structure and initial states of the finite-state machines. Some, mostly culled from Von Foerster (1971), are shown in Fig. 46 where the response frequencies are transformed and summarised as selective information measures. However, it is unnecessary to call any of these processes 'learning' and, in common with Von Foerster, I am inclined not to do so. These results show 'adaptation', no more and no less. Learning in a non-trivial sense, becomes possible in so far as the finite-state machines are augmented by an exploratory or attention directing mechanism able to modify their internal state sets, i.e. if, in Von Foerster's sense, at least one of them is a finite-function machine.

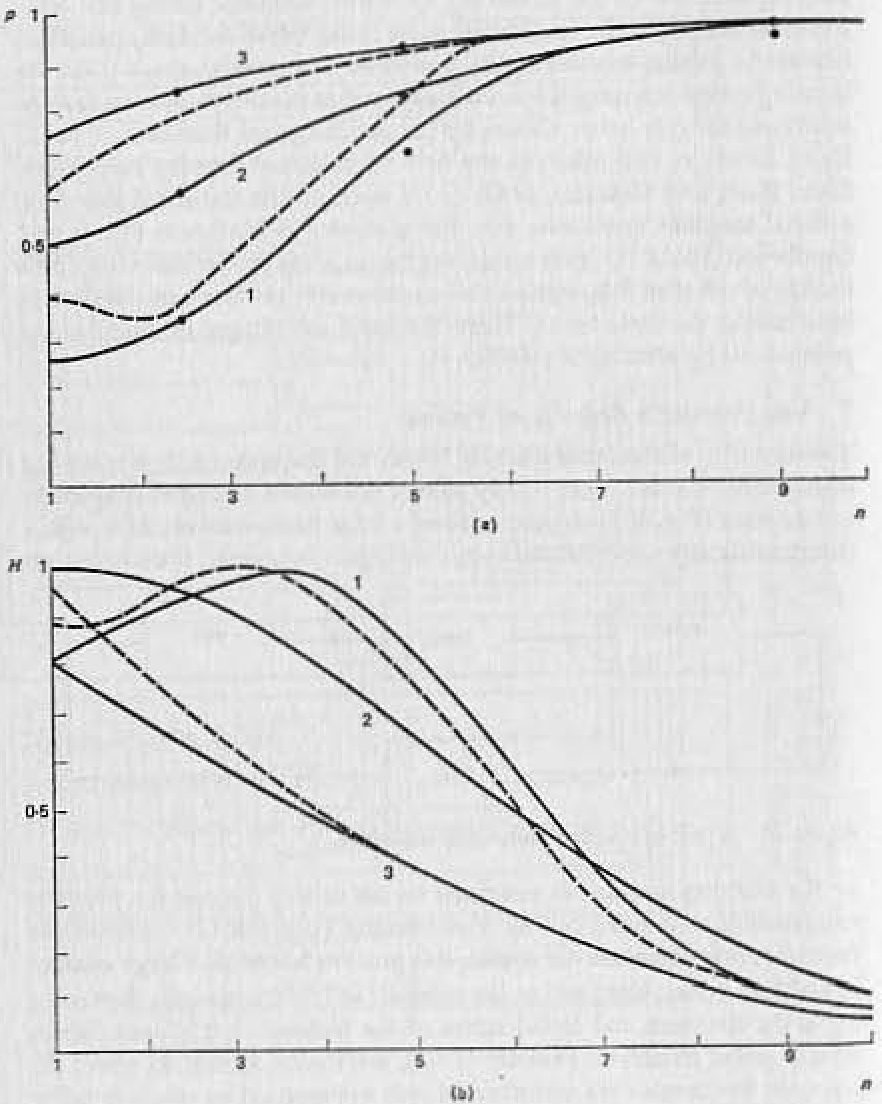


Figure 46 Typical correct response probability curves (*a*), and their transformed representation as indices of selective uncertainty/information with respect to an external observer (*b*). Three cases are from Von Foerster's simulation and two are from EUCRATES.

8 Attention Directing and Constructive Mechanisms

It is possible to obtain a (rather cackhanded) model for non-trivial learning by relinquishing the constraints imposed upon the system of Fig. 44 in order to secure two machines featuring as 'teacher' and 'learner'. The essential condition is that the learner's output can form a non-numericised fuzzy input to the teacher and, vice versa, that the teacher's output can form a non-numericised fuzzy input to the learner.

It will be evident that the resulting system is distributed; there is no functional distinction between the teacher and the learner. Or, to make a point also voiced by Von Foerster (and which is generally valid) *learning* entails a *teacher who learns*, just as teaching entails a learner who learns. Any theory of learning in the preferred sense is a theory of teaching, and vice versa.

Although the learner/teacher distinction is satisfactorily eliminated there remains an engine responsible for evaluating responses with respect to the relation $\mathscr{R}$ and determining whether or not the machine is reinforced. Crudely, but seriously enough, this engine is an arbiter of veridicial truth; if a response is taken to index a problem-contingent solution then only true solution-statements are rewarded. Since there is but one relation, the action of this part of the equipment trivialises the joint machines' operation by thwarting the attempt (which the joint machine is bound, by construction, to make) to direct its attention elsewhere and to explore its environment for further relations that may be learned.

The method of surmounting this obstacle, recommended in Pask (1960) is to augment the truth criterion by a criterion of agreement between subsystems of activity (initially, the subsystems localised in the teacher and learner parts of the apparatus) and to enlarge the apparatus so that physical limitations, such as the number of storage locations, no longer play a dominant role. Under these circumstances, replicas of the teacher and learner *organisations* can coexist and cooperate in the enlarged network as dynamic but stable entities (the stable configurations in networks described by Aleksander (1973) are examples of such entities). It is still true that any aggregate of two or more cooperating subsystems must *satisfy* $\mathscr{R}$ and that they must also *agree* (in the sense that the joint system is stable in the domain of $\mathscr{R}$). But the several distinct pairs (or aggregates) of subsystems coexisting in the network correspond to different descriptions of $\mathscr{R}$; in particular, each one realises a distinct method of computing $\mathscr{R}$, i.e. they are distinct *intentions* of the same extensionally specified relations or, in psychological parlance, they correspond to different methods of solving the same problem or performing the same skill.

This proposal remains valid and it has been embodied in a physical artifact which operates in this manner. But the proposal is not unique and it is worth considering at least one alternative which can be usefully contrasted with the first system.

Instead of a single relation $\mathscr{R}$ let there be a plexus of related relations; a relational mesh of which $\mathscr{R}$ is a part. Further, let the constituent relations be named; for example, by an index *i* so that $\mathscr{R}$ becomes (for some *i* value) $\mathscr{R}_i$. Finally, let there be a description of this relational mesh so that the $\mathscr{R}_i$ can be represented in the context or neighbourhood of other constituent relations.

For the moment, the question of how a mesh of relations should be constructed, of what constitutes a learnable field of relations and of what constitutes a satisfactory description are all left open to be addressed and answered at a later stage. Regardless of the particular answer, it is certainly true that EUCRATES could direct its attention from $\mathscr{R}_i$ (having learned $\mathscr{R}_i$) to a differently named relation in some other part of the mesh.

It could do so haphazardly (a dubious argument because something, perhaps a random selection, tells it what to attend to next). More cogently, there might be an organisation in EUCRATES with the ability to comprehend not the *relations* but *a description of the relations*. This organisation is conveniently regarded as residing at a higher level in an hierarchy of control and acts as a specific orienting or attention-directing unit. By symmetry, it would also be responsible for constructing or monitoring the construction of those processes (alias cooperative aggregates or putative learner/teacher *organisations*) that solve problems or satisfy relations; that is, the attention directing unit is able to perform the other functions (demanded from time to time in the previous discussion) of an 'hierarchical' system design.

Where does the hierarchy come from? It might be built into the structure. On the other hand, it need not be. The processes that direct attention and/or construct problem solving processes are not necessarily distinct in kind from the problem solving processes themselves; they merely act upon a different domain. If that *is* so, then the required mechanism develops already, in the first system, even though the first system is not explicitly hierarchical. In fact, the hierarchy is a fiction, introduced for the convenience of an external observer, which may or may not be reified as an organisation or structure.

The difference between the first proposal and the second is simply that between internalised replication of methods for computing the same relation (first proposal) and an attention-directing exploration, in which fresh relations are discovered and learned (second proposal). These proposals are not in the least contradictory and it can be argued that both of the activities they presage are manifest in a competent learner, either man or machine.

In either case, the basic organisation consists in cooperative elements to be identified with teacher and learner (though unlike the original teacher and learner, they are seldom localised in one part of the physical equipment). Viewed from the point of view of cognition rather than learning, they appear as the 'proposer' and 'critic' subroutines that Minsky and others believe to be important constituents of any competent 'artificially intelligent' program with a goal of the kind under discussion. According to the present point of view, the distinction between teacher and learner, in common with the distinction between hierarchical levels, is chiefly a matter of observational convenience (perhaps, of necessity in so far as sharp valued measurement depends upon marking such a distinction). But it is not generally a matter of *fact*. These distinctions become matters of fact only if the machine designer, or nature in that capacity, happens to build up structures in which the teacher and learner or the hierarchical levels are specially localised. When such constraints are introduced, as they are in most of the models discussed in the sequel, they are acknowledged to be arbitrary and, however convenient they may be, they are quite inessential.

6 Models for Learning with an Hierarchical Structure

1 A Family of Models for Code Learning Experiments

The family of models described in this chapter has an hierarchical structure built into it as a convenience both for observation and in wedding the model to the experimental or tutorial situation it represents. If one recalls the caveats of the last chapter, this structure is (as it is said to be) a convenience. All the same, comparable structures exist in the design of adaptive teaching and training machines.

The model is also partitioned into a component called the learning model and a component called the teaching model. The latter component is trivial; it simulates the mechanism described in Chapter 5 and is an elaboration of the steady state regulators to be discussed in Chapter 7. In contrast, the learning model is non-trivial though it does not tally with my current beliefs and has several outstanding but instructive deficiencies.

In principle, the learner could tackle almost any miniature environment. Although the learning model lacks the attention directing and strategic equipment needed to shift its operations from an immediate goal, it does contain a process for generating a signal indicating that there is an overload condition (problems posed under the goal it is dealing with are unintelligible) or a signal indicating an underload condition (problems posed under the goal it is dealing with can be solved, but it is impossible to construct any further procedures for solving them in different or more effective ways). The overload signal is interpreted as an instruction to build fresh procedures, since none exist (problems unintelligible) and its efforts may or may not be successful. If the efforts do prove successful the model uses the procedures it has built and it eventually reaches a condition of underload. If not, the overload condition persists and can only be resolved by a change in the focus of attention. Hence, either signal indicates an impasse unless there is some means of changing attention. A means for doing so could easily be built into the model; for example, the problem solving operations could be applied to a domain of problem descriptions (the description of a relational mesh posited in the last chapter). But this facility is not provided in the

learning model as it stands. Alternatively, the model's environment could change, by happy accident, to provide the model with the variety it needs.

In fact, the learning model is executed in concert with a teaching model which contains an attention directing mechanism represented as a tutorial strategy. This arrangement is employed to reflect the general thesis that the controller in a real adaptive teaching system (and that is the model's main field of application) acts as a surrogate for the real student's attention directing mechanism or, in so far as its instructions conflict with those of an active mental organisation, the controller usurps the position of the internal mechanism and overrides it. To represent free learning it is thus necessary to augment the learning model. But one, very restricted, augmentation is the teaching model, and the *combination* of teaching model and learning model may be viewed as a very specialised model for *free* learning.

Whichever interpretation is given (and, for the sake of clarity, only the first or 'adaptive system' interpretation is pursued) the learning model and the teaching model interact. This interaction may or may not involve the 'impasse' signal indicating a persistent condition of either overload or underload.

The teaching model, like any adaptive teaching machine, has a predictive rule built into its design, so that it can infer from the image of a student's differential performance when it is necessary to change the relation being dealt with, hence providing an environment that 'anticipates' the impasse signal and suppresses its occurrence. This alone may be sufficient to stabilise a learning process; it would be, if the prescription built into the teaching model incorporated an adequate description of the learning model or, in reality, if the prescription built into the adaptive teaching machine incorporated an adequate description of the (real) learning process. But the two components, either modelled or real, are bound to learn, in some sense, about one another, and they may get out of order. Transfer of the impasse signal is a minimal coupling which is often sufficient to bring them into register.

The hierarchical structure in the model has two levels, interpreted as levels of control (Tarjan, 1963; Mesaravic, 1962, 1963), which are henceforward designated *Lev 0* and *Lev 1*. The learning model's *Lev 0* repertoire consists in a collection of organised problem solving procedures or procedures for attaining goals/subgoals. These are called *0 operators*. Although the *Lev 0* repertoire may contain initial entries it is possible to execute the model even if the *Lev 0* repertoire is initially empty. The learning model's *Lev 1* repertoire is a collection of organised constructive processes called *1 operations*. The *1 operations* are fairly limited but there are no other than technical restrictions upon their form or complexity. In any case, the *Lev 1* repertoire is identified with the mental organisation acquired by a real student, through maturation, imprinting, and previous learning. The most pertinent possibility is that the repertoire is programmed into the subject's mind as part and parcel of the experimental or tutorial contract (Chapter 4) which he accepts in adopting the role of student.

Both *Lev 0* and *Lev 1* operations feature as problem solvers. They differ in respect of their domain. The *Lev 0* operations have a domain of descriptions. (Of 'inputs' or of 'symbolic stimuli' presented under a specified goal relation.) Input descriptions are interpreted as intelligible problems if and only if they belong to the domain of some *Lev 0* operator existing in the *Lev 0* repertoire. If so, the learning model can begin to solve the problem by applying *Lev 0* operators; a process which may lead to correct solution or a partial solution (or a mistaken though believed-to-be correct solution). The *solution* or partial solution *is* the series of operations that are applied in a proper order. As a result of executing the solution a *response* is selected. Hence, *responses* name classes of solutions.

In contrast, the domain of any *Lev 1* operator is either the *Lev 0* repertoire or (in the case of a 'create' operation) a randomised process. A *Lev 1* solution constructs or modifies the *Lev 0* repertoire which is its internal 'response' (in the case of the 'create' operator the 'response' is the addition of a primitive *Lev 0* operator to the *Lev 0* repertoire).

From the preceding comments it is evident that the model is a normative rather than a behavioural model. That is, the experimental or tutorial situation it represents is contractual or game-like and is associated with an, albeit primitive, language in which it is possible to describe conditions and to give commands or ask questions which the real subject, in the role of student, *understands* (for example, the command 'attend to a goal relation', or a question like 'which of several statements satisfies the goal relation?'). Similarly, either the learning model or the real subject entertains an hypothesis whenever it begins to solve a problem, and this hypothesis is open to confirmation or denial (recall that a description poses a problem under the current goal relation if it is in the domain of at least one of the *Lev 0* operators in the current repertoire. But the solution to this problem may be partial or mistaken). In so far as the teaching model or the real adaptive machine delivers a signal indicating that a response is correct or partially correct, this either confirms or disconfirms the prevailing hypothesis. Any condition of the learning model that refers to a problem or its solution or the confirmation of its solution is henceforward given the general title 'state of knowing'. This rubric covers any condition upon which the model can operate or upon which it is operating.

These remarks are important because the model is exhibited in the context of the code learning task described in Chapter 8 and the 'language'

involved is so embryonic that it is easy to argue it out of existence, or at least, to consign it to a category of constructs that are pedantic enough to appear irrelevant. As a matter of history, the entire (rather lengthy) process of programming and executing the models of this family, comparing them with experimental performance, and so on, was plagued by persistent attempts, on the part of more tidy minded research staff, to 'simplify' the concepts involved. These 'simplifications', though well intentioned, reduce the family of models to triviality and reduce comparisons between the model and reality to the most banal variety of curve fitting.

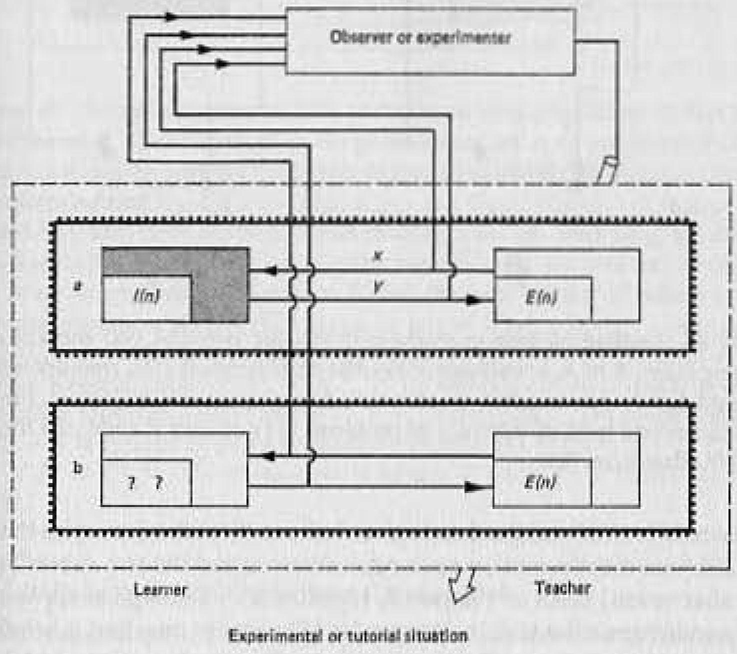


Figure 47 Summary of the family of models: $I(n)$, repertoire in learning model at nth trial; ??, repertoire of real subject at nth trial to be modelled; $E(n)$, tutorial rule together with data accumulated about the subject at the nth trial; hatched area, resource allocation and *Lev 1* operators but *not* attention direction system. *a*, simulated learning/teaching model; *b*, real learner/teaching machine (regulator).

2 Structure

The model (or family of models) is summarised in Fig. 47. A specific model is identified under a normative interpretation with a specific experimental or tutorial situation. By way of notation, the input descriptions are labelled

x, responses y and states of knowing u. The repertoire in the learning model at the nth trial in the experiment is called $I(n)$, and the tutorial rule in the teaching model, together with the data accumulated about the student, is called $E(n)$. In general the form of E is fixed throughout an experiment. But the form of E may be changed between different experiments by redesigning or reprogramming the adaptive machine.

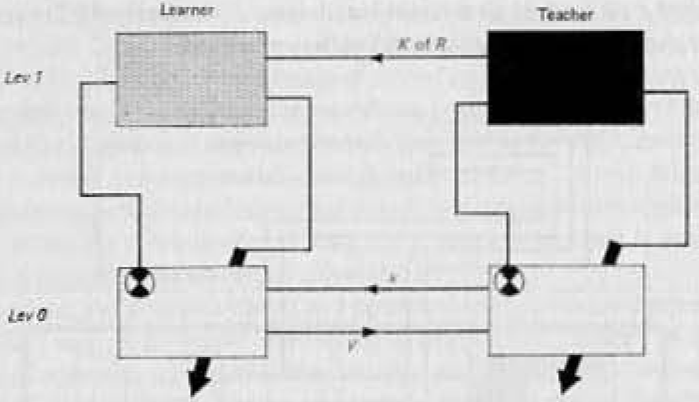


Figure 48 Outline of leaning/reaching model for one and two subskill code learning tasks: K of R, knowledge of results; cross-hatched area, resource allocation and *Lev 1* operators but *not* attention directing system; solid area, tutorial rule that selects level of difficulty of problems (x) and selects code rule for two subskill alternation tasks.

The results cited and the details given refer to identifications with the one subskill (one transformation code rule) situation and the two subskill (two rule alternation) tasks of Chapter 8, together with incomplete application to the transformation task in Chapter 10. (The reader may find it helpful to glance over the task specifications given in Chapter 8, sections 1, 2.3, 2.4 and 2.5 before proceeding.) These identifications are shown in Figs. 48 and 49. The descriptions of the subskills, used in specifying E, are isomorphic. Hence only one description is shown in Fig. 49. Certain subsets of the input sets (X_i; $X = \cup X_i$) are indexed collectively by the difficulty variable η and are referred to as sets X_η. The difficulty index (or the pair of difficulty indices η_1 and η_2) correspond (inversely) to degrees of simplification if, whenever the learning model is able to interpret an input $x_a \in X_\eta$ as a *problem* then an input $X_b \in X_{\eta-1}$ is nearer to solution than x_b and an input $x_c \in X_{\eta+1}$ is either further from solution or *uninterpretable*; that is, unintelligible. The notations, U, U_i' stand for a set of states of knowing and some of its subsets. Thus, if there is an input x which *could* be interpreted as

a problem *if* it were presented; then if x is presented ($x \in X_i$) the 'state of knowing' is $u \in U_i$.

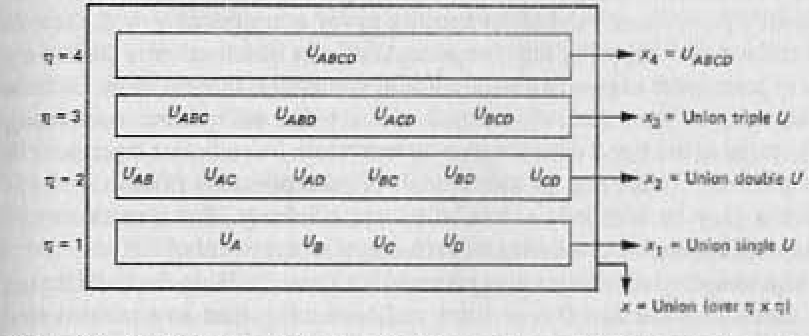


Figure 49 Stimuli are generated, as patterns, by assigning values to four signal variables A, B, C, D (2 valued in the present account, 4 valued in other experiments). η alters the *number* of variables treated as stimulus coordinates. There are 15 pattern subsets $U_A, U_B, \ldots, U_{AB}, U_{AC}, \ldots; U_{ABC}, U_{ABD}, \ldots; U_{ABCD}$. For 2 valued variables there are 16 elements in U_{ABCD}; but the generating set contains only 8 stimuli, x, so chosen that if further stimuli, $x \in X$ are obtained by deleting one or more coordinates, all subsets U with the same *number* of indices contain the same *number* of stimuli (8 in U_{ABCD} or any of $U_{ABC}, U_{ABD}, \ldots$; 4 in any of $U_{AB}, U_{AC}, \ldots$). If X is the union of all subsets, U, then $X \subset X$. Similarly, there are four response variables A^*, B^*, C^*, D^* for which there are 15 index subsets V. A complex response, y, is a member of $Y =$ Union of all V. The correct mapping Ω; $X \to Y$ is induced by a one to one correspondence Ω^*; $(A \times B \times C \times D) \to (A^* \times B^* \times C^* \times D^*)$. A response is correct if and only if $(x,y) \in \Omega$; $x \in X$, $y \in Y$.

3 Limitations on the Operations Carried Out

In the experiments of Chapter 8 a subject is placed under a temporal restriction. If a response, y, is to count as a correct response to an input, x, the response must be made within a short interval, δt, after the input is presented. In addition, a correct response to x must satisfy $y = \mathscr{R}(x)$ where $\mathscr{R}$ is a coding rule; Ω in Fig. 49 (or, in the case of more than one subskill, $y = \mathscr{R}_i(x)$ where $\mathscr{R}_i$ is the rule in force at the trial under discussion). Chapter 8 indicates that x may be a simple input (one signal lamp) or a complex input (several signal lamps, $x = \langle s_i, \ldots, s_c \rangle$) and that the number of members of x depends upon and increases with η to a maximum of 4. As the transformation rule $\mathscr{R} = \Omega$ is a coding rule, a correct response must have as many components as the input (that is, if $y = \Omega(x_i, \ldots, x_c)$ then $y = \langle r_i, \ldots, r_c \rangle$) but it must still be made within δt seconds after the appearance of x.

Hence, especially for larger values of η, the strict temporal constraint places a severe burden on the subject. Assuming that any mental operation whatever occupies a fixed minimum interval, it is impracticable to use some logically possible method of computing y; for example, at $\eta = 4$, there is no chance of computing the components of y in isolation. δt is chosen by the experimenter to ensure that condition. Generally, there is more latitude than this, so that the subject may *either* build up *efficient* computing structures in his *Lev 0* repertoire *or* he can retain less efficient structures in an accessible (meaning, in this model, often replicated) fashion *or* both criteria may be satisfied: accessibility and efficiency. But a modicum of trade-off between accessibility and efficiency is permissible.

The temporal constraint also restricts the time available for building up or modifying the *Lev 0* repertoire and, assuming that any constructive operation also occupies a finite interval, time is a basic resource at the subject's or at the learning model's disposal. The restriction, in this case, is imposed by the between trials interval Δt (in contrast to the solution and response constraint imposed by the lesser interval δt).

These ideas, advanced in the very specific context of one type of experiment, will bear generalisation. For any skill whatever, it is reasonable to posit an advantage in favour of efficient computation (to secure this, some resource must be expended) and to posit as well a general *resource*, available in only limited supply. This resource is needed to apply, or execute, mental operations, to build them, and to retain them as stable entities in storage.

The model incorporates this generalised concept of a mental resource which is imaged as an hybrid of data storage *capacity* and of *effort*, and which is used up in the execution of *Lev 0* operations, or the operations that build and modify the *Lev 0* repertoire. 'Storage' is conceived as a material in which operators and strings or structures of them are embodied.

It will be assumed that *Lev 1* operations in the *Lev 1* repertoire can be written and retained unchanged throughout the experiment in a costless long-term storage (not unlike the long-term memory model of Atkinson (1969), and Atkinson and Shiffrin (1967)). But there are restrictions upon the embodiment of the *Lev 0* repertoire and one way of representing them is to suppose that the prescriptions for the structures in the *Lev 0* repertoire give rise to bits of physical computing machinery which decay in time. In so far as the machinery in which the *Lev 0* structures are embodied *does* decay over time, an existing structure must be maintained or reproduced by a specifically directed activity. For this purpose a *Lev 1* operator called 'reproduce' is one *mandatory* constituent of any *Lev 1* repertoire.

This view of mentation is unashamedly drawn from the field of cellular biology. In that sense, it is idiosyncratic. With this caveat, it does no harm, to think of operations (in mentation) as analogous to allosteric enzyme

systems organised on a surface or by mutual specificity, and of problem solving transformations as analogous to the catalytic transformation of metabolites. Pursuing this correspondence one stage further, the mental computing machinery must be maintained and reproduced by a feedback-controlled, and essentially symbolic, process akin to the repressor feedback-controlled process, which involves DNA loci, messenger RNA and ribosomal transducers that reconstructs the cellular enzymes from transfer RNA-tagged amino acids, and thereby maintains the organisation in a cell. To avoid confusion we should insist that this loose analogy has nothing whatever to do with the biochemistry of brains, with the part played by messenger RNA and specific proteins in information storage (as first proposed by Hyden, 1963) or with other, equally interesting physiological issues. It is a correspondence between cellular organisation and mental organisation and that is all.

'Effort' (equivalently 'work'), denoted λ, is used up by the application of either *Lev 1* or *Lev 0* operations. However resources are allocated, there is some restriction upon the effort that can be expended or the average work that can be done in a unit interval. This effort may be spent in solving problems (by applying *Lev 0* operations), or in constructing *Lev 0* structures and in maintaining them against decay (by the application of *Lev 1* operations). Using the notion of a *Lev 1* operator 'reproduce', which maintains stored structures in the *Lev 0* repertoire against decay, and the notion that the application of this operator itself costs effort, it is possible to represent a dynamic and replicative structure which may be likened to the psychological entity 'intermediate memory' (perhaps Atkinson's (1969) short-term storage; for example, as it is organised for word-list rehearsal).

The idea of decay, together with restriction upon effort or work, leads to a system in which resources must be allocated in a very definite fashion if the computing structure is to survive; generally, its survival depends upon the introduction of at least some 'cooperative' or 'superadditive' composition rules whereby a more 'efficient' structure is more economic to maintain. The organisation responsible for allocating the resources in the learning model is a resource allocation programme.

Any resource allocation programme imposes a limit upon the average number of *Lev 1* or *Lev 0* operations that can be applied in a given interval; briefly, upon the effort, λ so that say, $\lambda_{max} \geqslant \lambda$. There is also a minimum limit, $\lambda \geqslant \lambda_{min}$, which reflects the fact that operations must be applied to something, though not necessarily to the problems that are deemed to be relevant by the experimenter.

Many resource allocation programmes satisfy these conditions. For example, the resource allocation executive of the model in Figures 50 and 51 satisfies:

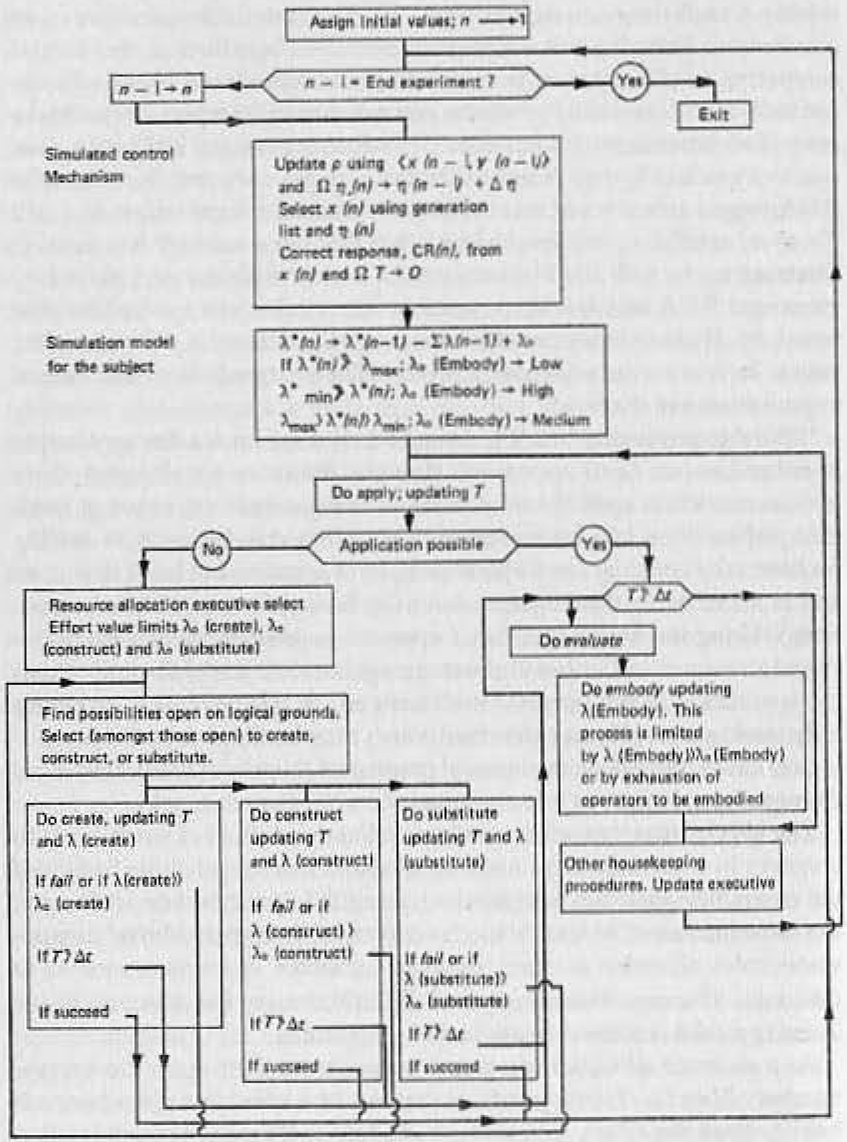


Figure 50 Overall organisation (simulated control mechanism and model for subject. Flowcharts for a single subskill version of the model (due to Feldmann, Lewis, Mallen and Pask) current in 1966–8.

(*a*) Problems x numbered as SN; 1–8, $x \in X_1$; 9–32, $x \in X_2$; 33–64, $x \in X_3$; 64 and above, $x \in X_4$. The program *decomposes* the problems into their constituents by rewriting rules based on the description in Figure 49.

(*b*) Response *components* of y designated by KP.

(*c*) Operators have *domain* and *range*; simple operators domain SN 1–8; *complex* operators, obtained via substitution, a multiple domain.

(*d*) *Range* is member(s) of KP (since Ω is one to one, any correct operation has domain of η elements).

(*e*) *Concatenation* produces *strings* via process of string *construction*.

(*f*) *Domain* of string: $SN > 8$.

(*g*) *Range* of string; set of operator names.

(*h*) Operators and strings have *lifespan* (trials to entry of a deletion stack) and a *utilisation* (their correct use record).

(*i*) Various stacks are used in program but two are distinguished as *operator store* and *string store*.

(*j*) $\lambda^*(n)$ is effort available at trial n.

(*k*) T is time taken; Δt is time permitted.

Notes on the main variants.

(*a*) Predictive control of effort using estimate of $\Delta\lambda^*$.

(*b*) For double subskill (or interference task) model there are two stimuli $x = x_A, x_B$ of which the control mechanism gives priority to x_A. *Other* subskill (interfering task) is assigned value, like a procedure, and x_B is selected for attention in the same way that procedure is selected.

On following pages: *Figure 51* Flowcharts showing details of model organisation: (*a*) *apply* operator procedure; (*b*) *create* operator procedure; (*c*) the most elementary type of concatenation (string construction) procedure ($x \in X_2$); (*d*) substitution procedure for constructing complex operators from existing strings and for replacing these strings; (*e*) outline of *Evaluate* response procedure and the *Embody* procedure (for updating *lifespans* so that operators/strings are retained in operator store/string store). Many variants were used. (*f*) Outline of resource allocation executive controlling the (relative) effort, λ assigned to different processes.

Figure 51(a)

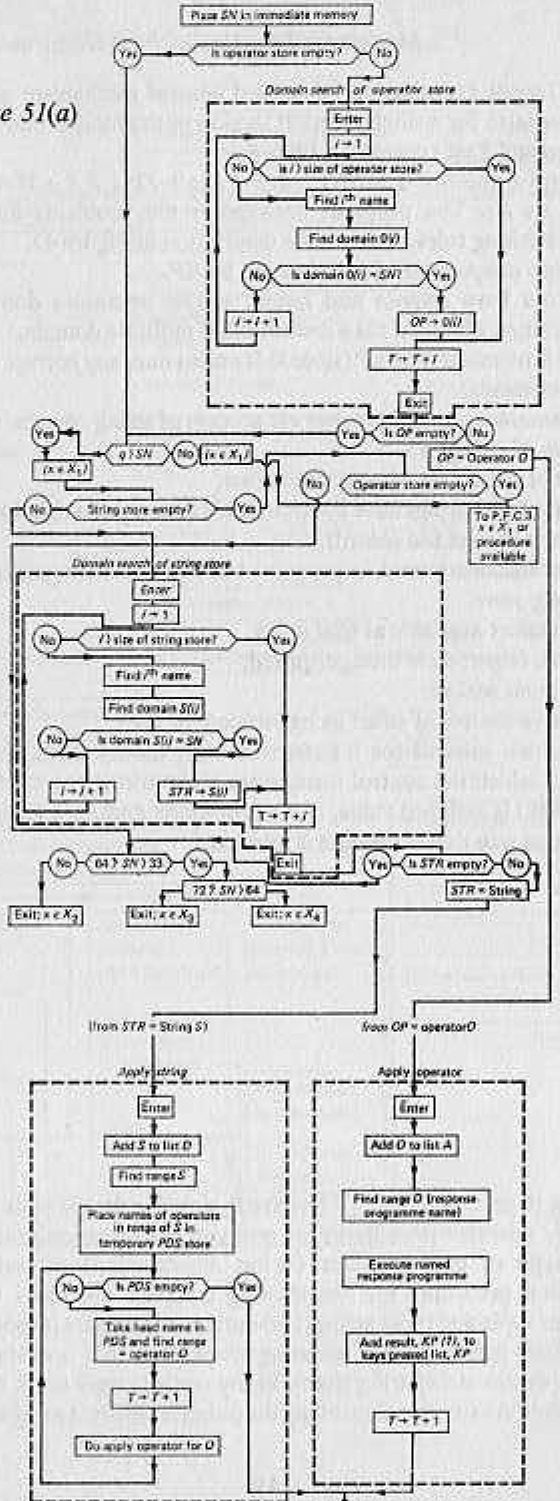


Figure 51(b)

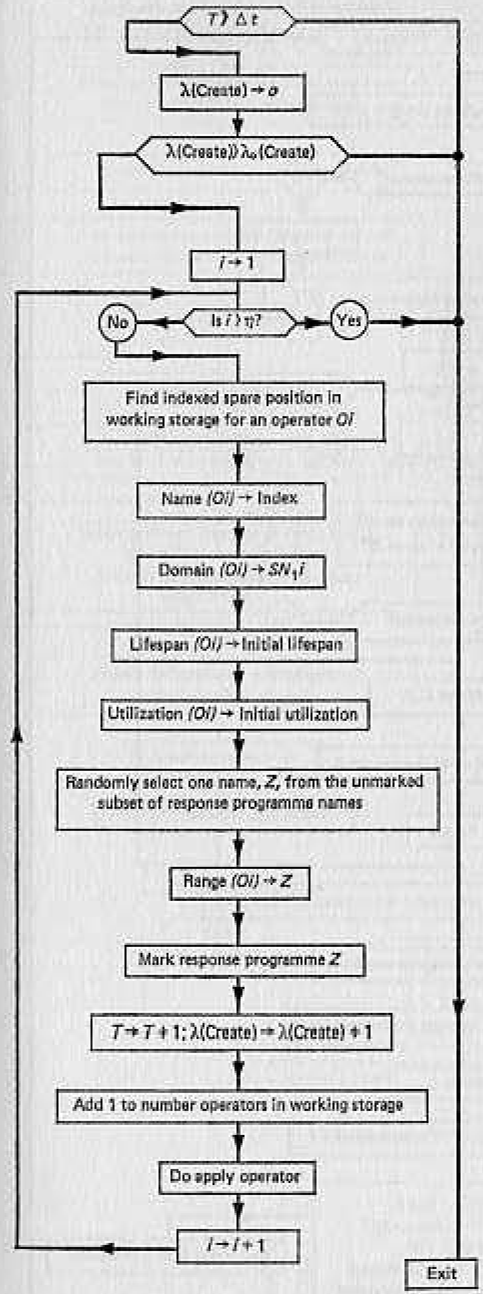


Figure 51(c)

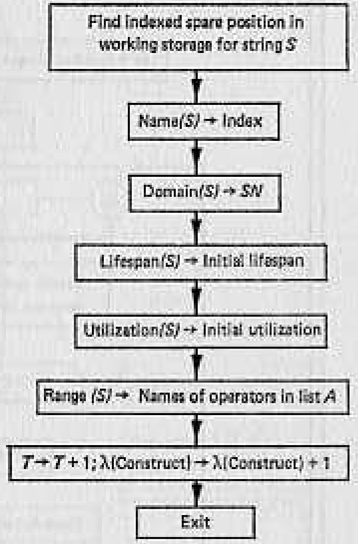


Figure 51(d)

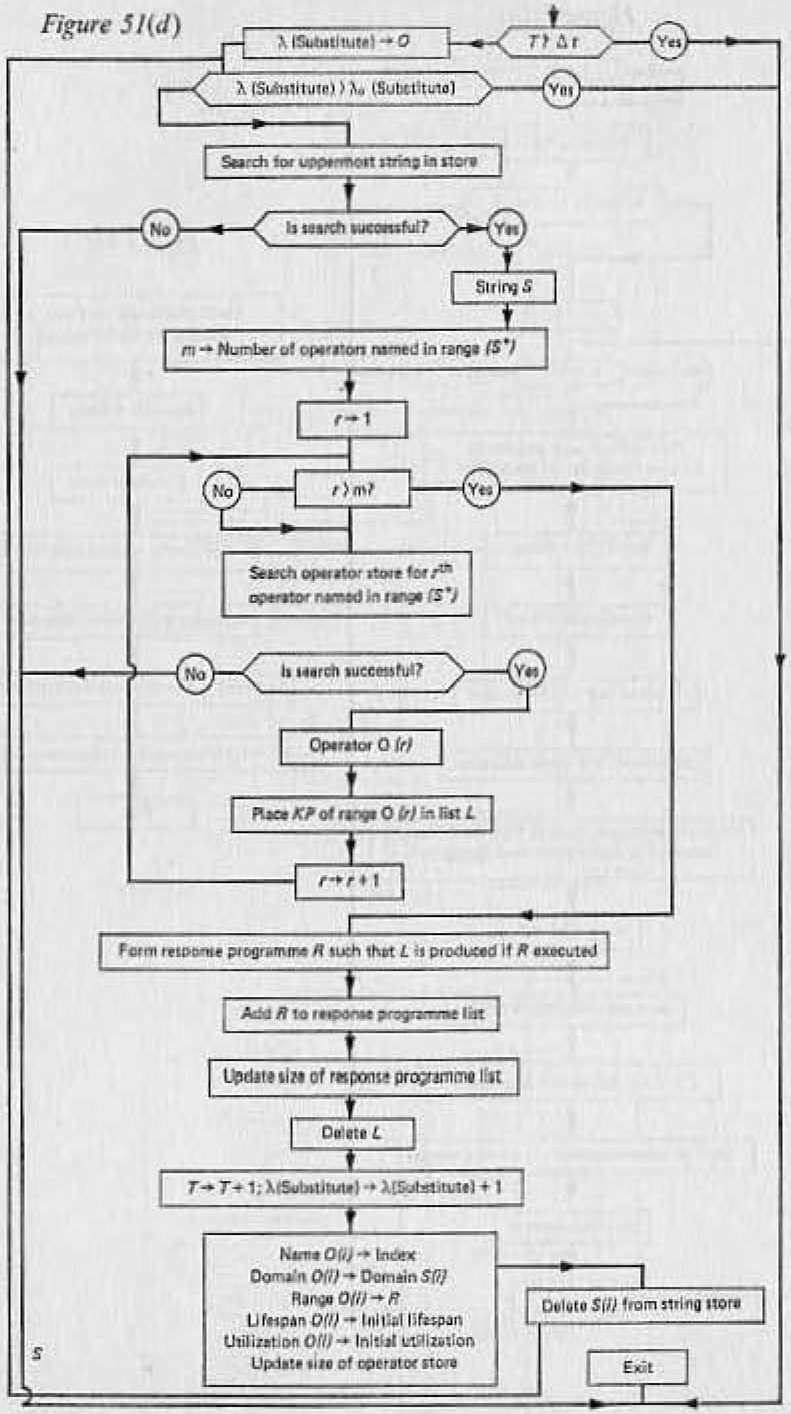


Figure 51(e)

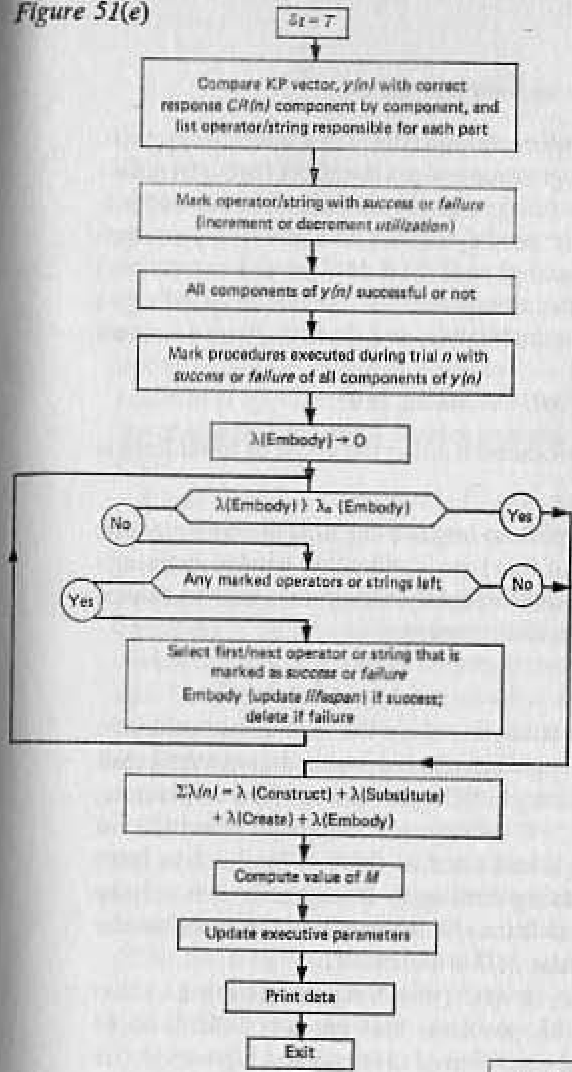
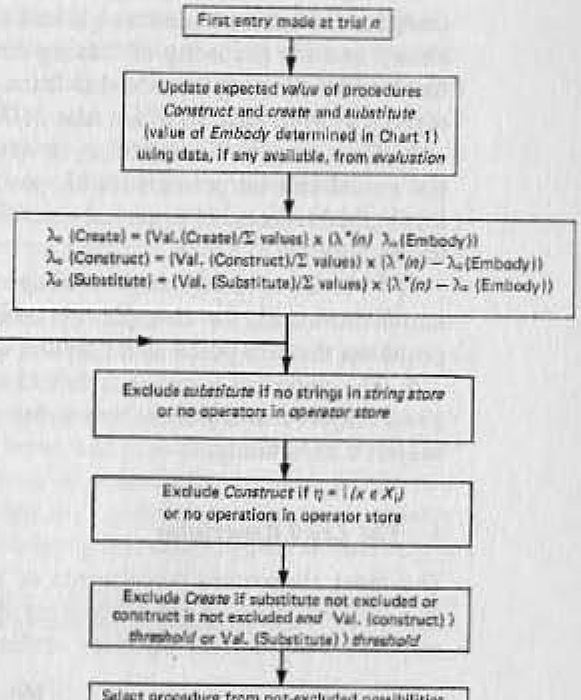


Figure 51(f)



1. The average value, λ^* of λ is constant, so that $\lambda_{max} > \lambda^* > \lambda_{min} > 0$.

2. The resource allocation programme assigns the effort (between different sorts of *Lev 1* and *Lev 0* operations) so that (a) at least some *Lev 1* operations are applied to construct or modify strings of embodied operators (hence, there is at least some learning) and, (b) if *MU* (model uncertainty) is a variable that increases whenever problems remain unsolved, that increases rapidly if a problem is unintelligible, and that otherwise decreases then

$$MU_{max} > MU^* > MU_{min} > 0$$

and an 'impasse' signal is being produced if either the upper or lower limit is contravened.

If a learning model able to interpret an impasse signal is governed by any resource allocation programme of this type it will either behave as though it was capable of curiosity or it will, if slightly elaborated, tend to change its attention and to reinterpret its environment.

In particular:

1. Learning is a process that tends to reduce the machine uncertainty, *MU* (interpreted as an uncertainty given a class of problems, regarding their solution) by applying and if necessary building structures of *Lev 0* operators.

2. The limit $MU > MU_{min} > 0$ reflects the empirical condition of underload. If this requirement is not satisfied there is 'nothing to learn about' and the absurdity of 'having nothing to learn about' can only be resolved by allowing the model to learn about something the experimenter regards as irrelevant; in which case *MU* is undefined.

3. The converse limit $MU_{max} > MU$ (which represents the fact that the model can be presented with problems that are too difficult to be intelligible) reflects the empirical condition of overload and high subjective uncertainty.

4. In order to satisfy these conditions the teaching model must control the environment, for example, by selecting the level of difficulty of the problems that are posed as a function of an estimate of *MU*.

5. The empirical correlate is that an adaptive teaching machine controls a real subject's environment by varying η as a function of its estimate of the subject's uncertainty.

4 The *Lev 0* Repertoire

The most elementary constituents of the *Lev 0* repertoire are simple *0 operators* which carry one component of an input into one component of a response. Any correct operator mediating a solution, carries the input

component into a correct response component. But an *0 operator* may, just as easily, lead to a mistaken response.

The next constituents are complex *0 operators* that carry pairs, triples, quadruples, etc., of input components into pairs, triples, quadruples, etc., of response components. Whereas simple *0 operators* might be applied (given appropriate tests and checks) to yield the correct response $y = \langle r_1, \ldots, r_c \rangle = \Omega(x)$ to a complex input $x = \langle s_1, \ldots, s_c \rangle$, a complex *c*-fold ($c > 1$) *0 operator* achieves this result at once. Both simple and complex *0 operators* can be created *de novo*; in essence, by guessing, but the chance of creating a correct complex *0 operator* in this manner is very small.

Finally, there are structures of *0 operators*, called *0 operator strings*, which are entities containing the addresses of *0 operators*. A *c*-fold *0 operator string* is applicable to an input $x = \langle s_1, \ldots, s_c \rangle$ for $c > 1$ and musters the simple *0 operators* needed to respond (correctly or not). A string and the operators it addresses may both and independently be mistaken; hence, mistaken parts of structure may be providently winnowed out. Strings may also be substituted to yield equivalent complex *0 operators* that solve the problem posed by $x = \langle s_1, \ldots, s_c \rangle$, at once. Using the string intermediary (if the resources are available to do so), the model is almost certain to build a complex *0 operator* that leads to a correct response.

The stringing operation is generalised to yield strings of strings[1] (and so on) and to yield strings of complex *0 operators*. A string of complex *0 operators* is also open to substitution, conversely, a string of one *0 operator* is counted as a *string*. With these interpretations, the data structure $l(n)$ of the *Lev 0* repertoire is shown for trial $n = T$ (the trial at which the skill is learned and performed with complete proficiency) in Fig. 52. The caption relates $l(T)$ to the previous discussion of problems and solutions.

Although *0 operators* are specified in a very simple fashion for the coding skill, the concept of $l(n)$ is open to an arbitrary degree of elaboration. For example, the simple *0 operators* are quite easily replaced by small finite-state machines (Fogel, Owens and Walsh, 1966; Pask *et al.*, 1968) in which case a string becomes a program of FSMs, and a complex *0 operator* a composite or restricted product, of FSMs.

Each *0 operator* or *0 operator string* in the *Lev 0* repertoire, $l(n)$, is associated with two variables called lifespan and utilisation. Both variables are used to simulate processes that could be realistically executed. Of the two, lifespan is given an initial value when an *0 operator* or a *String* is embodied in the *Lev 0* repertoire, $l(n)$; subsequently it is decremented at each step in

1. The notion of 'string of strings' was originally introduced by Mr G. L. Mallen who was responsible (see Tech. Rep. references) for programming one of this family of models.

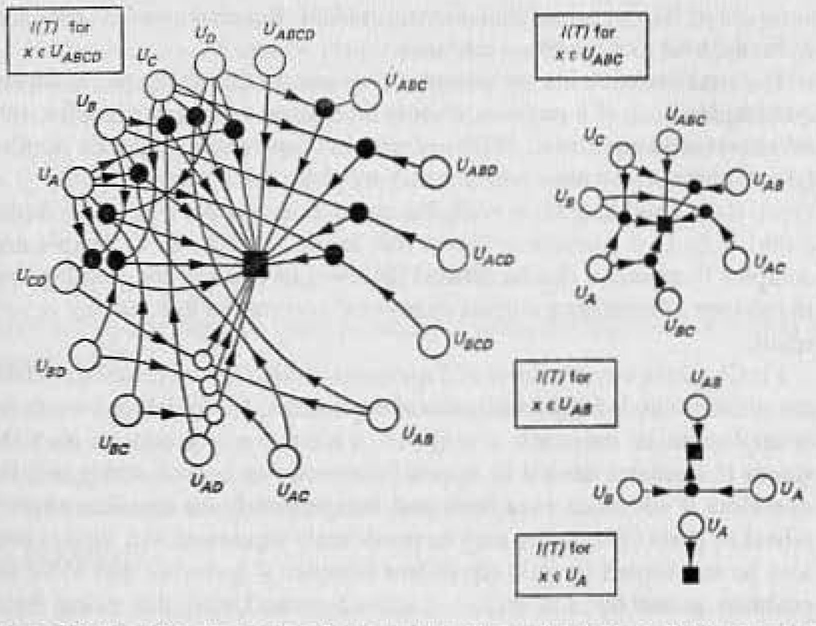


Figure 52 Each input, x, gives rise to a state of knowing because it is a member of some set U. Sets U are shown as nodes. A *problem* is posed by x, under rule Ω and is solved by applying a correct operation, a string of operations, or a complex operation (in turn, yielding other states of knowing). *Correct* operations are shown as directed arcs, which lead to a unique and central node (the *solution*); their application produces components of a response, y. The *description* of the correct operation paths is $I(T)$. To simplify the diagram $I(T)$ is shown for $x \in U_{ABCD}$ (representing $\eta = 4$) for $x \in U_{ABC}$ (representing $\eta = 3$) for $x \in U_{AB}$ (representing $\eta = 2$) and for $x \in U_A$ (representing $\eta = 1$).

the execution of the model to simulate decay and the operator or string is deleted from $I(n)$ if its lifespan becomes zero. Deletion of a string does not imply deletion of the *0 operators* it addresses nor does deletion of all the addressed *0 operators* imply the deletion of a string. Conversely, the value of lifespan is incremented if the operation *reproduce* is applied to the *0 operator* or string in question to simulate its replication. Utilisation is incremented whenever an *0 operator* or string is applied and its value is decremented, to a minimum of zero, for each trial.

5 An Account of the *Lev 1* Repertoire

The first *Lev 1* operation is APPLY the shortest applicable string of *Lev 0* operators in $I(n)$ to the problem denoted by an input. APPLY is only realised if such a string of operators exists in $I(n)$. (For example, for problem *a*

(Fig. 53) when it leads to the application of a single *Lev 0* operator or for problem *b*, when it leads to the application of a string of operators. APPLY is not realised for *d* or for *c*.) In so far as APPLY is realised, it is given precedence and it leads to a solution of a problem and to the partial satisfaction of the goal. If the model is presented with a problem such as *d* or *c*, the model is unable to APPLY and such a problem is interpreted as a sign invoking the *Lev 1* operations described below.

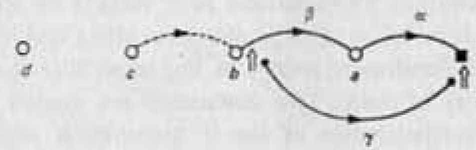


Figure 53 States of knowing (arbitrarily chosen ones) shown as nodes *a*, *b*, *c*, *d*. Operators are shown as directed lines, α, β. The dotted line indicates that concatenation onto the end of string (α, β) is possible, thus interpreting *c*; however, the chance of randomly constructing an operator to interpret *d*, though finite, is small. Double arrows on operator γ indicate substitution process.

The first of these is CONCATENATE. To CONCATENATE, the model searches an operation list for a *Lev 0* operator which may be applied to the problem posed by an input, or part of it, and which may also be adjoined to a string of operators in $I(n)$ which terminates in a solution state. The gap which the concatenation process is allowed to span is restricted (essentially, by the length of list search and the resource allocation programme). In Fig. 53 it is assumed that the gap between *b* and *c* allows for concatenation although that between *b* and *d* is too great. The model would have a vanishingly small chance of randomly CONSTRUCTING an operation which leads from *d* to *b* (and thus to *a* and the solution) within the interval allowed for solving the problem.

CONCATENATE is always followed by a *Lev 1* operation EMBODY the selected *Lev 0* operation as an operator in $I(n)$. Further EMBODY is always followed by APPLY and the persistence of the string of operators formed by CONCATENATE is indirectly dependent upon the receipt of a 'knowledge of results' signal (the result of an external test) which evaluates the application of the concatenated string of operators. Since concatenation is controlled by 'knowledge of results', $I(n)$ chiefly consists in strings of operators that lead to correct solutions.

The next *Lev 1* operation is SUBSTITUTE. The model performs this operation, if the resource allocation programme allows it to do so, whenever it has completed an application or has found that application and concatenation are impossible. Substitution consists in the placement or embodiment in $I(n)$, of a *complex 0* operator performing the same (or in principle a similar) job to an existing string of simple *0 operators*. Substitution using a complex *0 operator* leads to the construction of shorter and more efficient *Lev 0* structures. The basic substitution process is exemplified in Fig. 53 by the placement of γ in parallel with the string of operators, α, β. Since APPLY selects the shortest applicable string of operators γ will be used upon subsequent occasions and α, β, will not be used. As below, this may lead, indirectly, to the decay of α and β.

The *Lev 1* operation REPRODUCE is a variant or specialisation of SUBSTITUTE. It is called as often as resources allow and 'reproduces' any *Lev 0* structure (operator or string) in the sense that it substitutes the original by a copy of itself. Two comments are needed (a) a string is REPRODUCED independently of the *0 operators* it addresses and (b) REPRODUCE is simulated in the learning model by an incrementation of lifespan which is decremented, each unit interval, to simulate *decay*.

For the two subskill tasks where there are *Lev 0* repertoires $l_1(n)$, $l_2(n)$ the *Lev 1* repertoire is augmented by operations akin to substitution that carry out transfer of training; they take parts of $l_1(n)$ and place them in similar positions in $l_2(n)$ or vice versa.

Most versions of the model (including the version shown in Fig. 50 and Fig. 51) incorporate a process for attending to and analysing detailed knowledge of results (i.e. data regarding the rectitude of parts of a response obtained by comparing these parts with parts that are already established or a component-specific corrective input). In the case under discussion, though not generally, the cost of applying this operation is zero.

Finally, implicit in the SUBSTITUTE operation, there is an operation, DESCRIBE. In order to substitute, the model must know what operator or string of operators in $l(n)$ is similar to an operation on its operation list. To do so, it must describe the contents or some of the contents of $l(n)$. The elements that are described are coded like members of the operation list, and their similarity with other members is determined. Hence DESCRIBE is an operation for adjoining members to the initial operation list (for example, the coded description of the string of operators α, β, which turns out to be similar to γ).

Notice that the derivation of γ in Fig. 53 may either be conceived as the matching of an existing code 'γ' with the description 'α, β', or as the production of γ from α, β, by a rule applicable to descriptions of operator strings, ostensively defined by embodiment in $l(n)$ (recall that most of the strings of operators in $l(n)$ lead to correct solutions).

The execution of any operation decreases the amount of effort, λ, that is available. Conversely no operation can be applied (or even instituted by the resource allocation programme) unless the necessary effort *is* available. Effort is incremented at the conclusion of a trial and may be saved or carried over from one trial to the next.

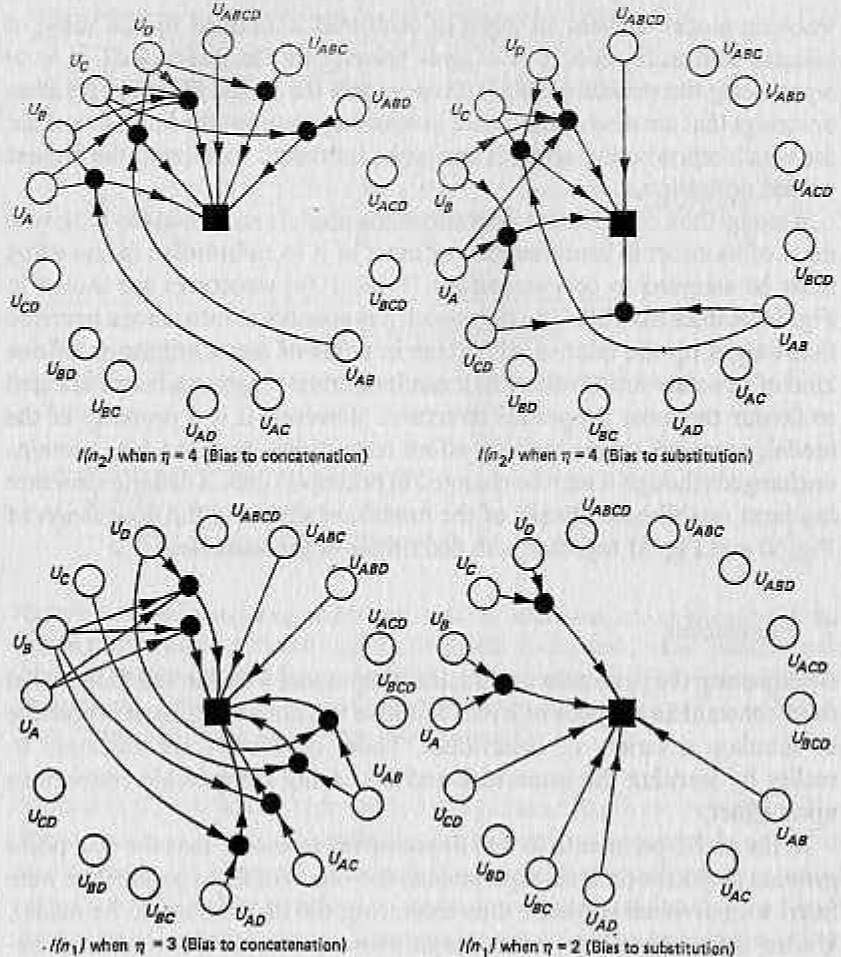


Figure 54 Typical structures of $l(n)$ shown on left for one successful mode (bias to concatenate operators) and on right for another mode (bias to substitute operators). In each case $n_2 > n_1$ and the sequence is associated, realistically, with increase in η as learning proceeds; with construction of some complex operators and deletion of others. The model does not (run normally) have the effort to use and apply the processes that muster four or even three operators. Further, unless redundant complex operators are deleted (shown in Bias to substitution case) the model does not have effort to embody essential operators.

6 Operation of the Learning Model

In the interval δt, or until a response is produced within that interval, the resource allocation programme gives priority to the *application* of *Lev 0* operators and strings in order to solve the problems posed when the teaching model delivers an input at each trial n. Guided by the value of lifespan and utilisation it also gives priority in the interval $\Delta t - \delta t$ to *reproducing* the structure in $I(n)$; it reproduces the lowest lifespan operators or strings that are also utilised and, in selecting amongst the limited number for which reproductive effort is available, it favours those with the highest valued utilisation.

Amongst the constructive operations the model can be biassed to devote most of its effort to *concatenation* or most of it to *substitution* (some effort must be assigned to concatenation). Typical $I(n)$ structures are shown in Fig. 54. Rather than biassing the model, it is possible to introduce a heuristic that weighs up the relative advantage in terms of mean utilisation of one kind of structure or the other. As a result of this evaluation a bias is selected to favour the most propitious structure. However, it is a property of the model, executed under realistic effort restrictions, that the bias remains unchanged (though it *may* be changed in principle) once a definite structure has been established. Details of the model are shown in the flow charts of Fig. 50 and Fig. 51 together with definitions of the variables.

7 Simulation

By adjusting the parameters of the learning model with the teaching model fixed constant as a replica of a real adaptive teaching machine, it is possible to simulate a variety of behaviours. Those of interest are anchored to reality by learning the same task and by having comparable restrictions upon effort.

In the real experiments Δt and δt are varied to ensure that the task poses genuine problems (in later experiments, the values of these parameters were fitted to individual subjects, thus enhancing the distinctions to be made). Under these conditions, all subjects show a sharply discriminated behaviour (a) in terms of their susceptibility to overload and (b) in terms of the distribution of complex response component latencies. The latency patterns serve as an indication of the problem solving method (taking individual subjects and examining response to each input separately, there is an all or none distinction between the learning patterns referred to as stringing and grouping in Chapter 8). The former (real-life) pattern corresponds to an $I(n)$ (in the learning model) biassed to concatenation (Fig. 54) and the latter to an $I(n)$ biassed to substitution; moreover, if the relative application latencies of the model are recorded, these closely replicate the real-life response latency profiles.

It is still possible to simulate many behaviours by adjusting the learning model parameters. Taken out of context such exercises have doubtful value. Consequently one phenomenon has been isolated as worth illustrating

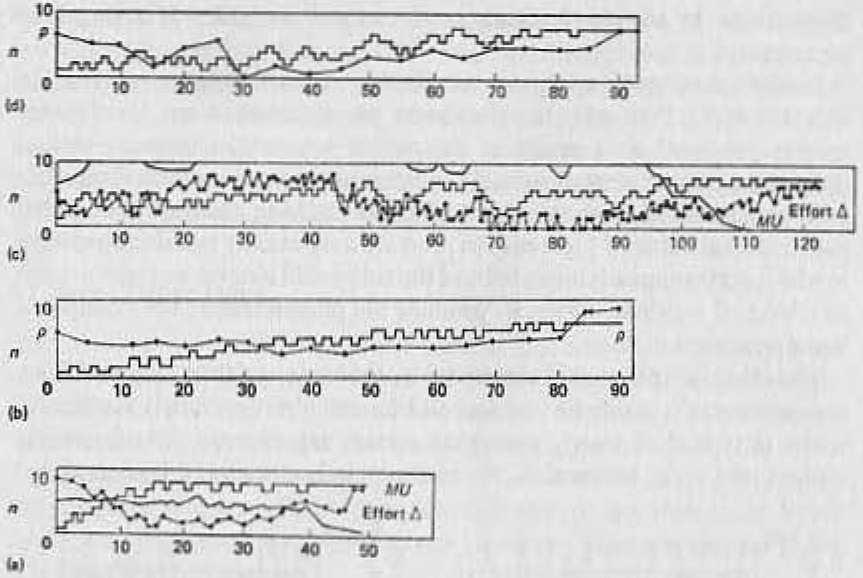


Figure 55 Data comparing stable and instable behaviour in a subject and in the learning and teaching model: (*a*) stable model (same task, same parameters); (*b*) stable subject (same task, same subject); (*c*) instable model (condition induced by overload); (*d*) instable subject (condition induced by overload). ξ = Steady State Level; ρ = Proficiency Index (see Chapter 8).

because it is particularly repeatable and is interesting in its own right. The phenomenon is a form of instability due to overload (apparently any kind of overload; for example, overload by an auxiliary task or by a small reduction in δt). The overload is most conveniently induced by a small reduction in a parameter ξ, common to the adaptive teaching machine and its veridical image, the teaching model, without an alteration in the learning model parameters. Some data are shown in Fig. 55 which is a comparison of stable against instable behaviour in a subject, and stable against instable behaviour in the learning and teaching model. The real system characteristics are typical.

The instability is manifest (Fig. 55) as an oscillatory fluctuation in η and a departure of ρ from its steady-state value. The curves for the learning/teaching model and for the subject/machine system are very similar in form. The course of the instability is evident in the model only; a periodic depletion of effort (recorded as an effort surplus, Δ, rather than the effort level, λ). Moreover, subjects who give a retrospective account of their experience consistently report periods of high uncertainty alternating with periods where they have a grasp of the task. It is tempting to liken these fluctuations to the fluctuations in the model variable, *MU* (machine uncertainty).

Similar comments apply to the double subskill system outlined in Chapter 8. In this case, the dominant phenomenon is an interference barrier produced as a result of destructive interaction between mental operations proper to each subskill, which must be surmounted if the skill is to be learned as a whole. If the adaptive teaching machine is operated near a critical value of ξ it is easy to produce a repeatedly instable condition in which performance is impaired and the entire skill is never learned; almost any kind of overload serves to produce the phenomenon; for example, a small reduction in δt.

The effect is replicated in the model by the same variation. Figure 56 is a comparison of a stable but critical and an unstable (overload) simulation which is typical of many, somewhat varied, experiments. Interference is evident (the η_1, η_2 interaction, for example) in both cases; it is engendered

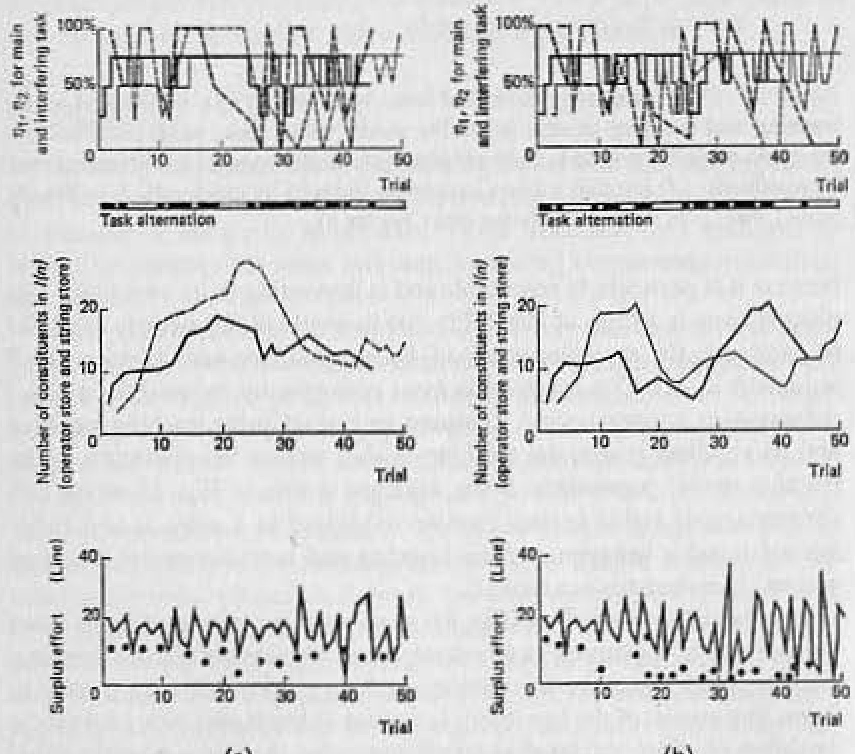


Figure 56 Comparison of: (*a*) stable and (*b*) instable simulation for double subskill conditions.

by the task specification and is unavoidable. However, no significant learning takes place in the unstable case. The direct cause is a failure to accumulate sufficient common structure in $l_1(n)$ and $l_2(n)$ (solid line and dotted line), one structure in the unstable model ($l_1(n)$) is abraded whilst the other structure ($l_2(n)$) is built up and vice versa. This degradation can be traced back to an indirect cause, namely a periodic depletion and oscillation of surplus effort evident (lowest graph) in the unstable case only.

8 Some Other Data

The learning model has also been used in connection with the type of situation noted in Chapter 10 where the task is isomorphic with the single subskill code learning task. The environment is partitioned to permit the specification of several distinct learning strategies of which the most important are jointly characterised by an incidence of 'strong' response latency patterns and a serial learning strategy, and (on the other hand) by an incidence of 'grouped' response latency patterns ('clump responses') and a holistic learning strategy.

Thus employed, the model demonstrates the salient distinctions in the experiments, not so much in terms of replicating the statistical results but by furnishing a mechanism to explain the empirical differences, quite generally. The findings are summarised as follows.

1. An effect, later called *cognitive fixity*, leads to entrenchment of a structure in $l(n)$; either the bias to concatenate structure or the 'bias to substitute' structure of Fig. 54. These structures are incompatible in the sense that effort spent in replicating one structure fails to replicate, or even degrades, the other structure.

2. A 'bias to concatenate' structure leads to string response latency patterns.

3. A 'bias to substitute' structure leads to grouped or 'clump' response latency patterns.

4. A 'bias to concatenate' organisation can be induced (in an otherwise unbiassed learning model) if it is driven by a serial teaching strategy (Chapter 10).

5. A 'bias to substitute' organisation can be induced (in an otherwise unbiassed learning model) if it is driven by a holistic teaching strategy (Chapter 10).

6. If the learning model is initially biassed by any means and it is instructed by a teaching model equipped with the converse teaching strategy, learning is inhibited and the model's activity closely resembles the empirical phenomenon of *mismatching*.

9 Criticism and Extensions of the Model

Learning and teaching models of this kind are readily made more complex and able to carry out cognitive operations that deserve the name 'problem solving'. For example, the simple *Lev 0* repertoire can be filled out with a program suite such as Newell Shaw and Simon's *General Problem Solver*, most succinctly reported in Ernst and Newell (1969), by a very different process such as Quillan's (1969) *Teachable Language Comprehender* or by any of the artificial intelligence programs reviewed in Minsky (1968). As an intermediary step there are many more specific programs such as those reported in Hunt, Marin and Stone (1966), Reitmann (1965), or Feigenbaum and Feldman (1963). By the same token the *Lev 1* repertoire can be filled out with operations for rewriting and amending these programs (the more elaborate exemplars, such as the *General Problem Solver* and *Teachable Language Comprehender* have these constructive operations already incorporated).

The distinction between *Lev 0* and *Lev 1* models is best made by programs that do incorporate such a facility. For example, Pereira (1973) has established a heuristic serial procedure for problem solving of a very general kind based upon an information measure (Chapter 1).

There are, however, several outstanding deficiencies which are more obtrusive if the simple model is kept in mind and are not eliminated simply by making it more *complex*. The specific elaborations of the last paragraph do not remedy the defects and an arbitary increase in complexity, *per se*, is utterly ineffectual.

1. Certain processes which might be executed are in fact imitated. The outstanding example is *reproduction* which, in this model, is sham (quite deliberately so, for the sake of computational efficiency). It *need* not be a sham, of course (Chapter 3), and it is vitally important that it should not be if the model is to penetrate further into the nature of cognitive processes. For, on seriously enquiring what a concept *is* or an intention (i.e. a goal) *is* or, first and foremost, what a memory *is* one answer turns out to be 'they *are* species of reproductive process' (in the sense of the theory of 'self reproducing automata' and its derivative disciplines). A memory, for example, is a procedure that reproduces a concept (itself a reproductive procedure); a point of view that is developed in Chapters 10 and 11.

Insightful readers may have noted that *Lev 1* operations and *Lev 0* operations figure as the components in one of Von Foerster's finite-function machines; the *Lev 1* repertoire acts upon the state sets of the 'machines' in the *Lev 0* repertoire. By a mild but useful extension of Von Foerster's theory the *Lev 0* operations are conceived as automata (not necessarily finite) and the *Lev 1* operations as reproductive automata that reproduce them, possibly with variation, mutation, recombination and the like, to generate an evolutionary process called *learning*. The point is that such a process is *imitated*, to an arbitrary degree of complication, by the model under discussion. With certain important caveats such a theory is *executed* by the models discussed in Chapter 3.

2. A process of this kind inhabits a very special environment *in* which it is reproduced. One candidate is a relational network (mooted earlier) together with a description of the related relations in terms of *what may be done* to bring them about or satisfy them and *what may be known* or learned (how one relation can be constructed by applying suitable operations to several others, also in the network). The whole notion of *knowability* or *memorability* is glossed in the present model (the reader may have felt uneasy over the inherent memorability of relations like $\mathscr{R}_1$, $\mathscr{R}_2$, that determine interfering subskills). To remedy this defect, canons of the following kind are mandatory. A domain of relations is knowable if (a) the relations in question are brought about by reproducible *Lev 0* structures, (b) there is a description scheme which allows the model to construct and reproduce them, and (c) each relation can be instanced in some world (for example, by constructing an artifact, doing an action, or writing a computer program that is executed).

Of these requirements (a) and (b) are commonsense restrictions that have recently been given a formal stature. (c) appears, at first sight, somewhat different in kind. It is not. The basic requirement is executability; a correct solution to a problem is a solution that can be executed to bring about or stabilise a relation.

3. The learning which is modelled takes place as part of a learner/teacher interaction. In the model as it stands, the learning and teaching parts are arbitrarily separated by two constraints (those of the last chapter), that stem from *imitating* rather than *executing*. The first constraint is the imitation of language rather than its realisation. It would not be difficult (even in the present framework) to make a better imitation; that is, to convert the interaction into a transaction scheme involving a formal *syntactic language* where (as in Chapter 4) a command form is taken to be interpreted as a command and a question form as a question, for metatheoretic reasons (the language game, the communication game) that are beyond the formal scheme. More than this is needed. As a minimum a *genuine* (not imitation) language involves command and question transactions that are *addressed* to pronominalised recipients for purposes (the pragmatics of the language) and the consequences of which are interpreted (the semantics of the language).

Such a language can be (and recently has been) provided in the real system, i.e. the mechanical 'teacher' can be furnished with the equipment to talk thus to a real subject who has the purpose of 'learning about' the same environment (Pask and Scott, 1973; Pask, 1973b). An immediate consequence is that nearly all operations (for example, the chains of explanation that are the linguistic forms of reproduction) become distributed, in the sense that parts are executed in the 'learner' and parts in the 'teacher' ('distributed' execution, as in Chapter 5).

4. The second constraint that comes from imitating rather than executing, is closely related to the doubts and difficulties aired in Chapter 5. The learning model and the teaching model are synchronised by edict, so that the order relations amongst the various processes is dependent upon factors outside the model itself. It is generally true that any systematic operation does entail synchronisation (or partial synchronisation). But, in real teaching and learning, the synchronisation is not built in from the outside. Learners and teachers are substantially asynchronous apart from the communication they engage in; they *become* synchronised *because* of it, or, viewed conversely, the communication (information transfer in Holt's (1968) and Petri's (1965) sense of the word) is a symptom of local synchronisation.

No real 'human brain' versus 'computing machine' issue is involved. A machine (though not a standard machine) can execute the requisite procedures. A brain which is a non-standard biological machine *does* so. But would the operation of a working 'model' that executes these procedures be deemed a model or a reality? It is not in any ordinary sense, a *simulation* though its operation may be 'simulated' by dint of the gambits (splitting processes into short segments, storage of intermediary results and random selection of one member from a set) that are usually employed to regularise partial synchronisation of fuzzy processes.

5. Systems of this type operating in the domain of intellectual learning are considered in the next volume and introduced towards the end of Chapter 10 and in Chapter 11. Very similar work has been done independently by Dexter (1972) in concert with Seigmann.

But, as promised the essential properties are captured by motor learning no more sophisticated than the transformation/coding/skill. For example, Scott (1963) has written a stripped-down model for a 'one handed' typist learning both *concepts* such as keyboard descriptions and *operations* like complex finger movements; it is a fairly elaborate piece of programming because the reproductive operations are genuine (not faked, as they are in the model under discussion). As a result, the model demonstrates the phenomena (such as the establishment of description schemes, their disappearance

from awareness as the skill becomes automatic and the motor operations are executed by many processes acting in parallel) that were observed during a lengthy and detailed investigation of teleprinter operator training (Pask, *et al.* (1969a) or Scott and Pask (1973) in press).

7 The Steady State or Null-Point Technique

One kind of relativistic system (Chapter 4, section 2) is a steady-state or null-point system. It is conveniently exemplified in the area of individual human psychology though similar techniques are used in connection with other kinds of observation in chemistry, physiology and sociology.

According to the classical rules of experimentation, a human subject is isolated as far as possible from extraneous influences and presented with an environment in which he performs a task; generally, a sequential task in which he responds to a series of stimuli. The subject's response is observed as a dependent variable. The task parameters are held constant during an experiment but between experiments some of them are changed, as independent variables, in order to examine the effect of different treatments.

These rules are tailored to fit a mode of thinking in which the subject and the environment are conceived of as finite-state mechanisms. These finite automata may be probabilistic devices; the subject certainly is. But, as an article of faith, the probabilistic automaton may be embedded in a much larger deterministic automaton. Of course, there is no suggestion that the structure of the subject automaton is known. The experimenter might be 'completely ignorant' in the sense that he looks at a 'black box' with input (including parametric input) and output. In *that* sense the experimenter has *no model* (Skinner's (1969) contention, for example). On the other hand, it can be argued that a very definite model exists in so far as the experimenter believes all that *may be known* about the subject is expressible by filling out the 'black box' as a (probabilistic) finite-state machine (or, the equivalent in continuous mathematics, as a generalised transfer function). The argument depends, of course, upon the identification of input *states* and output *states*. It is generally supposed that a stimulus is an input state (meaning 'the only relevant change in input state is the change indexed by the stimulus'). By the same token (and with the same qualification) a response *is* an output state.

The difficulties over this picture of things can be looked at from more or less fundamental points of view. For example, we might question the idea that the *goal directedness* that impels the subject to fixate attention on the

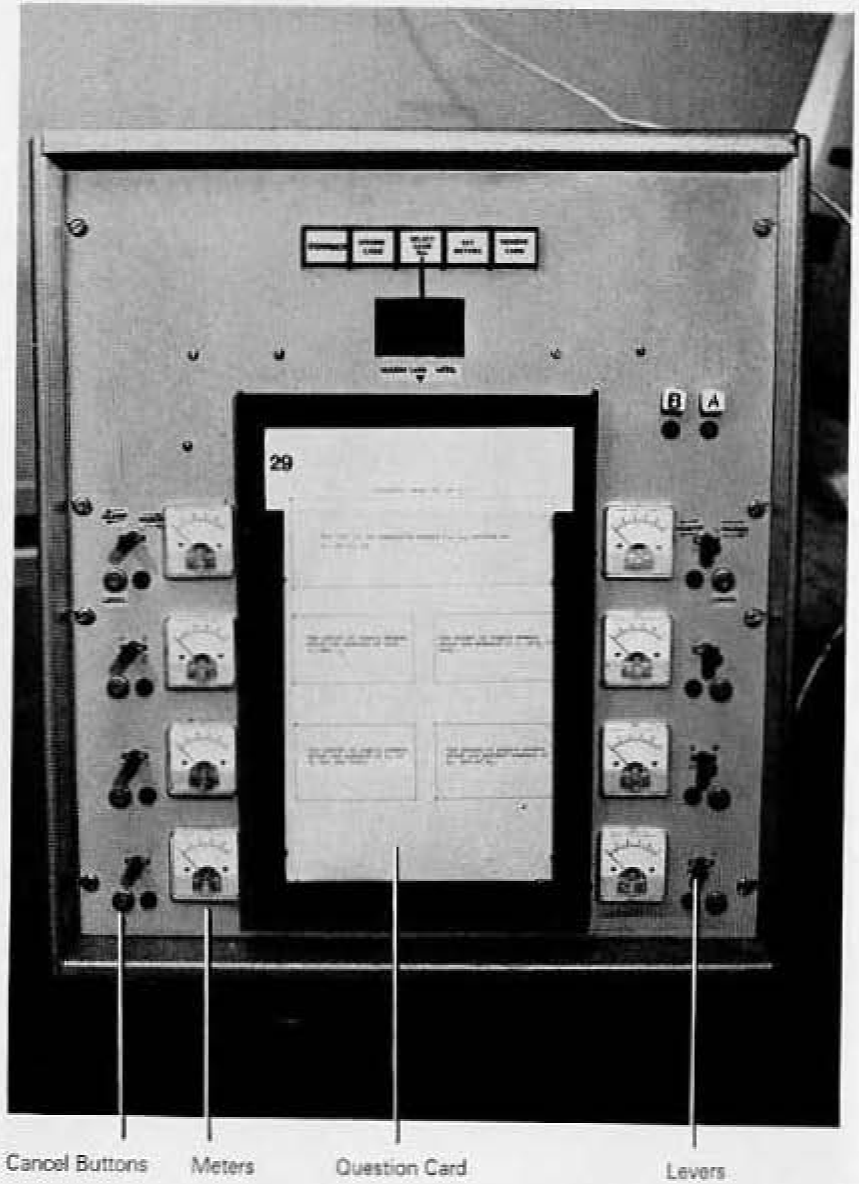
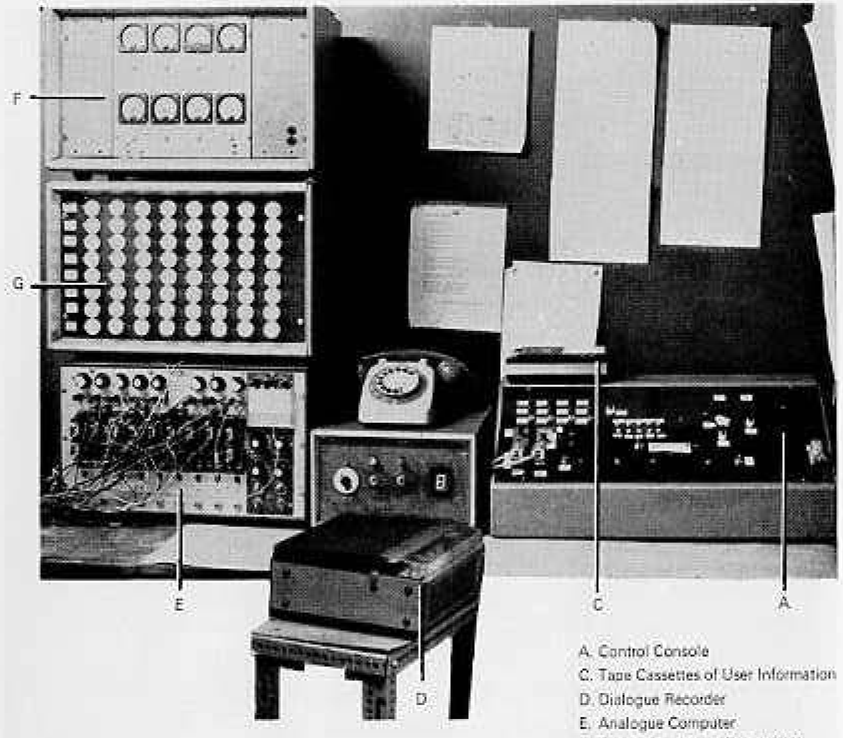


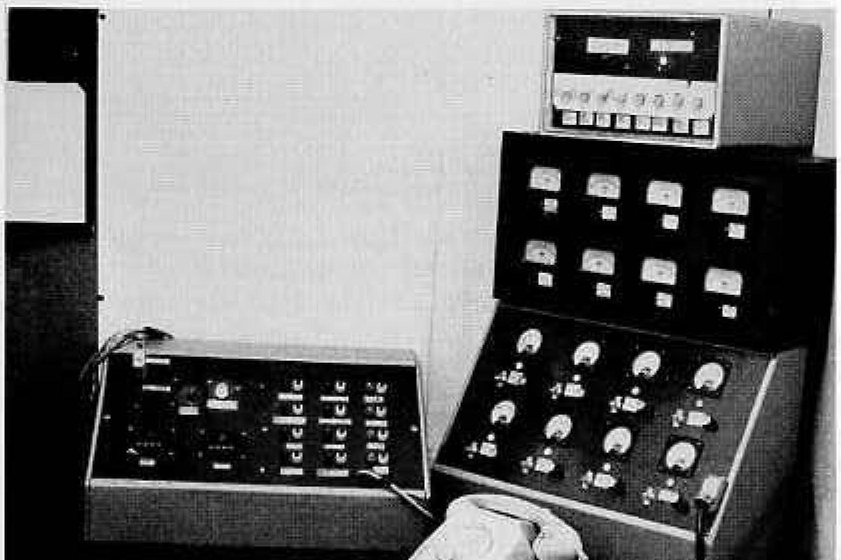
Plate 1 BOSS (Belief and Opinion Sampling System).

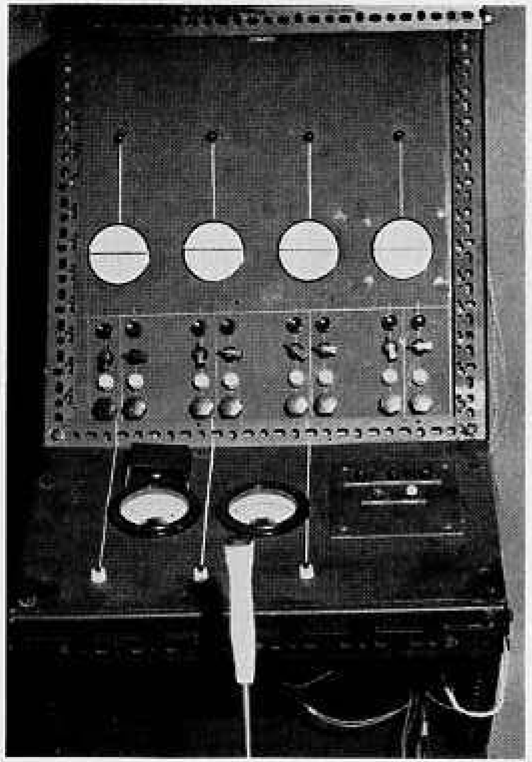


A. Control Console
C. Tape Cassettes of User Information
D. Dialogue Recorder
E. Analogue Computer
F. Respondent Prediction Meters
G. Information Input To Computer

Plate 2a Purchaser decision simulator: experimenter's console.

Plate 2b Purchaser decision simulator: respondent's console.





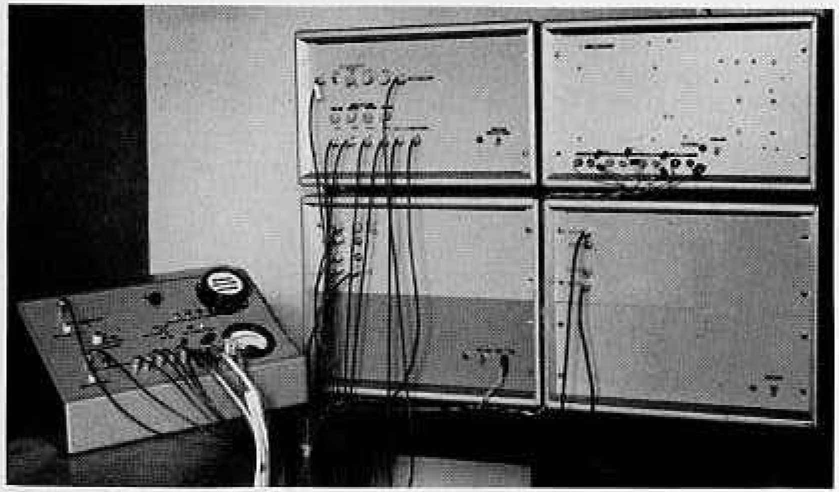


Plate 3a Compensatory tracking: subject's console.

Plate 3b Compensatory tracking: experimenter's console.

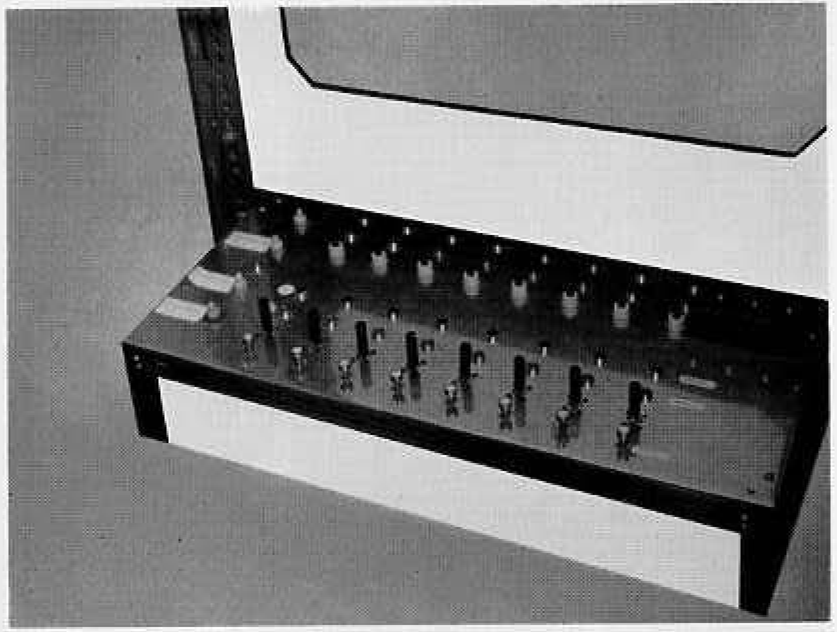
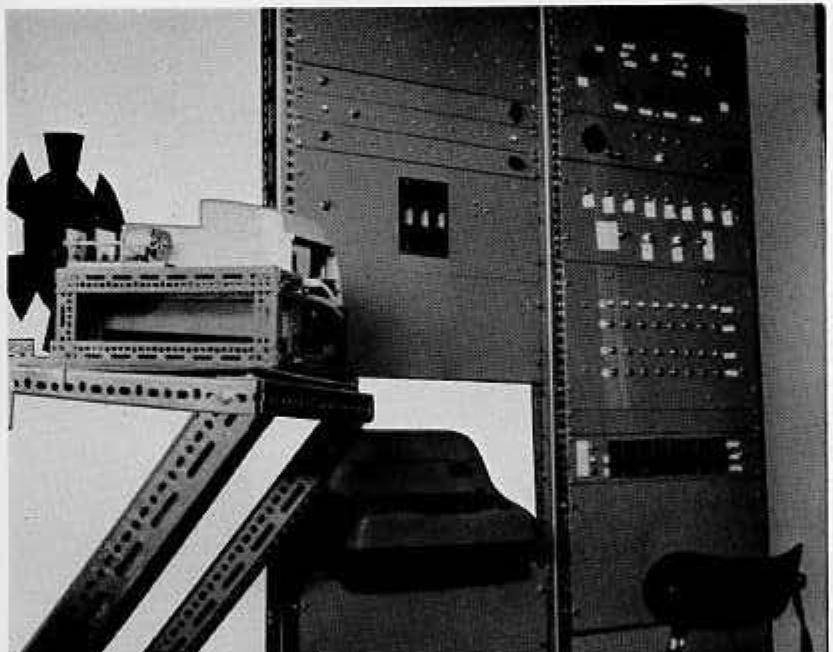
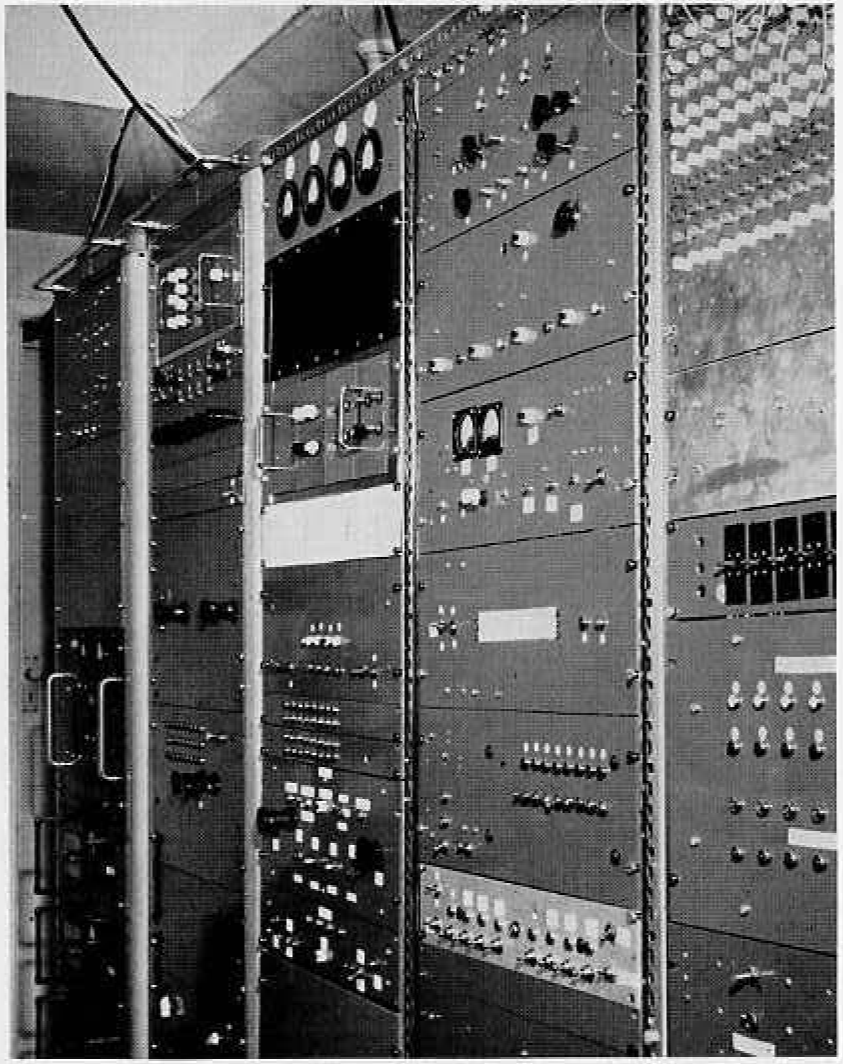


Plate 4a Attribute selection task: subject's console.

Plate 4b Attribute selection task: experimenter's console.





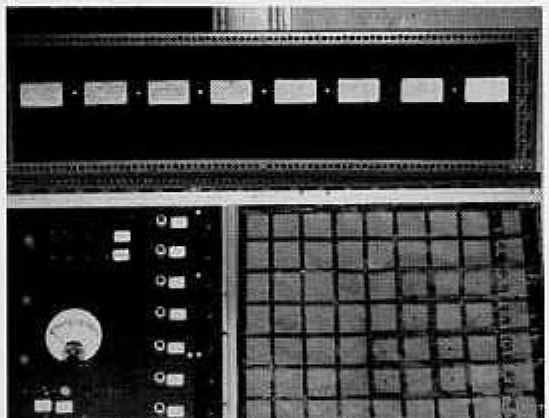


Plate 5 Special purpose computer for adaptively controlled experiments.

Plate 6 Target interception task: subject's console.

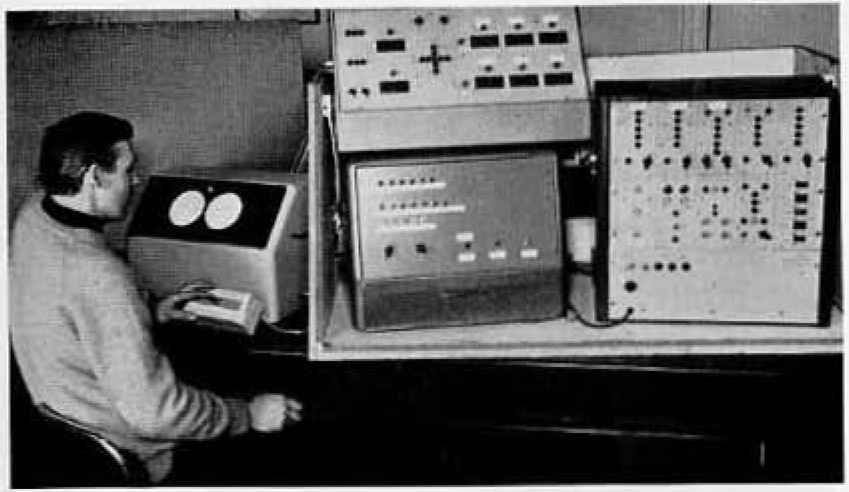
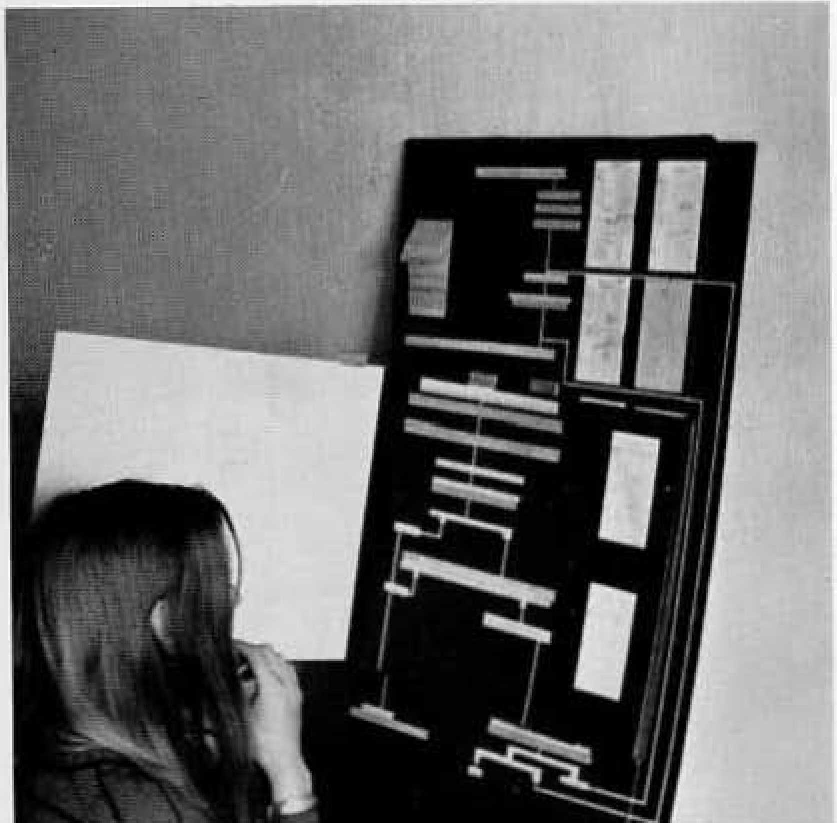


Plate 7 Adaptively controlled perceptual discrimination task with control equipment.

Plate 8 Subroutine flow chart training instrument for a clerical task.



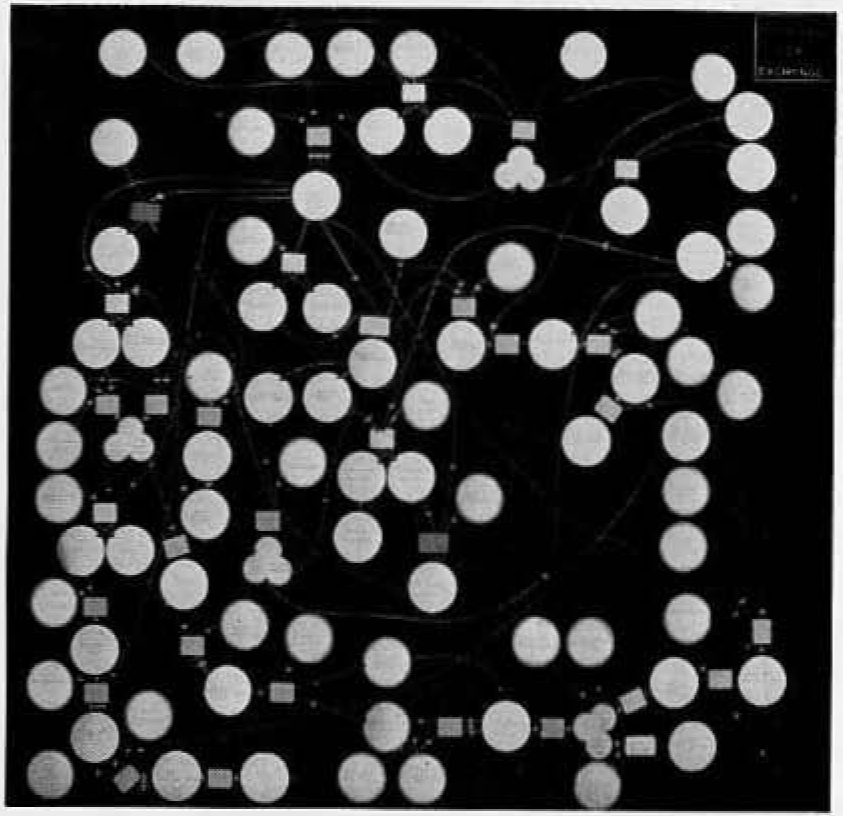


Plate 9 Representation of the process of carrying out a clerical task: that of processing a customer's order for the installation of some equipment. Circular nodes represent conditions; boxes represent transitions; 'cloverleaf' nodes represent conditions requiring information from outside the system in order to determine the next transition.

Plate 10 Subject free learning a zoological taxonomy.

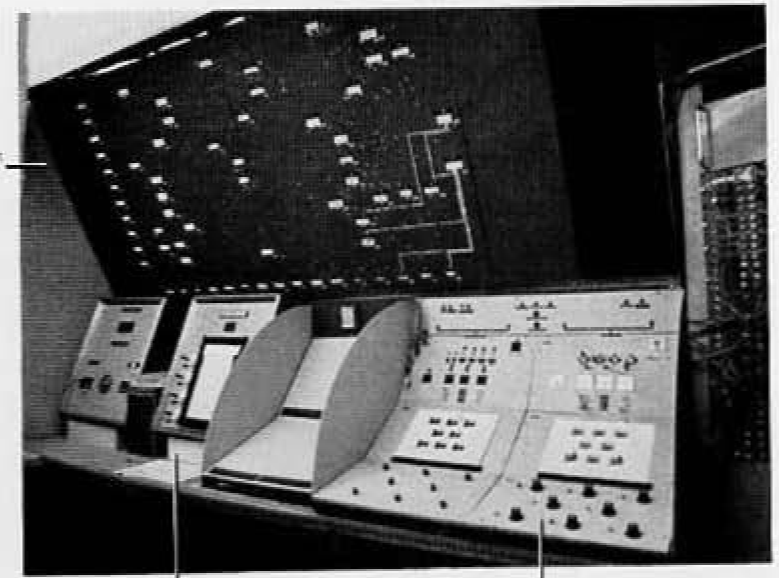
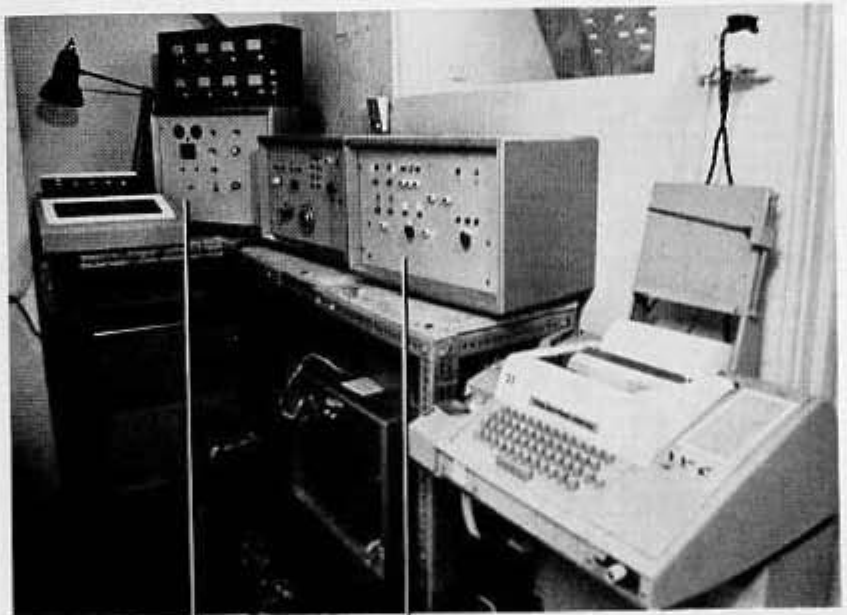


Plate 11 CASTE (Course Assembly System and Tutorial Environment): student system.

Plate 12 CASTE: operator's station with window overlooking student station.



task is simply a terse way of expressing a behavioural regularity (the notion that it is so forms the major tenet of behaviourism and underpins the experimental method). Or we might question the idea that *goal directed adaptation* is due to some kind of reinforcement (psychological term) or directed parametric modification (cybernetic term); once again, some such notion underpins the method. Perhaps the least sophisticated comment to make is that human beings, as well as rats and monkeys, are prone to all manner of internal and unobservable changes that alter their behaviour. The reality of coupling these organisms to an environment in such a way that stimuli can be *legitimately* regarded as input states, is dependent upon providing a certain kind of environment and task which places the organism in an *operating region* where the task is neither overloading nor underloading.

The underloading caveat is due to one of the inbuilt propensities for change noted a moment ago; it seems that an organism, and a human being in particular, is built to learn and, as a result, must be given something to learn about if he is to remain *genuinely* coupled to the environment (i.e. if the organism's input states are to be identifiable and in correspondence with the environment's output states and vice versa). If the underload limit is contravened then the human being attends to something else (which is unobservable) even though he may still devote *part* of his attention to the routine task. Conversely, a task that is too difficult is rejected as unintelligible. True, the human being may brood on the matter as a puzzle, but *that* is not observable in the stipulated framework. If either the underload or the overload conditions are contravened, the desired input state/output state/ relationship is disrupted. In either case the crucial closure condition which substantiates the coupling of the organism/environment subsystems as *the* system under observation is lost. Moreover, this must be so if man is made to learn, and built, also, with limitations upon capacity, storage and the like.

The *operating region* is not fixed. What does or does not 'load' the human being depends upon his experience and the extent of his adaptation to the environment. Hence, the classical idea of obtaining constant experimental conditions by setting a constant task in a carefully replicated environment from which irrelevant disturbances are excluded, is fated to become an inappropriate paradigm for experiments (or practical test and performance situations) that are at all prolonged. Either the task will overload the subject or it will come to underload the subject; in either case, the whole basis for identification is undermined. The objection, it should be stressed, is inapplicable to many *one shot* experiments (for example, in perceptual psychology) and it is pedantic for short experiments or where problem solving of a particular type is *meant* to be private (for example, experiments on the difficulty human beings experience over complementation and in extracting data from negating statements of 'what is not the case'). In its present form the objection applies only to lengthy experiments. With human subjects though, if a more fundamental point of view is adopted, the *principle* is universal.

A more appropriate paradigm for securing constant conditions would be an arrangement whereby the subject (in the examples, this is a human being) can be kept working within his individually determined and individualistically altered *operating region*.

The experimenter could, in principle, take the necessary precautions himself, i.e. he might adjust the rate and form of stimulation so that a response criterion, indicative of continuing task performance, is satisfied; for example, by making life more difficult if an index of correct performance increases in value and making life less difficult if the correct performance index decreases in value. To instrument this plan the observer needs a task and an environment which furnishes more than enough variety to overtax the subject at any point in the experiment.

Two problems result from allowing the experimenter to act in this way. First there is a practical administration problem. All the measuring and adjusting is likely to overtax the participant observer so that (even if he circumscribes a compartment of his mind to act as a *regulator* and leaves the rest of it to *observe*) he is likely to find himself so stressed that no energies are left for his main occupation, which is observing. Secondly, there is an experimental design problem due to the fact that the observer is no longer *external* but rather *participant* at least some of the time. It is later shown that the regulatory activities needed to secure the desired consistency 'subject working in operating region' entail other than causal transactions between the subject and the 'regulator', and thus render any 'regulator' a participant most of the time.

The proposed method is as follows. Instead of the classical organism/environment system in Fig. 57, construct the system of Fig. 58 in which the controller is a mechanical or computer programmed device for executing a steady state or null point operation. If ρ is a performance index and if a level of task difficulty is represented by η then the controller in question determines the value of ρ, compares it with a null point value ξ (indicative of satisfactory and correct task performance) and operates upon η, as follows, for experimental trials (or time instants) indexed n.

$$\eta(0) = 0$$
$$\eta(n) = \Sigma\Delta\eta(n)$$
$$\Delta\eta(n) = \begin{cases} +1 \text{ if, } \rho > \xi \text{ unless } \eta(n) = \eta_{max} \text{ when } \Delta\eta(n) = 0 \\ 0 \text{ if, } \rho = \xi \\ -1 \text{ if, } \xi > 0 \text{ unless } \eta(n) = 0 \text{ when } \Delta\eta(n) = 0 \end{cases}$$

As a convenience, it is usual to assume that ρ and η are normalised so that

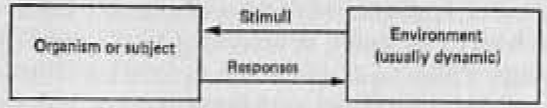


Figure 57 The classical organism/environment system.

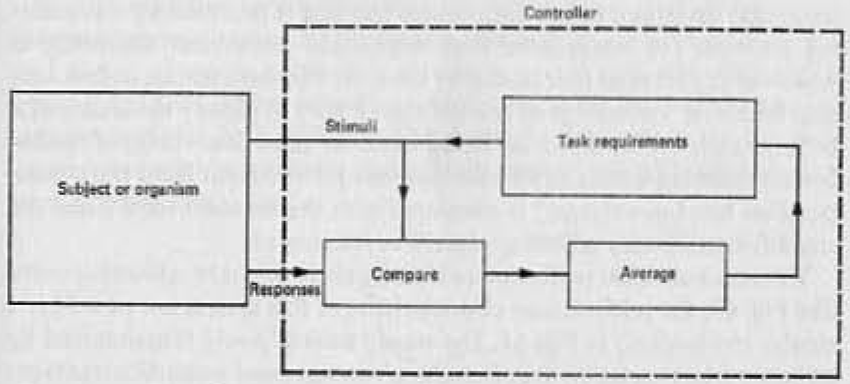


Figure 58 Alternative system to that of Figure 57.

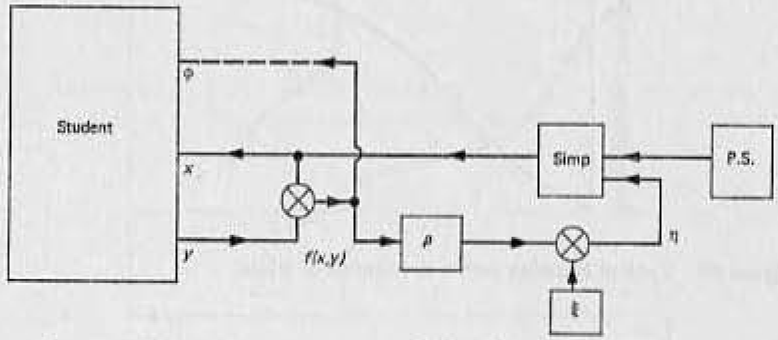


Figure 59 x, stimulus display; y, response; $f(x,y)$, 'correct' comparator output; P.S. problem source; Simp, problem simplification or difficulty variation; ϕ, knowledge of results, if given.

$1 > \rho > 0$ and $1 > \eta > 0$. Since task difficulty is modified, in effect, by *simplifying* task situations that are inherently too difficult, it is useful to introduce a degree of simplification index $\mu = 1 - \eta$.

The system (Fig. 59) operates in the context of a source of task requirements (PS or problem source since task requirements will later be identified with problems). The subject A receives problems that have been simplified to a variable degree, μ, by the controller, B, which may thus be regarded as cooperating to a variable extent in task performance. Differently phrased, B adjusts the level of difficulty, η, according to the stipulated algorithm. For this purpose, B must be designed to determine the value of ρ, typically, as a correct response rate or a sequential correct response function (so many correct responses in a row) or a Waldian- or Bellman-type *estimate* based on correct-response sequences or, for a quasi continuous task, an index of unwanted deviation. Hence the ρ computing box is preceded by a stimulus (x) response (y) comparator that determines correctness according to whatever x, y relation is ordained by the task. This comparator output may also furnish a 'knowledge of results' signal. (Or, to parody the stance of a behaviourist, a 'causative reinforcing stimulus' since 'knowledge of results' has no meaning within strict behaviourism.) The output from the ρ computation box (an averager) is compared with the constant term ξ and the simplifying/difficulty adjusting algorithm is executed.

Whereas a classical performance characteristic would be a learning curve like Fig. 60, the performance characteristics of this system are stated (as a similar idealisation) in Fig. 61. The steady state of $\rho \approx \xi$ is maintained by an appropriate variation in η. Certain advantages, and some disadvantages as well, stem from this fact.

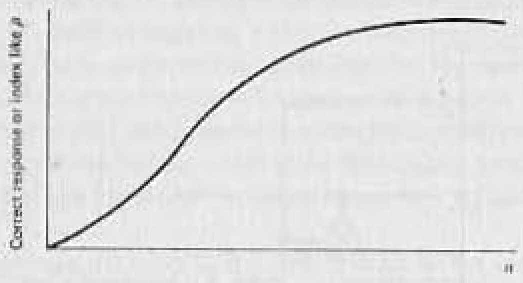


Figure 60 Typical learning curve. n, number of trials.

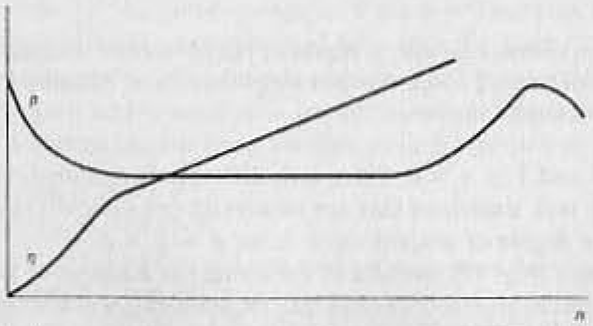


Figure 61 Performance characteristics of a steady-state system. n, number of trials.

1. The advantages are (a) that the subject works in his operating region where he is *able* to attend to the task which the experimenter deems relevant, and (b) that learning effects in the subsystem A are compensated by the action of B so that the *joint* system under observation is not a *learning* system or an adaptive system but a *control* system pure and simple, the interaction within which is directly observable.
2. The disadvantages are (a) that the *joint* system is relativistic (in a rather degenerate sense; the observable interactions are of A relative to B and B is a surrogate participant observer), and (b) the advantages only apply if the joint system is *stable*, which it might not be. The question of stability is obviously crucial. It will be shown, by citing examples, that such arrangements are empirically stable, for a certain to-be-specified class of environments and tasks.

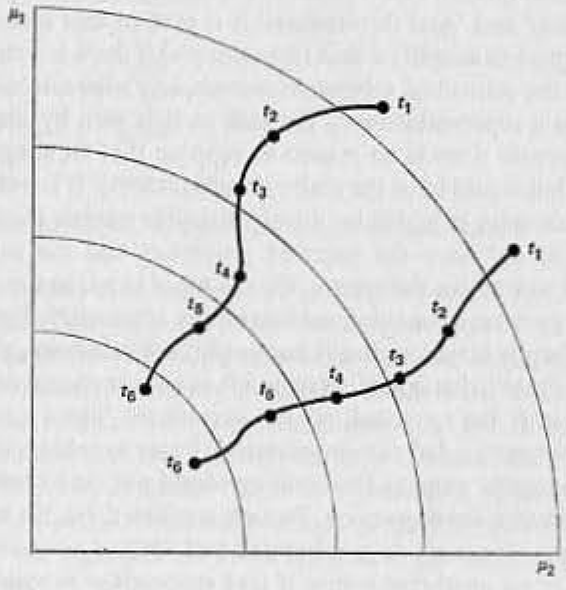


Figure 62 Trajectories for systems that are Lyapunov stable.

The most appropriate formal canon is Lyapunov stability. In the weak sense this condition has the following interpretation. Consider a space with $2m$ dimensions representing the ρ_i and the μ_i $(i = 1, \ldots, m;\ \eta_i = 1 - \mu_i)$ with the origin of these coordinates at zero. Since the derivatives of the ρ_i vanish $(\rho_i = \xi)$ concentrate on the m dimensional subspace of the μ_i (Fig. 62) on which Lyapunov functions are inscribed as closed surfaces surrounding the origin and nested beneath one another. Represent the system state trajectory (points $\vec{\mu}(n)$ on the surface) as a cyclogram. The system is Lyapunov-stable if the trajectory eventually approaches the origin and if it never crosses from the inside to the outside of a boundary.

In saying that systems are empirically stable, I mean that their irregularities satisfy this condition, on average, and may be constructed to do so, in all particulars, by the choice of appropriate functions. The proposed method can be elaborated a great deal. The algorithm is extremely primitive, for example, and is easily improved though (as a matter of fact) *most* improvements have little practical value. The controller described above is unidimensional. This restriction is inessential. The majority of steady-state controllers are actually multidimensional devices so that $\rho = \langle \rho_1, \ldots, \rho_m \rangle$ and $\eta = \langle \eta_1, \ldots, \eta_m \rangle$. The performance criterion is very rigid. That constraint is lifted in at least one of the examples to be discussed.

The crucial and outstanding issues are to do with some half concealed underlying assumptions. Apart from the previously voiced disquiet over the status of 'goals' and 'goal directedness' it is evident that a controller can only be designed to simplify a task (for example) if there is a model for the task and for the controlled subject. Moreover, any adequate model for the task must be a representation of the task as it is seen by the controlled subject (otherwise there is no reason to suppose that an allegedly simplifying operation would have the desired consequences). It is possible to cite numerous examples in which intuitively plausible models incorporated in controllers do *not* have the required properties and the controllers in question fail to stabilise the system. On the other hand, in the examples to be cited the necessary conditions are met quite adequately. Perhaps this is a matter of happy intuition. But it can be phrased otherwise, as follows. If the subject is held in his operating region then he is prepared and he is able to act as though the (physical) stimuli constituted 'input states' and the (physical) responses acted as 'output states'; hence to subserve the external observer's scientific dogma. He could or would not do so unless he were maintained in an operating region. To maintain him there, the experimenter starts with the observer's dogma in mind and designs a controller to hold the subject in an operating region *if* that supposition is valid. In fact, it *becomes* valid due to the control operation only, i.e. by a boot-strapping process. For the instances to be cited, the boot-strapping works.

1 Error Score Constancy in a Manual Task

The subject is presented with a compensatory display, usually on a cathode ray oscilloscope. In the one-dimensional case shown in Fig. 63 he perceives the locus of a point on a line and is required to adjust a manual control in order to keep the point within a line segment around a fixed position. His manual control determines the acceleration of an idealised vehicle (the

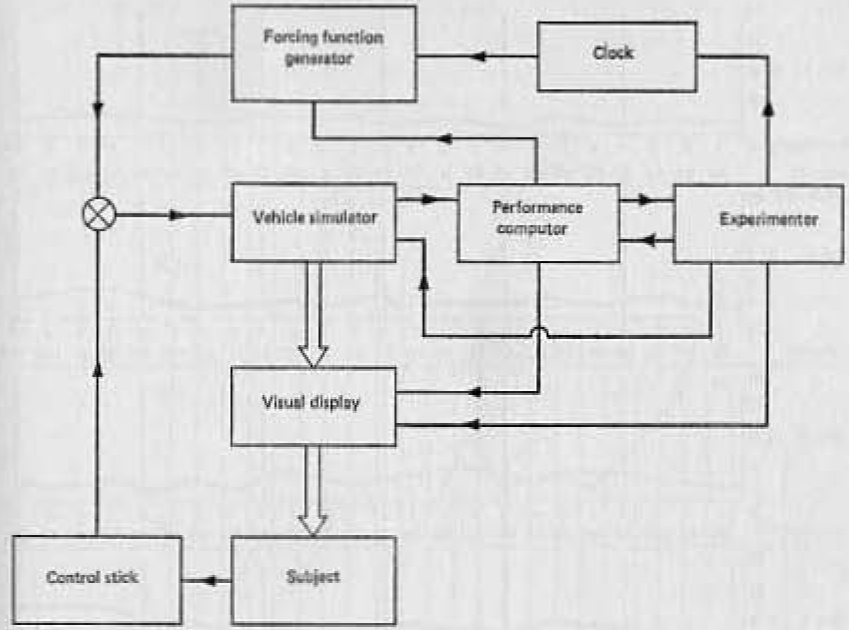


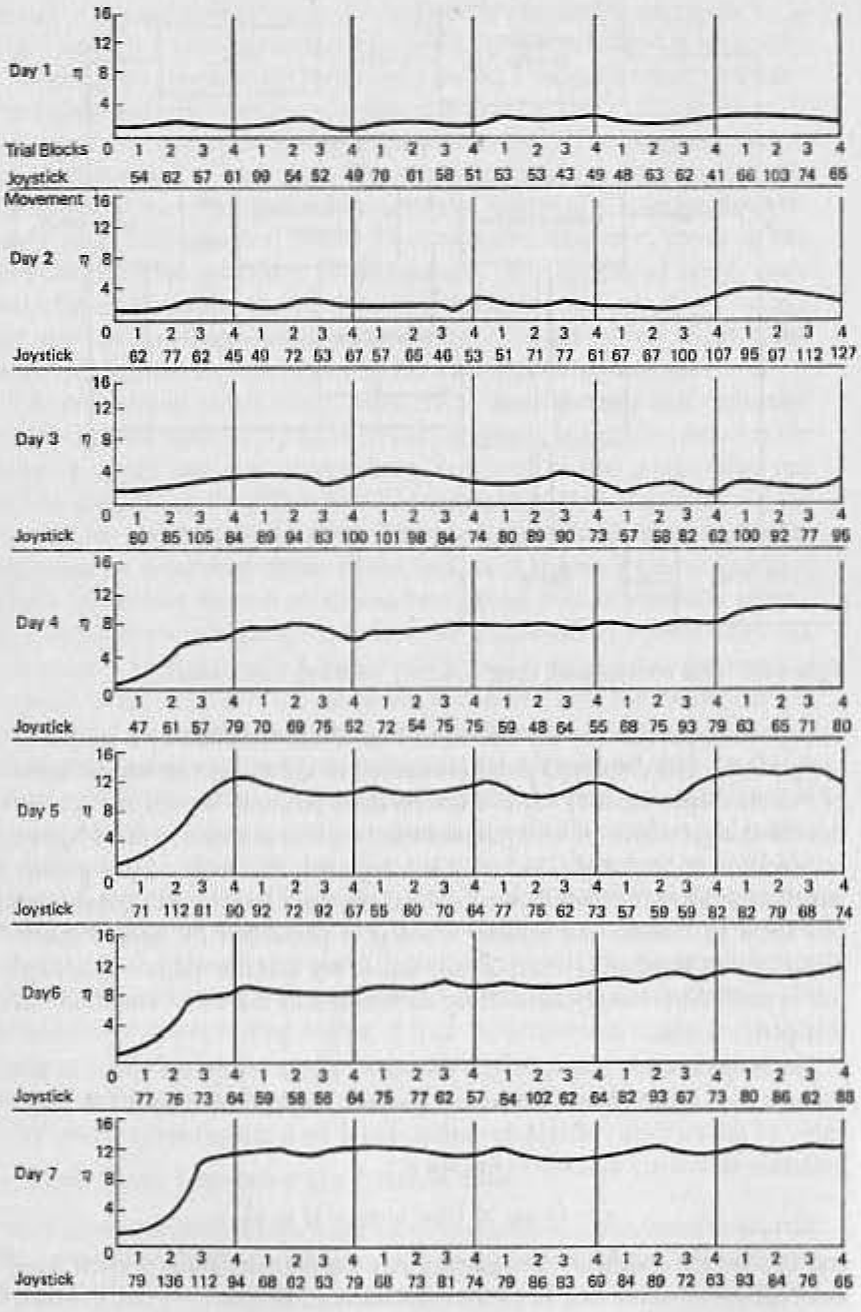
Figure 63 One dimensional, compensatory tracking, task system.

vehicle characteristics of the system in Fig. 63 are specified by a couple of integrators). The displayed point (described to the subject as 'an indicator of vehicle displacement') varies from its fixed position, even if the subject does nothing, because of an input perturbation that is added to the subject's acceleration control signal before it is integrated. The form of this perturbation is unlearnable and hence its value is unpredictable. But the subject can learn to 'handle the vehicle' when it is perturbed by unpredictable disturbances. Further, it is empirically safe to say that the 'vehicle-handling' job is made increasingly difficult by an increase in the mean amplitude of this perturbation.

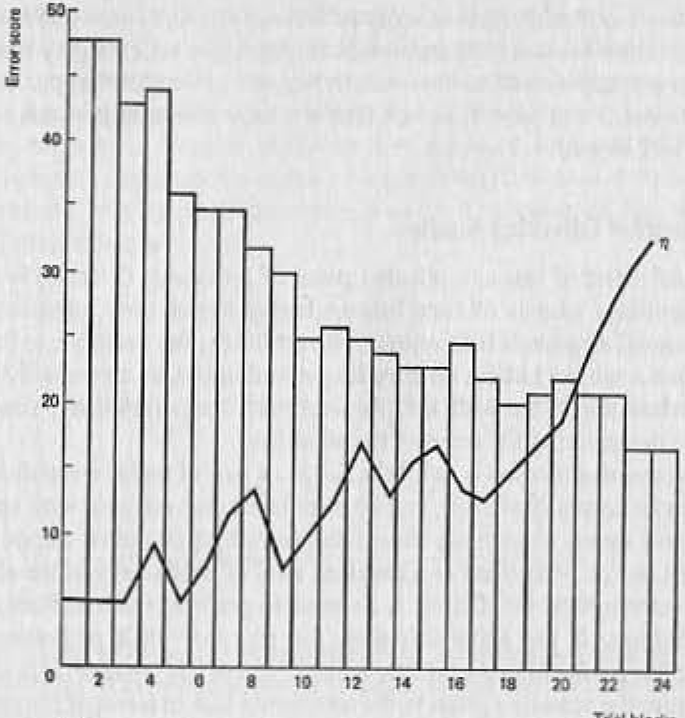
The subjects error score, γ, (the converse of ρ) is computed either as the average root-mean-square deviation from the fixed point or as the average value of the modulus of this deviation. Let ξ be a critical error score. To maintain constancy of error score, we set

$$\eta = \text{Const} \times \text{Time Integral} (\xi - \gamma)$$

and choose the (positive) constant so that the man–machine system does not become unstable due to over-compensation. In other words, the task difficulty is continually modulated in order to maintain the chosen form of constancy, that $\xi - \gamma \approx 0$ (or, phrased in a slightly different way, the



(a)



(b)

Figure 64 Typical data for a compensatory-tracking task: (*a*) using a joystick control (Pask *et al.*, 1967); (*b*) using push-button controls (Pask *et al.*, 1969), with trial blocks administered four per day.

environment is altered in order to maintain a constant relative or subjective difficulty, that is, a 'difficulty seen by a subject' and as indicated by his performance).

Typical response characteristics are shown in Fig. 64 and reveal stabilisation overlaid by a long-term adaptation effect. Systems of this type and their multidimensional analogues have been investigated by Hudson (1962) and Kelly and Prosin (1968) in the United States and, in Britain, by Gaines (1968; 1969b) and myself (Pask *et al.*,1967; Pask *et al.*, 1969b). One early application, to pursuit radar, is reported in Pask (1957). Only, in this case, the control operation consisted in manoeuvring a vehicle in two spatial dimensions and the manoeuvres were required to satisfy additional conditions regarding a safe attack method; a primitive version of this equipment was demonstrated and gave rise to a good deal of amusement at the British Psychological Society conference in Manchester in 1956.

Plate 3 shows a multidimensional equipment typical of later work – only two of the four double-beam display tubes were normally used but each tube can accommodate one bidimensional compensatory task (display spot controlled vertically as well as horizontally) or one bidimensional pursuit task (like the radar equipment, except that the tube diameter is much smaller than a PPI screen).

2 Attention Directing Studies

The chief merit of this complicated piece of apparatus (Plate 3) is that it accommodates studies of how human beings direct their attention. The conventional approach is to obtain a direct index; for example, to find out how often a subject notices an intruding stimulus and, as an embellishment, to correlate the index with the physical magnitude (intensity, size) of a variable determining the form of the stimulus.

An alternative method is to set the subject a pair of tasks; a standard task to which he normally attends, and an interfering task on a par with appreciating (and doing something about) the intruding stimulus. Suppose the standard task is performed at a constant level of proficiency in the absence of any interference, i.e. if there is no need to perform the interfering task. By definition, if the skills interfere, the standard task performance is impaired by performing the interfering task, and so it is possible, in principle, to measure the attention given to the interfering task in terms of impairment of standard task performance.

The trouble is that the constant levels of proficiency are rarely achieved; an even more striking difficulty is that interference overloads the subject so that he is displaced from his *operating region*; a level of loading at which it is possible to perform the standard task. Conversely it may happen, for lengthy performances, that tedium sets in and the subject no longer attends to the standard task (once again, he is working outside his operating region but on the opposite or underload side of it).

A scrutiny of Fig. 64 will show that steady-state conditions are maintained by virtue of the controller's compensating action which keeps the subject balanced at a null-point that (for an appropriate choice of ξ) *is* within his operating region. Moreover, the controller is able to compensate for any disturbance within the compass of the task determined environment, whether it is due to fatigue/adaptation or due to external perturbations that shift the subject's focus of attention. Moreover, within quite wide limits, the controller ensures that the subject works within his operating region for the main task (tracking or the like). Under these circumstances it is possible to have confidence in a differential measure of the attention which is given to an interfering task.

Specifically this main task was used, in various experiments, as the

standard task. Other display tubes in Plate 3 were used to display intruding stimuli (meeting with no response) or, in a different study, an input signal for a simple interfering task that the subject was periodically required to perform. Instead of measuring the influence of different intruding stimuli/interfering tasks by a decrease in the level of standard task performance (γ,ρ) the index of impairment was the compensation ($\Delta\eta$) needed to maintain a constant level of performance ($\gamma,\rho \approx \xi$). The curve in Fig. 65 is typical of most subjects.

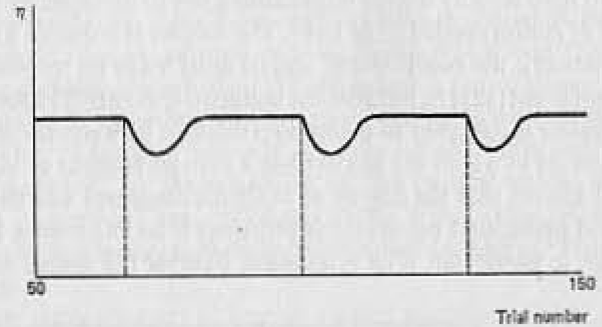


Figure 65 Form of curve for typical subject in compensatory-tracking task showing the effect of intruding/interfering stimuli. Dotted lines indicate onset of interfering stimulus.

The steady-state or null-point method of measurement leads to far more constant results than the conventional techniques; a point made independently and in different contexts by Gaines, Kelly, Seigman, and Sime. In connection with attention directing, it was possible to compare two kinds of measure: (a) a conventional differential measure; increase in r.m.s error (decrease in performance) for an *uncompensated* tracking task, and (b) a steady-state differential measure; decrease in η (to maintain constant r.m.s error on the standard task). The interfering stimuli were the same (moving visual) presentation in each case. Unit scores were taken as the area beneath the deviation in main task performance curve; the unit scores were averaged, for each subject separately, over repetitions of the same kind of interfering stimulus to obtain two mean scores calculated from (a) and (b) for each subject, indicating the attention he had given to each kind of interfering stimulus. For a given subject there is no difference between the ranking of 'attention given' by criterion (a) or (b), i.e. a particular individual consistently treated the interfering stimuli as more or less salient regardless of the method employed. But the mean score variance is very much greater for the (a) scores than it is for the (b) scores and this difference is exhibited consistently by all subjects.

3 Systems for Maintaining a Constant Level of Vigilance

In Fig. 66 the subject looks out for an important or relevant event in a background of irrelevant disturbing events. In my laboratory these events have been specific subpatterns (X_i) in a visually presented series of patterns. A correct response mode is predefined for each subpattern and the patterns may be ranked by importance, as presenting, for example, various levels of hazard. This refinement is not considered in the sequel where stimuli (x) are taken to *have* equally urgent implications but to differ either in form or in the type of action needed or in both. The subject is required to make the correct response if the event occurs, and he must make no response if there is no relevant event; that is, he must not hallucinate events. The arrangement described below is designed to maintain a relation between the subject and his environment in which the subject has a constant degree of vigilance.

It is well known that the degree of vigilance decreases and the subject's selective and perceptual behaviour is impaired if he has been a long while at a job and, in particular, if he is fatigued. Further the degree of vigilance

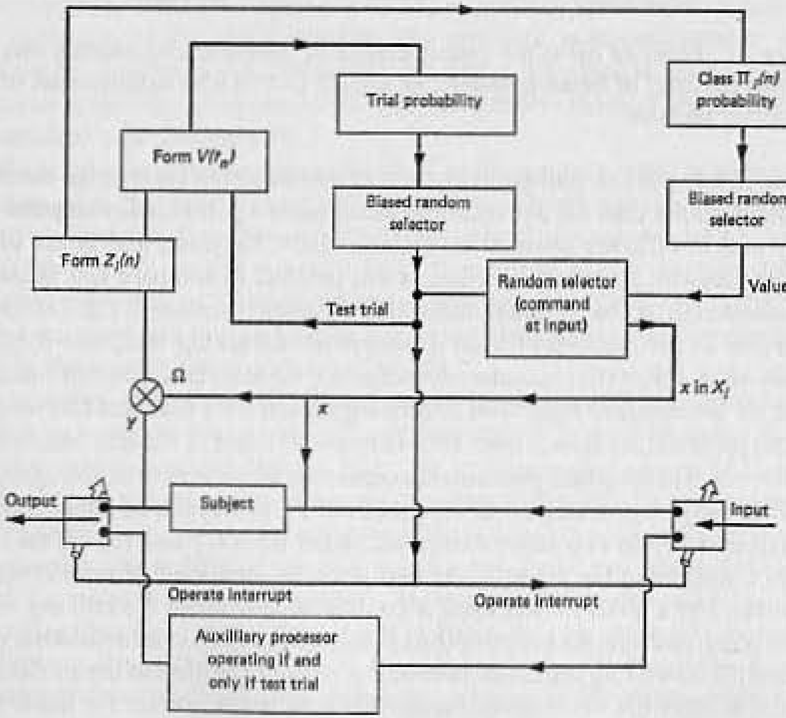


Figure 66 Vigilance system.

varies differentially according to the class of relevant event (a subject may be competent to deal with some sorts of events, but incapable of dealing with others).

In Fig. 66 there are m classes of relevant events (X_i; $i = 1,2, \ldots, m$) each to be dealt with according to a rule $\mathscr{R}$. In other words, the desired response y, given event x, is $y = \mathscr{R}(x)$ where y is selected from a set Y and where x is selected from a subset X_i of X (the set of all relevant events including the null event).

A basic design assumption underlying this type of system is that off-loading is possible, i.e. there is an auxiliary data processor in parallel with the subject (it may be another human being) which can take over the subject's job. Thus, unknown to the subject, it is occasionally possible to delegate his job to the other data processor and to do without the subject's services. The off-loading intervals must be spaced fairly uniformally in time. They are used for the processing of 'test signals' which, so far as the subject is concerned, are indistinguishable from relevant events. His behaviour with reference to them and to reality is assumed to be and actually *is* identical.

Constant vigilance control is most readily described under an artificial supposition that the control mechanism is able to inject a test signal at *any* instant, although it aims to minimise the amount of time spent in this fashion. This assumption is relaxed or even renounced in practical applications of the system; for example, the control of inspection and sampling vigilance.

The control mechanism first computes the conditional frequency estimate.

$$z_i(n) = \frac{\text{Number of test trials between } n \text{ and } (n-k) \text{ for which } x \in X_i \text{ and } y = \mathscr{R}(x)}{\text{Number of test trials between } n \text{ and } (n-k) \text{ for which } x \in X_i}$$

so *that* as t increases, $z_i(n)$ approaches the conditional correct response probability $p_i(n) = p(y = \mathscr{R}(x)/x)$, $x \in X_i$, for $t = t_n$.

The vigilance at the nth trial (or at the instant t_n) is estimated by

$$V(n) = V(t_n) = 1/m \sum_i z_i(n)$$

on the assumption that the probability, $\pi_i(n)$ that $x \in X_i$ at the nth test trial approaches $1/m$. This assumption is rendered plausible due to the activity of the control mechanism as it is described below.

The first part of the control operation determines the probability $R(t)$ that some test trial will be made in the interval between t and $(t + dt)$ where dt is short. (If a test trial is made in this interval and if it is the nth test trial then t_n is between t and $t + dt$.) The present controller is designed so that

1. $$R(t_n) \simeq 1 - V(t_n - 1)$$

The next part of the control operation selects the class of events from which one event will be 'randomly' sampled at $t = t_n$, if some test trial is made on this occasion. The selection probability for the class X_i is denoted π_i; and the controller of Fig. 66 is designed so that

2. $$\pi_i(n) \simeq 1 - z_i(n)$$

Finally each of the terms $R(t_n)$ and $\pi_i(n)$ is interpreted as a bias applied to a probabilistic selection device (a variable 'window width' random address generator). The effect of equation (1) is to increase the test trial rate as the subject's vigilance decreases, and the effect of equation (2) is to *rehearse*, most frequently, the most neglected events. The latter process maintains the average values of the $\pi_i(n)$ close to $1/m$ as required in the definition of vigilance.

As a theoretical aside, it is worth commenting that the recursive definition embedded in the design is typical of relativistic systems. Here, vigilance is defined on the assumption that the $\pi_i(n)$ are nearly equal. There is a *rule* (namely a psychological principle of familiarisation due to repetition) which, if valid, justifies the last control operation as an expedient for *securing* equality of the $\pi_i(n)$. Since the *rule* is psychological, the entire operation is a *heuristic* recursion.

4 Maintaining a Constant Proficiency in a Coordinate Transformation Skill

The subject is presented with the display and response board shown in Fig. 67. Each trial, Δt apart, he is presented with a figure in the alphabetic display and required to select the row and column buttons that designate the row and column coordinates of the figure in the 3 × 4 rectangular display before δt seconds after the stimulus. With δt set at 2·5 seconds, Δt at 5 seconds, and the between-trial interval at 1·5 seconds this is a difficult job, and the novice is unable to do it unless he is provided with cueing information.

The skill is characterised by a couple of error factors (in the sense of Harlow, 1959), namely, a row and column error factor, since the subject's response may be row-correct and column-correct independently. Hence, the cueing information is delivered with reference to the row-response selection and the column-response selection separately. The information concerned is provided by the row lamps and the column lamps of Fig. 67 which are illuminated at a variable interval (within the allowed interval of δt seconds) after the appearance of a stimulus figure in the alphabetic display.

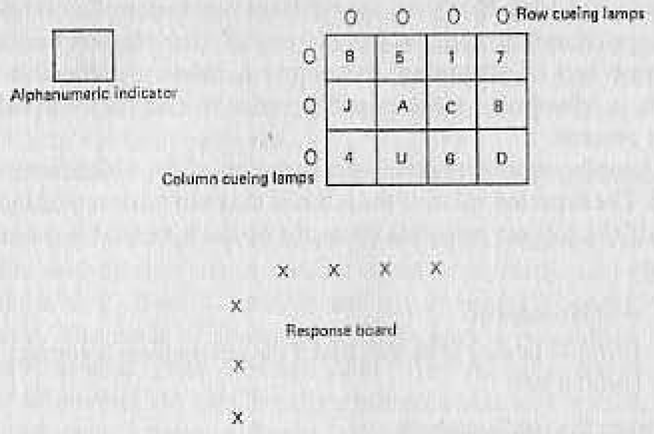


Figure 67 Coordinate transformation task showing display and response facilities.

Let i = 'column' or 'row', and let ρ_i be a proficiency index

$$\rho_i = \text{Average value } (\phi_i)$$

where

$$\phi_i = \begin{cases} +1, \text{ if the } i\text{th type response selection is correct and presented before the } i\text{th type cueing information;} \\ 0, \text{ if the } i\text{th type selection is correct but too late;} \\ -1, \text{ if the } i\text{th type response selection is mistaken or absent} \end{cases}$$

The value of +1 in this rule may, with advantage, be replaced by the term Δt − latency. Define η_i as the proportion of Δt that elapses before the ith type of cueing information is delivered to the subject. The maximum value of η_i is Δt and if $\eta_i = \eta_{max}$ then no cueing information of the ith type is delivered.

The control mechanism for variable delay cueing satisfies

$$\eta_i = \text{Constant} \times \text{Time Integral } (\rho_i - \xi)dt$$

where ξ is the required level of proficiency and the constant is a positive rate term. The system is started in the initial condition $\eta_i(0) = 0$. The value of η_i is defined unless or until $\eta_i = \eta_{max}$.

4.1 Clamping Techniques The assumption underlying simplification by a variable delay cueing procedure, with Δt constant, is that the subject operates decisively (to select responses) at the same rate throughout the experiment. Given this, the cueing information can be delivered at an optimal (constancy-maintaining) instant during the decision process taking place at a given trial. However, the constant rate assumption is (at most) a crude approximation and the serious use of variable-delay cueing rests upon some sort of 'clamping' technique; a technique whereby the trial duration is 'clamped' to the possibly variable interval occupied by the decision process.

The clamping technique involves a variation of Δt, which is now written as $\Delta t(n)$. The expected value of the interval that will be occupied by decision making if the subject responds correctly at the n_0th trial is estimated for $n_0 > n$ as

$$t^*(n_0) = \text{mean value of} \begin{Bmatrix} \Delta t(n) - \text{latency at the } n\text{th trial if the } n\text{th response is correct} \\ \Delta t(n) \text{ if not} \end{Bmatrix}$$

To complete the feedback loop

$$\Delta t(n_0) = t^*(n_0) + \gamma$$

where γ is positive (thus increasing the pace as proficiency improves).

Since η_t is defined proportionally, the cueing delay becomes a proportion of $\Delta t(n)$ rather than a proportion of Δt and the entire process is variably paced.

4.2 Results Some response characteristics are shown in Fig. 68. There is overall adaptation as the subject comes to grips with the skill but, before that, the interaction between the row and column components of the skill is manifest as a pronounced, but duly compensated, interference effect.

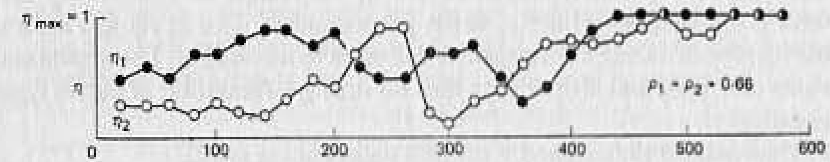


Figure 68 Coordinate transformation task. Response characteristics for row difficulty level (η_1) and column difficulty level (η_2). There is appreciable interference before adaptation.

The clamping operation keeps the subject working within his operating limits with respect to both components of the skill. Under these circumstances it is possible to perform an 'internal' study of attention directing. Either component may be regarded as the *standard* task of the previous example and the other component as the interfering task. Only here there is no initial polarity or task dominance and both tasks (being, in fact, part of the same skill) are performed continually. It is still possible to obtain a

profile of the microstructure of adaptation and to determine how this profile varies from subject to subject, or under the influence of perceptually discriminable changes (making the row lamps brighter than the column lamps), or value assignments telling the subject that a row response is more important than a column response.

All of these measurements have practical consequence. In summary they are as follows.

(a) It has been found that, although all subjects experience some interference, the method they use to combat its adverse effects, as well as the rate at which they do so, is variable and a very useful individual ability discriminant. The mode of combatting interference is generalised for one subject over several quite different skills. For example, the following categories of method are significantly distinguishable and reliable as test indices (predictions between different test tasks, for each individual at the 0·1 per cent level of significance and score differences between groups of individuals using different methods also at the 0·1 per cent level; thirty-five subjects).

Method 1: 'One at once'; an oscillatory behaviour with long cycles (like the data in Fig. 68).

Method 2: 'Assemble subskills'; (the η_1 curve and the η_2 curve increase, more slowly together. At points of difficulty there are very short but high amplitude cyclic fluctuations).

Method 0: 'Bash on regardless' (like the 'assembler method' except that the subject in question apparently has no assembly rule. Adaptation is *very* prolonged; even for this skill in the order of 1500 trials rather than the 500 to 750 trials of Method 1 or Method 2).

(b) The effect of value assignment instructions is in the expected sense (a subject attends more to the positively valued component) but the notice taken of these instructions differs from subject to subject though (between instructions) it is consistent for any one individual. It is tempting to advance this measure as an index of susceptibility to authority, and I believe it is legitimate to employ it as an index of the extent to which individuals are prepared to accept arbitrary value edicts.

No serious work has been done in my own laboratory upon the influence of perceptual variables such as intensity of illumination. Kelley, on the other hand, has done a great deal and reports on the current state of the art in his compendious manual (Kelley and Kelley, 1970).

4.3 Different Skills It is noteworthy that very similar results are obtained for tasks in which the component subskills are rapidly alternated over blocks of trials rather than performed in the course of the same trial. Figure 69 is data culled from a code-learning study; the two subskills

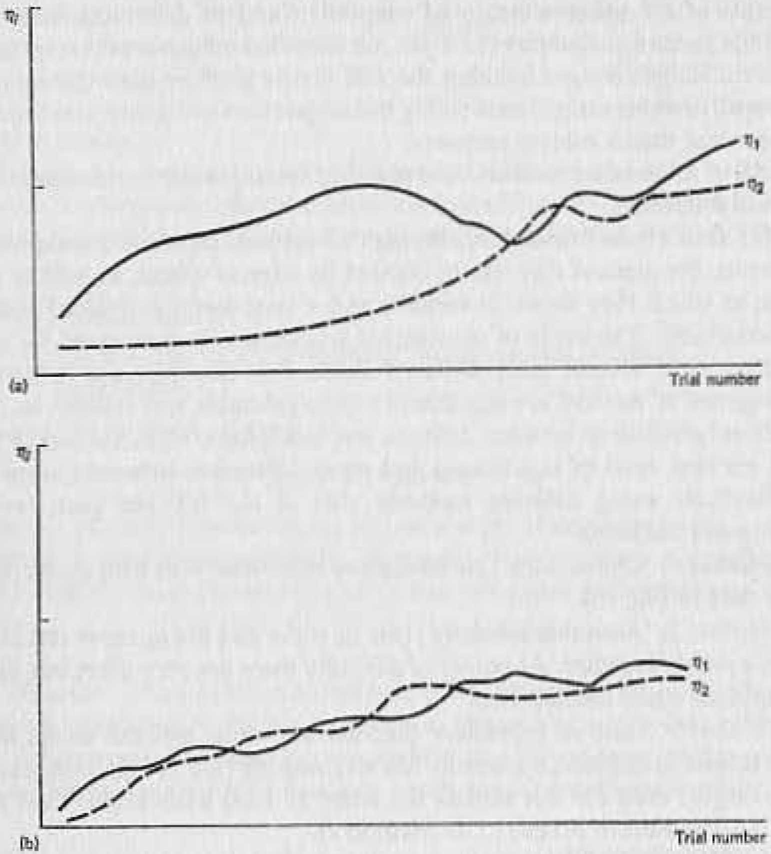


Figure 69 Typical data from code-learning experiments (Pask and Lewis, 1967): (*a*) method 1; (*b*) method 2.

corresponding to two distinct coding rules, applied alternately. This is one of the test tasks used in order to obtain the results in (a) and there is an obvious difference, even to the eye, between the *Method 1* subject and the *Method 2* subject. Here, the effect of clamping is achieved by a different technique; 'probabilistic alternation' which forces the subject to rehearse the subskill in which he is least proficient with greater frequency (however the rehearsal probability is not allowed to reach 1 or 0; and in the 'minimal condition' both skills are rehearsed with equal frequency).

4.4 Complex Forms The task employed can be quite complex and, if so, much more information can be garnered whilst it is performed. For example, we have carried out experiments (mental tests perhaps) using a

six property classification task for visually displayed ambiguous patterns. The subject is required to assign the value +, −, or *null* to each of six properties designating features such as 'closure' or 'curvature' of a displayed figure, and to do so for each one of a series of patterns which are presented on a cathode ray tube against a 'noise' background of constant intensity. To aid the subject, he is provided with cueing information in the sense that one feature (the 'upper curvature', for example) is brightened relative to the constant intensity 'noise' background as the pattern is inscribed on the screen. The display and response equipment is shown in Fig. 70. It receives an input from and provides an output to an adaptive device of the kind already described.

Clamping conditions are used, as in the coordinate transformation skill, so that $\Delta t(n)$ estimates the subjects expected decision time. Within this

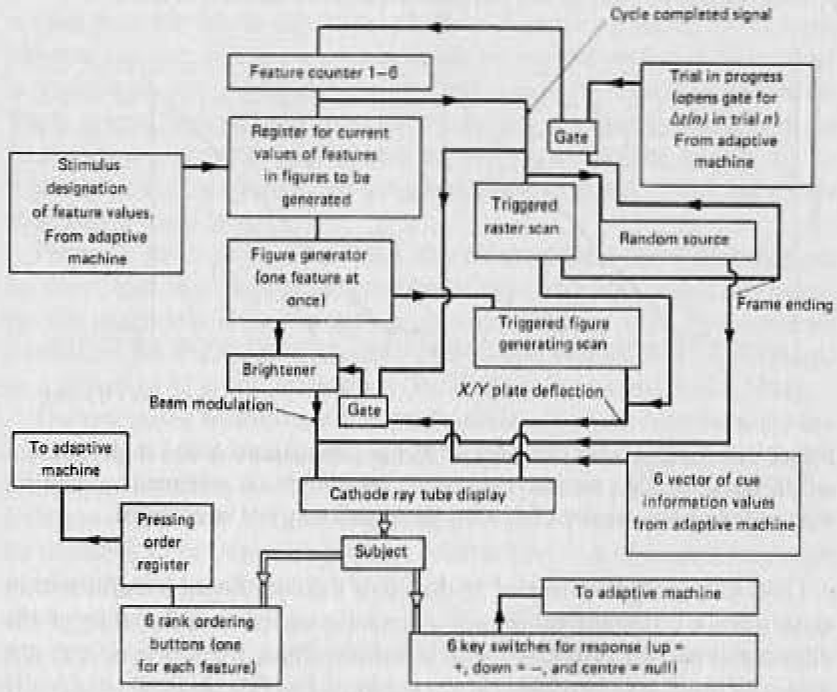


Figure 70 Ambiguous figures task. Figures with six relevant features appear against a random background and are classified by assigning actual values to each feature, via six key switches. For example, the feature *Upper curvature* has possible values: Right (+), Left (−) and none (null, a straight line). If cueing information is delivered in respect of a given feature, the 'writing' beam is 'brightened' whilst that feature is (repetitively) inscribed on all occasions in $t(n)$ subsequent to delivery; hence feature is made prominent against random background.

interval, cueing information is delivered sequentially and delayed with respect to the decision time coordinate. Giving cue information at trial n (with respect to a particular feature/property) means that the feature's inscription is brightened/intensified at the trial in question after and only after a 'cueing threshold' is exceeded. The cueing process is illustrated by Fig. 71 in which the main parameter is the area under the waveform which sequentially switches off the brightener in the cueing display (a response is correct in classifying the ith property if the classification is correct and the response denoting it occurs before the delivery of cueing information about the ith property). Specifically, it is assumed in Fig. 71, (1) that cueing information determining the values of properties a, b, c, d, e and f is removed in alphabetic order, with the f value removed first, and (2) μ is modulated to maintain $\rho = \xi$ where ρ is an average taken over the subject's performance ρ_i with respect to each of the properties in the classification task.

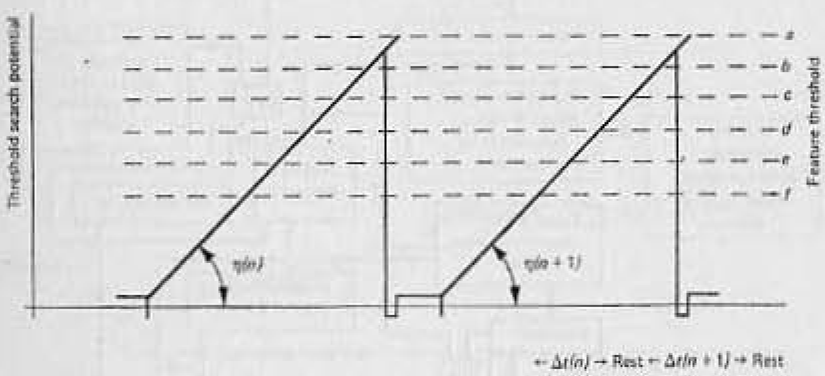


Figure 71 Arrangement for delay of cueing information. A cue is given if and only if threshold for a feature is exceeded. Thresholds are permuted by subject's rank ordering and, on any trial, some thresholds may not be exceeded.

Used as a multidimensional analogue of the coordinate transformation task, the six threshold values are adaptively varied as a function of the differential correct response to each feature. Thus, for $i = a, b, c, d, e, f$, values of proficiency variables ρ_i are computed as functions of specifically correct responses. Specific cueing thresholds $\mu_i = 1 - \eta_i$ (difficulty is increased by lowering the cueing threshold) are regulated individually to values such that $\rho_i = \rho +$ constant $(1 - \rho)$. In this case, the system yields data of the kind already described.

For a skill of this type, however, the η_i values may be given an interestingly different interpretation; as indices of psychological similarity or dissimilarity. For example, if η_i and $\eta_j (j \neq i;\ i,j = 1, \ldots, 6)$ have the

same value then it can be argued that feature i and feature j are psychologically equivalent. Conversely, if $\eta_i \neq \eta_j$, it can be argued that feature i at 'strength' (or more accurately 'with information') μ_i is psychologically equivalent to feature j at strength μ_j.

4.5 Other Types of Output Data In pursuit of this idea (and as an alternative method) suppose the arrangements for specific η_i (or μ_i) adjustment are deleted, but the subject is allowed to rank order the features or properties a, b, c, d, e, and f so that he receives cueing information sequentially and in his preferred order (rather than an arbitary, alphabetic order). For this purpose, he is provided with six response buttons which apply electrical charges to capacitors associated with each of the six property specific variables that determine the feature thresholds. The higher the charge on the capacitor, the higher the value of μ_i or, the lower η_i. The subject pays for this facility in terms of a commodity ('money in his bank balance'; in fact, charge on a main capacitor but the 'monetary metaphor' is useful and very intelligible) so that he cannot be utterly improvident. These μ_i charges leak away from the capacitors at a fixed rate and in order to maintain a property-ranking the subject must continue to pay for its reinstatement. The entire process is constrained so that the bank balance is kept nearly equal to η.

From the machine's point of view, the informational status is unchanged by the operation of imposing a rank ordering pattern although it is changed by the automatic adjustment of $\mu(\eta)$ to maintain $\rho = \xi$. This example introduces the technique of monetary regulation that has been widely used in the control of group interaction (Pask, 1964; Lewis and Pask, 1964).

The remaining examples of the steady-state techniques have been chosen to show that a performance index need not be *absolute* in the sense that it relies upon an external criterion of rectitude; either a rigid or a probabilistic criterion. All indices like ρ undoubtedly do so. As a technical trick they can be modified to rest upon disjunctive criteria (any one of several responses are correct, not just one response that is correct) but this expedient does not change the picture in an *essential way*. The self-consistency indices used in the sequel are, however, quite distinct from an absolute performance index though their values are used by the controller in much the same fashion as the values of ρ.

The examples also demonstrate that a steady-state system, though in one sense *convergent*, can be used to foster essentially divergent behaviour (invention of classifying attributes; innovation of fresh rules).

4.6 Classification Tasks To begin with, notice a common perceptual oddity of almost any human being. If he is free to classify objects the human being is prone to use familiar attributes as the basis for his classification. These are idiosyncratic, but tend to be banal and easily named rather than immediately perceived or mentally manipulated, e.g. geometrical attributes like rectangle, square, etc., or exact physical magnitudes. The experimenter can avoid this tendency, by telling the subject in advance that he *must* use particular classifying attributes. In visual classification for example, the subject may be told to classify each member of a sequence of tachistoscopically-exposed pattern stimuli according to attributes such as 'size', 'number of distinct parts', 'circularity', and so on; these names are assigned, in a mechanised experiment, to response buttons that are pressed after each tachistoscopic exposure. This expedient fixes the attribute names and thus prevents an experimenter finding out how the subject chooses new attributes.

But, to reiterate the original statement, if a subject is allowed to name the buttons as he desires (to select his own attributes) he nearly always reverts to familiar feature names and, as a result of this, his behaviour exhibits problem-solving according to familiar methods rather than problem-posing or the construction of novel attributes, tests or methods.

These comments about the freely choosing subjects are fairly accurate if the stimuli are unambiguously presented and if the subject is able to use his collection of descriptive attributes in a consistent and successful fashion. If the presentation is ambiguous, the subject may be forced to relinquish his original choice of attributes (because he cannot use them consistently) and to select new ones. Although re-orientation takes place when ambiguity is introduced (conveniently by reducing the tachistoscope exposure interval) there is a delicate balance between the tendency to search for new attributes and a tendency to give up altogether, in the belief that it is impossible to make sense of the environment.

In a free classification task the subject is required to select classifying attributes of his own choice in order to satisfy criteria that depend upon the attributes chosen; for example, to produce an informative and self consistent classification of the patterns or objects by employing the attributes he opted to specify.

Parenthetically, such tasks have an inherent interest because of their relation to Kelly's (1955) repertory grid technique for eliciting *constructs* (of which a freely chosen attribute is the limiting case). The *constructs*, here treated as attributes, are generated by the subject (Bannister and Mair, 1968) not given by the experimenter.

5 Maintaining a Constant Level of Ambiguity of Visual Patterns

Some years ago (Pask, 1964a,b) Lewis and I attempted to induce problem posing or attribute choice in a task of this type by regulating the level of stimulus ambiguity, using the equipment shown in Plate 4.

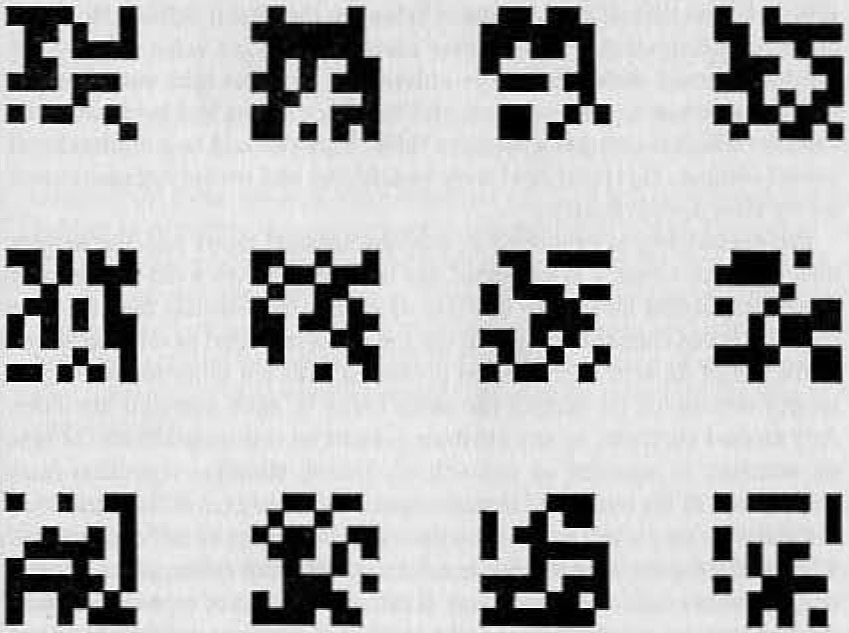


Figure 72 Examples of chequerboard patterns used in the attribute-selection task.

The stimuli were arranged in blocks or subsequences of about fifty items and were all of the same type: chequerboard patterns (Fig. 72). Knowing the type of stimulus and the criteria to be satisfied, the subject first selects the attributes that are to be used by naming response buttons that determine the value +, − or null he assigns to an attribute. The criteria are simply that any chosen classification is self-consistent and informative.

After dealing with a block of stimuli by classifying them, the subject is allowed to rename the classifying response buttons (to redefine his attributes). The experimental data consists in the list of the attribute names that are recorded after each block or sub-sequence of stimuli. Within a particular block of trials, the subject assigns names as labels to a maximum of eight buttons; he may use less than eight if he wishes, providing that his categorisation is informative and also self-consistent. Before the experiment was conducted, we described carefully what is and is not informative and self-consistent. If the subject had any difficulties, these were discussed.

5.1 An assignment of attribute names is *informative* if the subject can use the named attributes to divide the stimuli into coherent subsets. An attribute like 'being a chequerboard pattern' is not informative because the subject knows that all of the patterns belong to the class it defines. Nor is an attribute informative if it can never assume a positive value. Ideally, an attribute should dichotomise the universe of patterns into one class for which it assumes a positive value, and another disjoint and equinumerous class for which it assumes a negative value. There should be a minimal null valued residue. This ideal can rarely be achieved and we did not insist upon a very close approximation.

Self-consistency is explained in commonsensical terms and the subject does not know how it is *measured*. He is provided with a self-consistency score (the digital indicators in Plate 4) chiefly to convince him that the quantity is not chimerical. In fact, the subject is regarded as self-consistent in his use of an attribute if, when presented with the same *stimulus* upon *several* occasions, he assigns the same value to each non-null attribute. Any unused attribute, or any attribute present on one occasion and absent on another, is regarded as null-valued. Hence, stimulus repetition (and comparison of the initial and the subsequent values of each of the attributes) is the device employed to measure the subject's degree of self-consistency. The machine calculates the self-consistency value by reference to the last few attribute selection values which it stores in respect of repeated stimuli. At any rate on most occasions, the subject is unaware that an identical stimulus is repeated. From his point of view, self-consistency is a matter of using his own selection of attributes in a consistent fashion.

5.2 The calculations needed to obtain a consistency index are straightforward except for the notation involved. To simplify the description, let $y = y_1, \ldots, y_8$ represent the response elicited with respect to the pattern displayed at the nth trial $x(n)$ which is, in fact, a particular pattern x_i. Thus $x(n) = x_i$. Let $y^* = y^*_1, \ldots, y^*_8$ be the response to the last *presentation* of x_i (at some previous trial $n - k$; so that $x(n) = x(n - k) = x_i$). Each entry in y or $y^*(y_j; j = 1, \ldots, 8)$ has a value + or − or *null*, since *null* is assigned if an attribute button has not been labelled. Let $r_j = 1$ or 0 and let $r_j = 1$ if and only if $y_j = y^*_j$. A consistency score for the nth trial is obtained by comparing stored response vectors y, y^*, as

$$\begin{array}{c} y_1, \ldots, y_8 \\ y^*_1, \ldots, y^*_8 \\ \hline r_1, \ldots, r_8 \\ \hline \end{array}$$

and $r(n) = \sum_j r_j/8$. The value $r(n)$ of self-consistency is the sum of these terms, $r(n)$ over a block of trials, weighted by the value of $r(n)$ at the beginning of the current block of trials.

$$r(n) = \tfrac{1}{2}\Big(\sum_j r(n) + r^{\square}(n)\Big)$$

Block in
which n
is situated

where $r^{\square}(n)$ is the value of $r(n)$ for the last block. It will be recalled that the subject is permitted or encouraged to relabel his attribute buttons at the end of each block, and this computation is based upon the idea that he does so. A more provident estimate is obtained by carrying out a decremental summation over many blocks (the experimenter enters the names of attribute buttons that *have* been relabelled at the beginning of each block, so that the machine can count these attributes as null valued for the first of its calculations).

5.3 The 'informativeness' measure was used mainly to ensure non-triviality. For this purpose, it is sufficient to ensure that any labelled attribute is assigned at least one '+' value and (for a different pattern stimulus) at least one '−' value in each block. A check upon this condition is easily introduced by a circuit which detects, for any labelled attribute, the occurrence of a *first* non-null value and the occurrence of a first *different* non-null value. Any attribute button which has both a *first* non-null and a first *different* non-null value in a given block is deemed 'permissible' and the crude 'informativeness' index, displayed to the subject, is merely the number of attributes that are deemed permissible. In practice the subject was required to maintain this index at a value in excess of two.

Various refinements are possible and some of them have been used. For example, it is easy to retain all values of y and y^* over a block and, in calculating $r^{\square}(n)$, to discount the contribution of all other than permissible attributes. Again, it is easy to compute a genuine informativeness index with respect to attributes regarded as *independent*; namely, if N_j^+ is the number of stimuli in a block for which the jth attribute has $y_j = +$, if N_j^- is the number of stimuli in a block for which the jth attribute has $y_j = -$, and if there are N stimuli in a block, the selective information function for each block of trials is

$$Attr\ Info = \sum_j \frac{N_j^+}{N} \log \frac{N_j^+}{N} + \frac{N_j^-}{N} \log \frac{N_j^-}{N}$$

It is more difficult to estimate the informativeness with respect to the subject's statements about patterns (at least, there is more latitude in choosing the appropriate form of expression).

5.4 Several types of steady-state controller were employed (using $r(n)$ in place of $\rho(n)$, and subject to the overriding constraint that the attribute values selected are minimally informative). In one scheme (Pask and Lewis, 1964; Pask, 1964a) the pattern blocks, including repetitions, were predetermined and the slides for each pattern were arranged in a projector cassette.

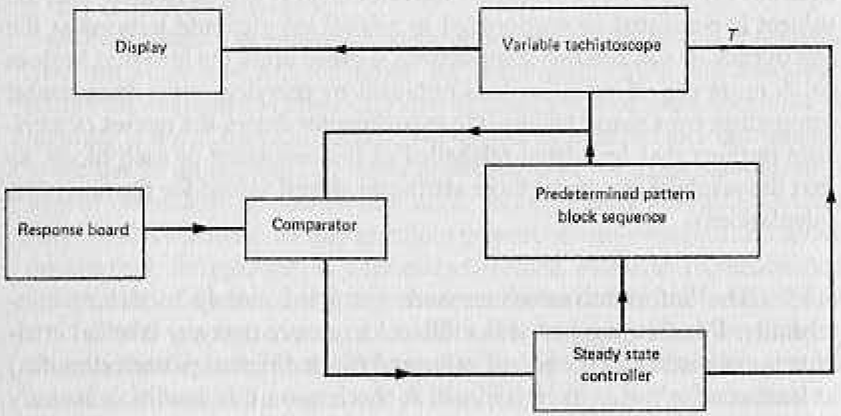


Figure 73 Steady-state control system for maintaining constant level of stimulus ambiguity.

The variable determining the ambiguity (analogous to the difficulty variable η was the tachistoscopic exposure interval T of each stimulus slide using a minimum value of 200 ms (reached if $\eta = \eta_{max} = 1$) and a maximum of 2·1 s. T decreased with η so that the exposure at each trial maintained $r(n) \approx \xi$ as in Fig. 73. Under these circumstances, the subject is bound to deal with an environment that becomes increasingly ambiguous as he learns to achieve self-consistency and typically he does so by choosing fresh attribute names. Most subjects, for example, start off by listing the readily verbalised but pedestrian names discussed in the introduction. Later, they progress to names that are idiosyncratic and often fanciful; moreover, they usually name more of the attributes (there is an obvious penalty in terms of consistency score for naming attributes that are not assigned values; hence, for gratuitously maintaining no longer used initial attributes in the current list). This arrangement can be operated within wide limits but there is a maximum of ξ for which it is impossible to catch onto regularities amongst the stimulus patterns and a minimum value of ξ for which the adjustment is swamped by the equivocacy of pattern classification even if the patterns are viewed for as long as desired.

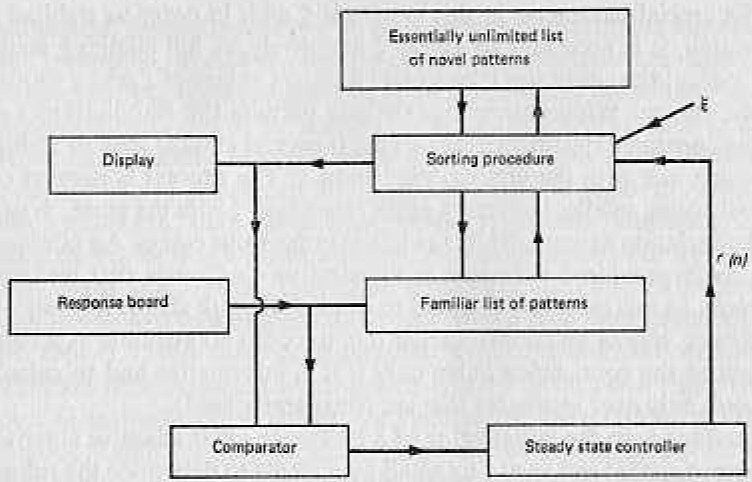


Figure 74 Steady-state controller that maintains constant level of stimulus ambiguity on the basis of differential data about a subject's self-consistency, $r(n)$.

5.5 Another scheme (Fig. 74) is more complicated (Lewis and Pask, 1964; Pask, 1971a). The slide blocks are constructed on the basis of differential data about an individual subject's consistency which is collected, as a set of scores that are averaged with respect to each pattern (over and above the computation of $r(n)$). As a general constraint upon the operation, any slide block that is constructed contains two or more repetitions of each slide so that a consistency value may be defined. Given that, consider the initial block (say with fifteen different patterns and forty-eight slides some containing two or more repetitions). At the end of the block either $r(n) > \xi$ or $\xi > r(n)$. If $r(n) > \xi$ then a sorting algorithm is used to remove a number, ζ, of slides from the block and thus to generate a fresh block. Those slides removed are the slides of stimulus patterns having the highest consistency scores; these are placed on a *familiar* list as candidates for use in a converse process as below. If $\xi > r(n)$ then a number, ζ, of slides from the familiar list (if it is defined) are used to replace the slides of patterns with the lowest consistency score. The sorting operation is repeated after each block of trials. In other words, if the subject detects a regularity in a certain part of his environment, as indicated by a high differential consistency score, the detected regularity, or the data underlying it, is removed from the environment when the overall consistency $r(n)$ is high. Conversely, if $r(n)$ is low, the environment is buttressed by patterns that previously obtained high differential consistency scores and may be expected to make the environment more tractable to the subject.

The crucial parameter in this system is ζ and, in order to stabilise the operation, it is necessary to adjust ζ adaptively. A hill climbing method will suffice; the controller tries widely different values of ζ and ξ choosing as the current value pair whichever one permits the stabilisation $r = \xi$ by the ongoing adjustment ('rate ζ') or, if several choices lead to stability, choosing one with the highest 'rate' term ζ. The process is iterated continually using smaller increment or decrements of ζ with the sense, + or −, of the variation determined by the initial or previous essays. An alternative and more provident technique is to select an $\langle \xi, \zeta \rangle$ pair that maximises *Attrinfo*. In this case, it is essential to display *Attrinfo* and to introduce the penalising checks mentioned earlier (i.e. to count an attribute as a contribution to the consistency index only if it is informative and to calculate *Attrinfo* only over attributes that are consistently used).

Provided with the initial value of ζ by one or other adaptive algorithm, the experimenter can vary ξ by small increments to determine the subject's resilience, before stability breaks down. The limiting values of $\langle \xi, \zeta \rangle$ or $\langle \xi, \zeta$ *Attrinfo*$\rangle$ provide an index of the subject's tolerances for ambiguity and, it is believed, of his tendency to innovate under circumstances that attach a premium to innovation. The characterisation appears to be stable, so far as an individual is concerned, over a couple of tasks. It was also found that the experience of working for a couple of hours, the length of a typical session, in a system of this kind gives rise to marked emotional effects. Some subjects, usually those who like to innovate, found the experience pleasurable or even euphoric. In contrast, other subjects found it distasteful and, if not allowed to opt out, disorienting (the effect persisted for long enough after the experimental session to prove embarrassing on administrative grounds).

Although these experiments were piloted and initial results were obtained over a decade ago it has been impracticable to pursue the matter in the interim. Work is starting again because there is, at this stage, a sufficiently comprehensive theory of interaction (the conversational theory, discussed later) to account for the affective changes that takes place and the clusters of stable limit values.

5.6 It is of interest to characterise the psychological constancy maintained by this type of SST system. There is a grain of truth in the view that a subject is held on a continuum of which the polar extremities are sensory deprivation (crass underload) and overload pure and simple. The trouble is that these terms are insufficiently exact: overload may mean 'processing rate' for repetitious inputs or 'high complexity', while the meaning of sensory deprivation depends upon the conditions as well as the length of deprivation, i.e. 'sensory deprivation' acts upon several aspects of man;

physiological, psychological, and so on. Moreover, each action has several facets. It is useful to place the constancy on a more specific continuum.

At one extreme, there is a guessing experiment where the subject faces a source which he knows to be random, i.e. he can make no rational sense of its structure. Typically, however, subjects adopt a 'superstitious' behaviour and make sense of the source by imagining a rule or regularity which is locally tenable but, since the source *is* random, generally unsound. At the other extreme, there are experiments such as those carried out by Warren (1961), Evans *et al.* (1967), Lilly *et al.* (1968) and Lilly (1972) in which a subject listens to a monotonous input (say from a tape loop, repeating the word 'kettle'). Under these circumstances, which approximate *one* kind of deprivation (the miniscule environment could not be *more* regular) the subject hallucinates other words; for example he hears 'pittle' or 'tittle' instead of 'kettle'. Moreover, these constructions are continued to form chains of fresh words, which suddenly enter perception. In terms of this continuum the SST can be used to balance the subject's operating region between the extremes of 'no *rule* to learn' and 'no *variation* at the input'.

6 Commentary on the Components of an SST System and their Interaction

At first sight these case histories bear witness to the success of a ludicrously simplistic method. If viewed as a causal or strictly behaviouristic system the subject has a 'goal' which is *simply* an observable behavioural regularity and a performance index, ρ, is a measure of the extent to which the actual behaviour fits the regularity the experimenter *previously* detected as a possible 'goal' and currently has in mind. A difficult index (η) could have a comparable status, i.e. it may have been observed that certain stimuli (or causative input events) give rise to less deviation from the ideal than others and this ordering could be modelled from the *experimenters* point of view, by something like a nesting of stimulus subsets. If repetition, perhaps with reinforcement, is held to transform this nesting (so that, as a result of adaptation, more difficult stimuli come to act as less difficult stimuli used to act) then the SST is merely a device for maintaining a certain *appearance* of performance by selecting a stimulus sequence, to suit the prevailing level of adaptation, from an appropriate subset.

7 Comment on Goals

These precepts ('goal directed' is an observed regularity, 'performance' is an external observer's construct) have already been questioned and this is an appropriate point at which to assert their insufficiency and counterfactuality. The SST works *in spite* of these suppositions; not *because* of them. Moreover, the steady-state system is, in fact brought into existence by quite different means.

(a) The subject is *goal-directed* in the thoroughly full-blooded sense of entertaining an intention or purpose.[1] That is, the subject's brain acts as a general purpose computer which executes goal-directed procedures, alias problem-solving procedures, alias intentions. Generally these procedures are non-deterministic programmes in the sense of Manna (1970) or fuzzy algorithms in the sense of Zadeh (1973).

(b) Under these circumstances, the environment is symbolically interpreted and selected by the subject, so, similarly, is the procedure he executes. The subject attends to a field of attention. The experimenter may determine what he attends to by specifying (as a programming operation) the procedure that he executes. Also, the experimenter may pose problems as conditions in which the goal is not satisfied. But he does not construct an input to which the subject reacts in any other sense.

(c) Under these circumstances the mooted equivalence between difficulty and simplification makes sense. A simplified problem is a partially solved problem and is less difficult than the undiluted problem, in so far as the solution-method used to obtain the partial solution is compatible with the methods realised by the current goal directed procedure.

In general, there is no guarantee that a simplification (partial solution) which seems plausible to the experimenter will serve as a simplification to the subject. In order to satisfy this condition it is necessary to have a model that replicates the subject's problem solving procedure. The cases considered are specially tractable either because the subject's method is dictated or because it is possible to choose an ordering of difficulties (either one ordering indexed by the scaler η or several ordering indexed by η_i) which systematically simplify in respect of *any* method.

(d) The various performance indices, ρ, are interpreted as estimates of the extent to which an internal-to-the-subject-hypothesis is confirmed or an internal-to-the-subject criterion achieved. All of them are measures of the subject's degree of belief in the rectitude of a goal directed, i.e. intentional or purposeful, action. In addition, of course, they index the extent to which the behaviour under this goal satisfies the external observer's criteria. But, first and foremost, they are estimates of subjective quantities and it is a subjective quantity (the subject's degree of certainty) that is stabilised by the SST. This point is obvious in connection with reflective measures, like r, as these make little sense outside such a framework.

(e) The goal directedness requirements ((a), (b), (c), and (d)) are satisfied by the existence of an experimental contract in which the subject agrees to

1. R. L. Gregory independently makes the same point, in saying that the brain under these circumstances, acts as a *deductive* mechanism.

participate. This is a normative or game-like scheme (not a behavioural paradigm) which is set up by a commanding or programming operation 'giving the experimental instructions'. It is useful to stress that the contract is *agreed* to by the subject. 'Giving instructions' is not a matter of presenting a specially forceful stimulus. The contract has a semantic and a pragmatic aspect. The subject, for his part, agrees to interpret symbols in a particular way; namely, as members of a field of attention. The experimenter, for his part, agrees to provide an SST controller that does so as well. The subject agrees to adopt a *role*, that is, to aim for a certain class of goals and the experimenter, for his part, agrees to furnish a controller that will cooperate (in the agreed domain) by simplifying problems posed under the goal so that the subject may be able to keep his part of the contract. Notably, the controller is designed under the assumption that the contract is kept. It does cooperate and is *able* to cooperate if, and only if, the subject has the intention of keeping the contract in the first place.

(f) Though this aspect is trivial for the case so far cited, the contract also has a syntactic component. The subject and the controller speak the same interaction language (formally an *object language* distinct from the observer's descriptive metalanguage) which has definite grammatical rules, which, for example, determine what is the form of a correct response or the construction of a logical subproblem (subgoal) of the original. The goal-directed procedures are programmed in this language, the rules of the game are stated in it.

(g) Although the theory does not demand a physiological interpretation, it is interesting that the subjective conditions (a), (b), (c), and (d) have reliable neurophysiological concommitants in man. These are described by Grey Walter (1969) who is responsible for much of the original work (this paper is essential reading).

By way of summary, consider a subject as a reactive device in receipt of stimuli that are pure stimuli, and no more than that.

If an auditory stimulus is presented to the subject it is usually possible to record an electrical response from the specialised auditory region of the subject's cortex. The same is true of a visual stimulus, or any other, as each stimulus evokes electrical responses from regions proper to its modality. In addition, for each effective stimulus that does evoke an electrical response, there is a transient electrical response discernible against a background of the usual electrical rhythms, in the frontal regions of the cortex. If traces of activity are laid out in time following the stimulus and if these traces are inscribed in computer storage and their values are averaged over a period of time in which several stimuli are delivered (so as to suppress the unwanted background by cancellation), then the transients are accentuated. But no other changes of response pattern are observed whilst the stimulus remains literally a stimulus: an event with no symbolic consequence. The picture is altered as soon as the subject is led to anticipate some consequence; either by repetition that indicates a relation between events or a requirement to make a motor response that acts upon the form of future stimuli or (immediately) by verbal mandate, i.e. saying 'this stimulus poses a problem to be solved'. Under these circumstances, the electrical response (sampled after averaging) in the frontal cortex displays a contingent negative variation (CNV) or 'expectancy wave' which increases in magnitude during the problem-solving phase and rapidly decreases as soon as the problem is solved and shortly before a motor response is elicited (supposing that an overt response is called for; in general, the solution may simply be kept in mind). The CNV is due to an interaction between loci dispersed about the frontal cortex and certain arousal systems in the reticular core and those parts of the limbic system responsible for orientation. The CNV can be extinguished by the repetition of stimuli uncorrelated with subsequent events and reinstated either slowly by correlated stimulation or immediately by command. Whilst the CNV is evoked, the brain is acting as a problem solver or symbol processor; it executes a programme with a simple or complex, 'If . . . then . . . else' structure. In these conditions it is possible to speak of information to the system (by the same token of uncertainty, equivocation or redundancy) and the system is goal-directed rather than reactive.

(h) The linguistic transactions of (f) call for a brief comment because 'language' is more than a convenient figure of speech (even though the languages involved in the system so far considered are admittedly quite trivial). Perhaps the most cogent empirical demonstration that a genuine (though trivial) language *exists* and is not merely blessed upon the system, comes from a phenomenon called 'participant interaction' that consists of misusage of the grammar *because* the subject is anxious to say more than he is legally allowed to say.

Under certain conditions, notably when an SST system is operated well below the overload point (low ξ) subjects typically engage in a game-like interaction with the controller. For example, they make responses which they know to be mistaken and which are thus illegal (according to the contract, the subject must make a correct response if he is able to do so, as he *is* at the moment in question). On examining sequences of illegal responses, or equally, on interrogating the subject after the experiment, it is clear that the illegal responses are made in an attempt to learn the algorithm in the controller and/or to influence the controller. For example, deliberate mistakes in a unidimensional SST system will reduce η (in a rather complex fashion if ρ is calculated from sequences of correct responses or by using an estimation rule). In a multidimensional system each η_i may be reduced and,

if the subcontrollers responsible for setting $\rho_i = \xi$ by adjusting η_i interact, as they do in most of the systems to be discussed in the sequel, then elaborate patterns of play and discovery are both possible and observable. If subjects use a system of transactions in an ungrammatical fashion (as they do) either judging from retrospective reports or by inference from records of their behaviour then, trivially or not, there must be a grammar. The construction is buttressed by noting that the phenomenon of illegal usage is inhibited, and the play converted into legal usage, if the subject is provided with transactions that logically allow him to question and issue (conditional) commands to the controller, by means of which he is able to bring about more expeditiously the controller-behaviours he manages to bring about in a very cumbersome manner by illegal usage or misusage of the original transactions. As a point for later reference the augmenting transactions, that allow the subject to interact with the SST controller, operate or comment upon the original problem-solving interaction and are conveniently assigned to a *higher* level of interaction. In other words, this primitive language is stratified into a lower level concerned with problem-solving and an upper level concerned with comments to the controller about problem-solving.

8 Outline

The next chapter is concerned with an extension of the SST along these lines; namely, adaptive control. One example has already been noted (pattern ambiguity, the adaptive regulation of ξ and ζ). The adaptation of the controller, even in this case, compensated for a kind of learning on the part of the subject which could not be satisfactorily represented as a form of adaptation. However, this learning was not, in itself, relevant to the constancy being maintained and it was not emphasised. In the adaptive control systems to be discussed, the constancy is preserved in order to influence the learning process itself (for example, to maximise the individual subject's rate of learning) and the controllers in question are quite explicitly built as teaching and training devices. Since most of them are devices for instructing skills, it is still possible to minimise the attention-directing or exploratory *modes* of a human being, i.e. to minimise the issue of learning strategy but not to suppress it altogether. Moreover, most of the systems rely upon proficiencies ρ, ρ_i, as estimates of an underlying degree of belief.

8 Adaptive Teaching Machines

A steady-state system, used in connection with a task such as teleprinting (Pask *et al.* 1969a), speed reading or various tracking and compensation activities, has a training function in its own right since it keeps the subject (student or trainee) poised in an operating region, where the problems posed by the skill are difficult enough to be taxing without becoming unintelligible; as they would be in extreme overload. In particular, it is sometimes possible to choose ξ so as to maximise the overall rate of adaptation. As a specific example, consider the following *code* learning or *transformation* task which has been used widely in experimental studies and which underlies the simulation described in Chapter 6.

Two versions of the task were employed; one with a six-fold input (Pask and Lewis, 1967) and one with an eight-fold input (Pask and Lewis, 1968). The latter is described below.

1 Code Learning and Transformation Tasks

A subject is presented with an effectively unlearnable sequence of inputs, spaced equally in time, which are configurations of up to four illuminated signal lamps, selected from a group of eight lamps. Before the experiment begins, he is told that a transformation rule, Ω, relates the group of signal lamps to a row of eight response buttons; he knows this rule and is able to recite it. Given an input, x, he is required to solve the problem of producing a response, $y = \Omega(x)$, within a deliberately restricted interval, δt. This response is complex, for he must press several buttons to denote the transformation of the signal configuration.

The size of the signal configuration, the value of δt and the inter-trial interval, Δt, are chosen so that (a) a fully proficient subject can perform the skill accurately, but (b) a novice finds the problems posed by x in the context of Ω unsolvable (e.g. his uncertainty regarding the value of y when x is given, is such that he is grossly overloaded). Typically, $\Delta t = 5{\cdot}5$ s, and $\delta t = 4$ s.

In these conditions a novice can neither perform the skill nor learn to

perform it unless he is provided with external assistance. If a marginal amount of external assistance is provided by a steady-state regulator, then the subject is forced to exhibit various problem-solving behaviours which would, in more relaxed circumstances, be unobservable.

In order to start a learning process, the regulator must reduce the difficulty of the situation. It does so by 'simplifying', or 'partially solving' the problems posed by the inputs. A combination of two simplifying procedures is used for this purpose.

One procedure involves the reduction of the number of lamps in a signal configuration. The other consists in providing cue information that partially specifies the correct response. Both of these procedures, as well as their combination (to form a conjoint simplifying operation), have been checked empirically, that is, the percentage of correct responses elicited by a sequence of more simplified input problems is never less than the percentage of correct responses elicited by a sequence of less simple input problems and, with the exception of a fully proficient subject, it is greater. The snare to be avoided is that some intuitively acceptable simplifying procedures suffer from an 'inversion' noticed by van der Veldt (1928, cited in Woodworth, 1950). If a subject is taught to respond to a complex stimulus by way of simple exemplars, the subsequent presentation of a less complex stimulus may give rise to more, rather than less, difficulty. The subject acquires a perceptual motor organisation applicable to the complex situation; the organisation apposite to less complex situations has decayed, since it amounted only to cognitive scaffolding needed to reach and deal with the complex problems.

Specifically, if m designates the number of signal lamps in an input configuration, then at any trial the equipment quasi randomly and equiprobably selects one of the roman numbered columns in the table shown and four levels of simplification are obtained by selecting one row (ordered in decreasing difficulty, η, from top to bottom) where the letters designate signal lamps.

m	I	II	III	IV	V	VI	VII	VIII
4	*abch*	*abdc*	*bdce*	*cedf*	*egdf*	*gehf*	*gafh*	*gahb*
3	*abc*	*bdc*	*dce*	*def*	*efg*	*gfh*	*hga*	*ahb*
2	*ac*	*bd*	*ce*	*df*	*eg*	*fh*	*ga*	*hb*
1	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*

The other part of the simplifying operation provides cue information through a further row of lamps adjacent to the response buttons and within the interval, δt, in which a response can be submitted and accepted. Cue information partially specifies a correct response, $y = \Omega(x)$. It is provided if a variable $z = 1$ and is not provided if $z = 0$. Specifically, if $z = 1$ and if x is an m-lamp configuration, then after the subject has selected $m = 1$ correct members of y, he is automatically provided with a specification of the remaining selection that he must make in order to achieve a correct response.

These procedures are combined to provide eight degrees of difficulty indexed by the variable, η. The combination of procedures is determined by the listing shown below; values of η being determined by the steady-state control mechanism. As an initial condition for the system $\eta = 0$

Value of η	8	7	6	5	4	3	2	1
Value of m	4	4	3	3	2	2	1	1
Value of z	0	1	0	1	0	1	0	1

The response, y, is complex and is obtained as follows. The subject responds by selecting a configuration, y, of response buttons and is required to select the particular configuration $y = \Omega(x)$. If Ω is a permutation of signal lamps with respect to response buttons and if the input at the nth trial, $x(n)$, has m components, then a correct response, $y(n) = \Omega[x(n)]$, also has m components. In order to count as a correct response, $y(n)$ must be completed within the interval δt during which $x(n)$ is exposed; and the response buttons are automatically rendered ineffective after δt has elapsed.

However, the response buttons need not be pressed simultaneously and the subject is permitted to construct y in any order he pleases. To avoid an unwanted load on his immediate memory capacity, each response button is associated with a 'lock' relay which is energised if the corresponding button is pressed and remains energised until 0·5 s after the end of δt. Whilst the relay is energised, an indicator adjacent to the response button is illuminated and the set of illuminated indicators provides a visual record of y. The interval δt is terminated by the disappearance of the input signal and the coincident illumination of 'knowledge of results' indicators, in register with the response buttons, that indicate the correct response $\Omega(x)$. This presentation persists for 0·5 s. after the end of δt (so that it coincides with the image of y thus allowing the subject to compare $\Omega(x)$ and his actual response, y). Coincidentally, the subject receives an auditory and visual signal (buzzer and lamp) to indicate whether his response is completely correct (all response selections correct) or mistaken. At an instant, $\delta t + 0{\cdot}5$ s after the appearance of the stimulus, the 'lock' relays are de-energised and the indicators are extinguished. There is a rest interval of 1 s and the next stimulus is presented $\Delta t = 5{\cdot}5\text{ s} = \delta t + 0{\cdot}5\text{ s} + 1\text{ s}$ after the last.

A proficiency index, ρ, is calculated by averaging correct response frequencies, either latency weighted or not, over coherent blocks of trials.

As in the last chapter the steady state control mechanism adjusts η to maintain a null-point value of ρ, at or near to a null-point value ξ.

In a skill of this type, many ξ values are stable, provided they are within a certain range. But, regarding the system as a tutorial arrangement, the preferred value is one which maximises the rate of adaptation or satisfies a similar terminal performance condition (near to the limit at which the steady-state regulation has nearly exhausted the variety it is in a position to introduce); for example, when $\eta = 8$ and when $\rho > 0{\cdot}8$. Call the number of trials needed (by a given individual) to reach this terminal performance condition, T. The mean values of T are determined for groups of students (ten in each group for the data shown below) working under different, but stable value of ξ and the T means are compared in order to ascertain the most 'effective' value of ξ which minimises the value of T.

	T(mean)	T(standard deviation)
$\xi = 0{\cdot}55$	247·4	38·34
$\xi = 0{\cdot}65$	386·5	68·78
$\xi = 0{\cdot}75$	503·4	79·89

Using Jonckheere's (1954) trend test, the evident trend in $T(\xi)$ is significant at the 0·1 per cent level ($0{\cdot}001 > p$). Further reduction in ξ leads to instability in the steady-state control system. Thus, for a skill of *this* kind, adaptation rate is maximised by choosing ξ near to the overload point and thus presenting problems that are as difficult as possible, though not unintelligible. The adjustment is a typical 'tuning' operation (in the sense of Chapter 4, Fig. 32).

2 Adaptive Regulation

Such a choice of ξ relies upon an assumed constancy of the operating region boundaries between individuals. In general, this assumption is untenable; moreover, it is usually untrue that an optimum ξ remains fixed, even for one individual, during a training session.

In order to estimate and establish effective values of ξ under *these* circumstances, it is possible to use adaptive machines that operate as local optimisers. Such a machine selects values of ξ to determine the stable boundaries; it selects *test values* of ξ, within these boundaries, and, by successive sampling, picks whatever value maximises a function such as the (approximated) rate of increase in η with ξ (and thus ρ) held at the currently selected test value. From this point onwards, the current value of ξ is maintained for most of the time at the 'best' value so far discovered, but some time is allocated to sampling performance under a *test value* which is incrementally distinct from the current value. If the current value is superior, then an increment in the opposite sense is selected as the next test value and if not the next current value becomes the test value. The process is iterated and is chiefly limited, in practice, by the number of trials over which the rate of change in η can be estimated. Such a device is shown in Fig. 75. This, and

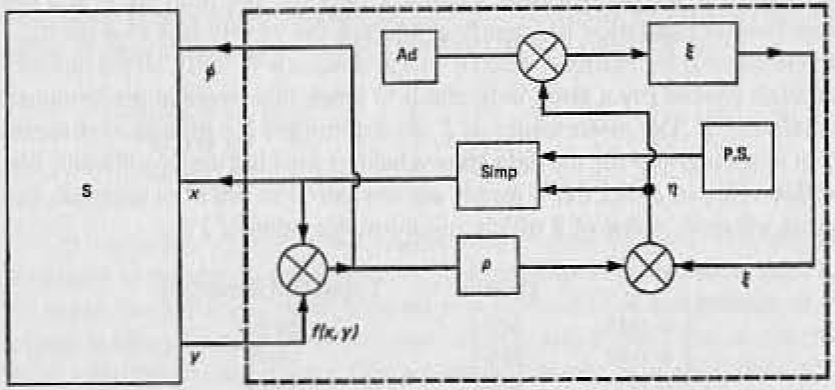


Figure 75 Adaptively controlled system: x, input; y, response; ϕ, knowledge of results; P.S., source of problems; Simp, simplification; $f(x,y)$, correct response function (for example, 1 if $\langle x,y \rangle \in \Omega$; 0 if not); ρ, performance index; ξ, steady-state criterion; η, difficulty index; Ad, adaptation rule; s, student; ⊗, comparators, as in control engineering notation.

other more complex arrangements (described in the following sections) were programmed on a special-purpose computer (Plate 5) as well as being built into numerous experimental or industrially used equipments (Plates 6 and 7).

2.1 Two variations on this theme indicate the scope of adaptive regulation and the adaptive machine as a training mechanism. One variation is based upon the code-learning or transformation skill of section 1, and the other upon a multidimensional steady-state system, exemplified by a trajectory recognition and target interception skill, briefly described in the next section.

2.2 The cellular display of Plate 6a consists in a matrix of signal lamps which are turned on and off at variable intensities to form a confusing or 'noisy' background. Trajectories of illuminated signal lamps, moving from left to right across the display, can be superimposed upon the 'noisy' background at various intensities and thus (if other things were equal; for example, if all trajectories were equally recognisable) would correspond to varying degrees of prominence of discriminability.

There are eight classes of trajectory, differing in their pattern of motion. Members of each class are sampled equiprobably, but each class of trajectory may be presented with higher or lower intensity (the lower the intensity for the *i*th class, the greater is η_i).

The trainee is required to intercept the target responsible for a trajectory by using one or, if need be, both of two vertical interceptors that move up the right-hand side of the display. He is warned if a target *may* be in the offing but is *not* told which class of trajectory will characterise its motion. This information, which is essential in order to intercept the target at all and which must be received as early as possible to achieve a provident interception, is garnered from observing the first phase of a trajectory and combining the clues so obtained with background knowledge about the possible classes of trajectory (in certain variants of the task, also from positional information, furnished by the cue lamps bordering the display, about regions where targets are likely to be found).

Training[1] begins with $\eta_i = 0$ for all $i = 1, \ldots, 8$. A proficiency measure ρ_i is computed with respect to each trajectory class as a function of success and the number of interceptors used up at a trial. The resulting ρ_i values form the input to eight subcontrollers that adjust the η_i, to maintain $\rho_i = \xi$. With $\eta_i = 0$ the task is quite easy to perform. At the upper limit of loading and it is impossible for anyone to maintain perfect and provident interception.

The adaptive form of this block of eight steady-state subcontrollers contains an additional overall controller which prescribes separately determined values; ξ_i, to maximise the expected rate of increase in the product of the stabilised η_i. It is shown in Fig. 76. In fact, it is difficult to operate the system in a stable manner without the adaptive control loop. Typical data are shown in Fig. 77.

2.3 An obvious extension of the code learning or transformation task described in section 1 is to require a student to deal with two or more coding rules; say Ω_1 and Ω_2. For example, the tenure of rule, Ω_1 or Ω_2 is alternated, haphazardly, from one *block* of six or eight trials to the next (though the student is informed which code rule is applicable at any trial *block*).

In changing the transformation or code rule, it is, of course, necessary to change the steady-state controller for there is no reason whatever to suppose that Ω_1 and Ω_2 are homogeneous. Thus there are two subcontrollers, one

1. Several variations on this training system have been used from time to time. For example, in one of them, the relative frequency of appearance of members of the trajectory classes is regulated, as η_i, in place of display intensity.

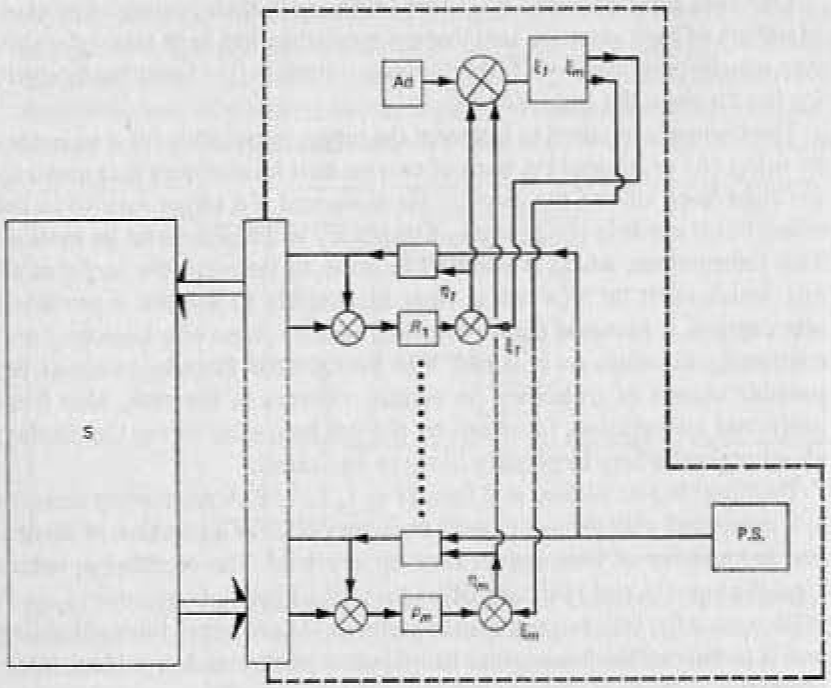


Figure 76 Multidimensional adaptive system. Note single problem source (P.S.). The system has m controllers and one overall controller. Apart from the subscripts, the notation is as presented in Fig. 75.

for Ω_1 and one for Ω_2; one varies η_1 to maintain $\rho_1 = \xi$ and one varies η_2 to maintain $\rho_2 = \xi$.

Under these circumstances, especially if the rules are chosen to have a few features in common but most features distinct, there is pronounced interference which hampers learning to the extent that most students are unable to acquire the skill unless the time constraints are relaxed. However, with the same time constraints imposed, the skill is learnable if the student is allowed to choose the rule he will concentrate upon during a trial block (in a pause prior to presenting the block inputs) under the caveat that he must ultimately alternate the code rules in a 50 per cent fashion.

Data from two subjects learning in this condition are shown in Fig. 78. The shaded region between the curves for η_1 and η_2 is an index of the degree of interference between acquiring Ω_1 competence and acquiring Ω_2 competence and the code rule applied over a given trial block is shown on the upper line (shaded for Ω_1 or blank for Ω_2).

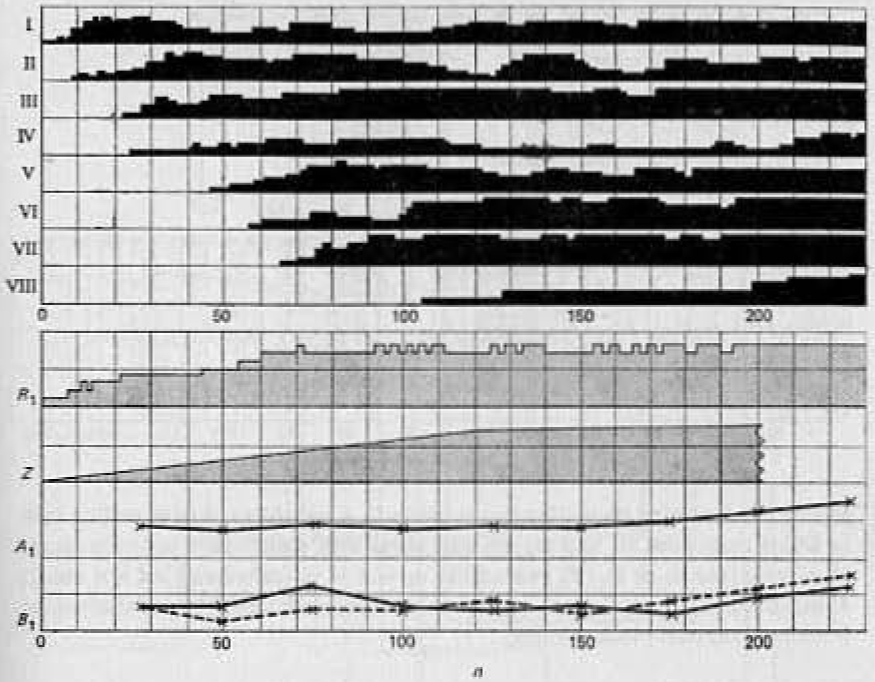


Figure 77 Learning curves for one subject (see text): I to VIII, values of η_1 to η_8; R_1, mean of the η_1; Z, product function used as control variable; A_1, mean value of the ρ_1; B_1, differential mean ρ values for double interception (solid line) and single interception (dashed line); n, trial number.

2.4 The adaptive form of this training system is shown in Fig. 79. The overall controller does not change the steady-state parameters, ξ, of the subcontrollers (though it *could* do so, in principle). Instead, it executes a simple selection strategy, which consists in assigning probabilistic weights π_1 and $\pi_2 = 1 - \pi_1$ to the selection of Ω_1 (and its subcontroller) or Ω_2 (and its subcontroller) at the beginning of a trial block. In the typical, simplest case examined, the value of

$$\pi_1 = \frac{\eta_{max} - \eta_2}{(\eta_{max} - \eta_1) + (\eta_{max} - \eta_2)}$$

is applied as a biassing input to a quasi-random selection mechanism (since there are only two subcontrollers and $\pi_1 = 1 - \pi_2$ this input is sufficient). In general, an adaptive controller of this kind furnishes $M - 1$ biassing inputs and a selection is made between $M > 2$ subcontrollers; for example,

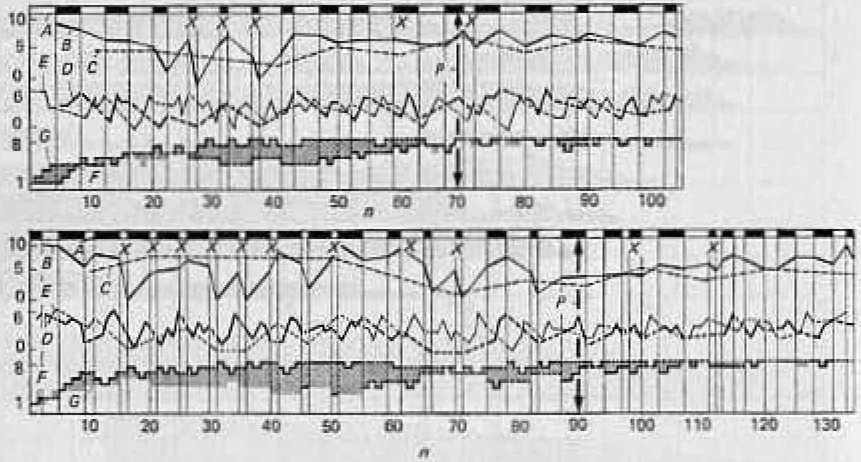


Figure 78 Learning curves for two subjects in a transformation or coding task in which code rules Ω_1 and Ω_2 are alternated with steady-state subcontrollers: *A*, current rule Ω_1 or Ω_2; *B*, probability $\pi_1 = 1 - \pi_2$ (calculated but not used); *C*, mean ρ; *D*, ρ_1; *E*, ρ_2; *F*, η_1; *G*, η_2; *P*, point at which definite strategy of alternation is adopted; *n*, trial block number.

in Pask (1964b). That reference describes an adaptive system for training in electronic fault detection.

Some typical data are shown in Fig. 80, which should be compared with the curves shown in Fig. 78. The salient difference, apart from the more rapid learning in the adaptive system, is the suppression of interference.

2.5 One further extension of the system is worth examination. There are good reasons to believe that the 'free learning' of Fig. 78 is ineffective because students are substantially unaware of their own ability to assimilate the subskills learned under Ω_1 and Ω_2 or to integrate them into one coherent skill, as they must do to perform effectively. For *all* students there is a breakpoint, shown as 'P' in Fig. 78, when a definite strategy is adopted after which the student does come to terms with the task. Some students (the lower graph in Fig. 78 for example) recognise that a strategy is an *operational* requirement very tardily, though most people are willing to pay lip service to it; that is, they talk of 'planning' and the like. Incidentally, it does not follow that people with the most readily verbalised 'plans' are the most effective learners. They often tackle the task with spirit and high motivation but are no more successful, on average, than individuals who have plans that *evolve* in the course of learning and cannot be neatly described (at any rate, before the event). These comments are supported by a specific sub-study described in a technical report (Pask, Mallen and Scott,

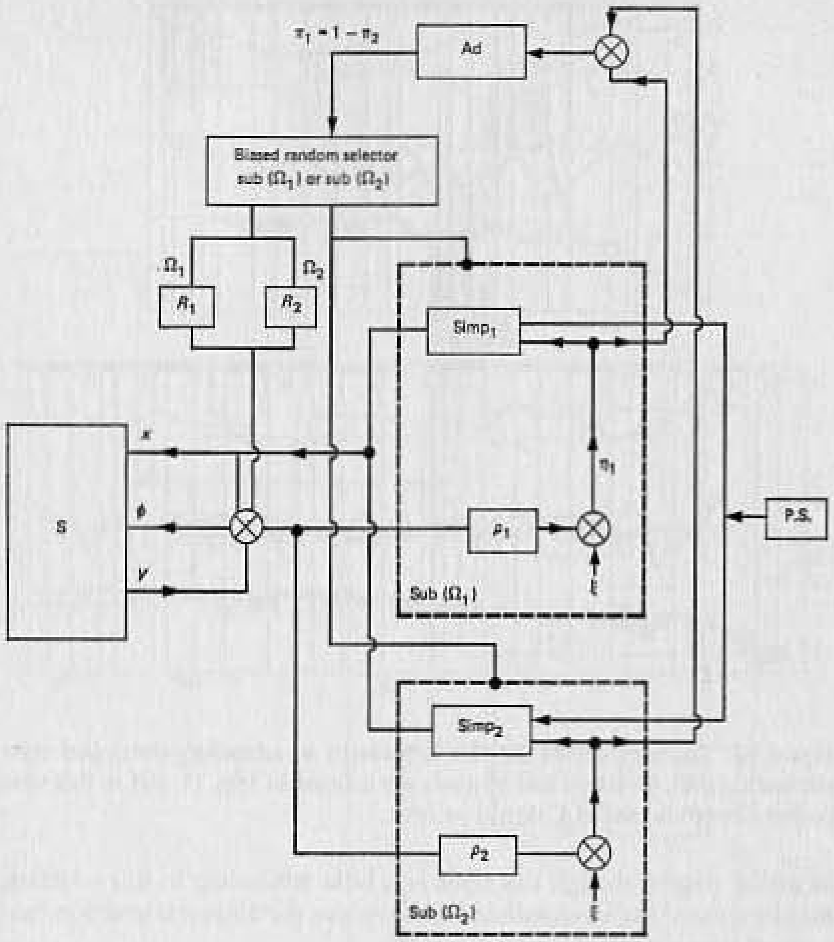


Figure 79 Code learning with subskill alternation. Subcontrollers for each subskill are enclosed in dashed lines and designated sub (Ω_1) and sub (Ω_2). Ω_1 and Ω_2, coding rule inscriptions as function tables; symbol ●, selection as alternatives, one and only one at once. Other notation is as presented in Figs. 75 and 76.

1968), carried out in connection with the overtly strategic learning task discussed in Chapter 11.

On the other hand, though learning in an adaptive system is impressively fast and free from interference (Fig. 80), this effect is most certainly not eliminated (for example, the right-hand graph). At least some of the residual interference is due to the participant interaction, noted in the last chapter, and the following arrangement (called an 'adaptive metasystem'

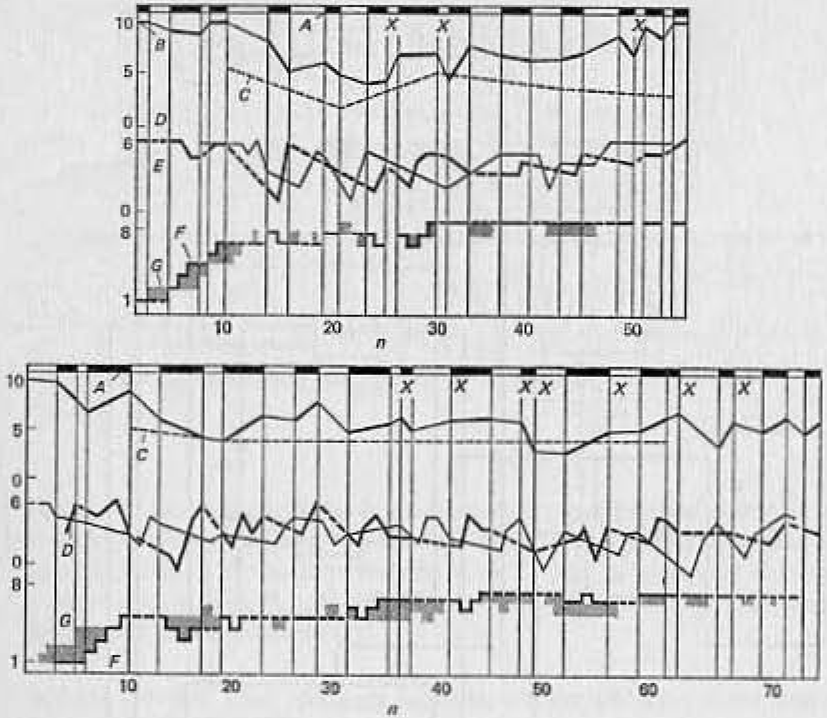


Figure 80 Learning curves for two subjects in an adaptively-controlled code alternation skill. Notation and symbols are as used in Fig. 78, but in this case probability (*B*) is used as a control variable.

in earlier papers, though the name is a little misleading in this context), may be regarded as an expedient for converting participant interaction into a 'legal' interaction.

The arrangement brings about a compromise between the student's choice of the code rule to work on and the overall controller's selection. For each trial block, here indexed *n*, the overall controller calculates the dominance function

$$Domin = \sum_{i} p_i \cdot \eta_i (1 - \|\pi_i - \tfrac{1}{2}\|); \; i = \{1,2\}$$

with the properties that for $n = 0$ *Domin* $= 0$ and that for $n = T$ (the terminal condition, here requiring proficiency in performing a hybrid or amalgam of both subskills) *Domin* $= 1$. As suggested by its name, *Domin* determines the overall controller's degree of dominance over the student. At the beginning of each trial block, the student is allowed to submit his *proposal* for which subskill shall be rehearsed (either deterministically by

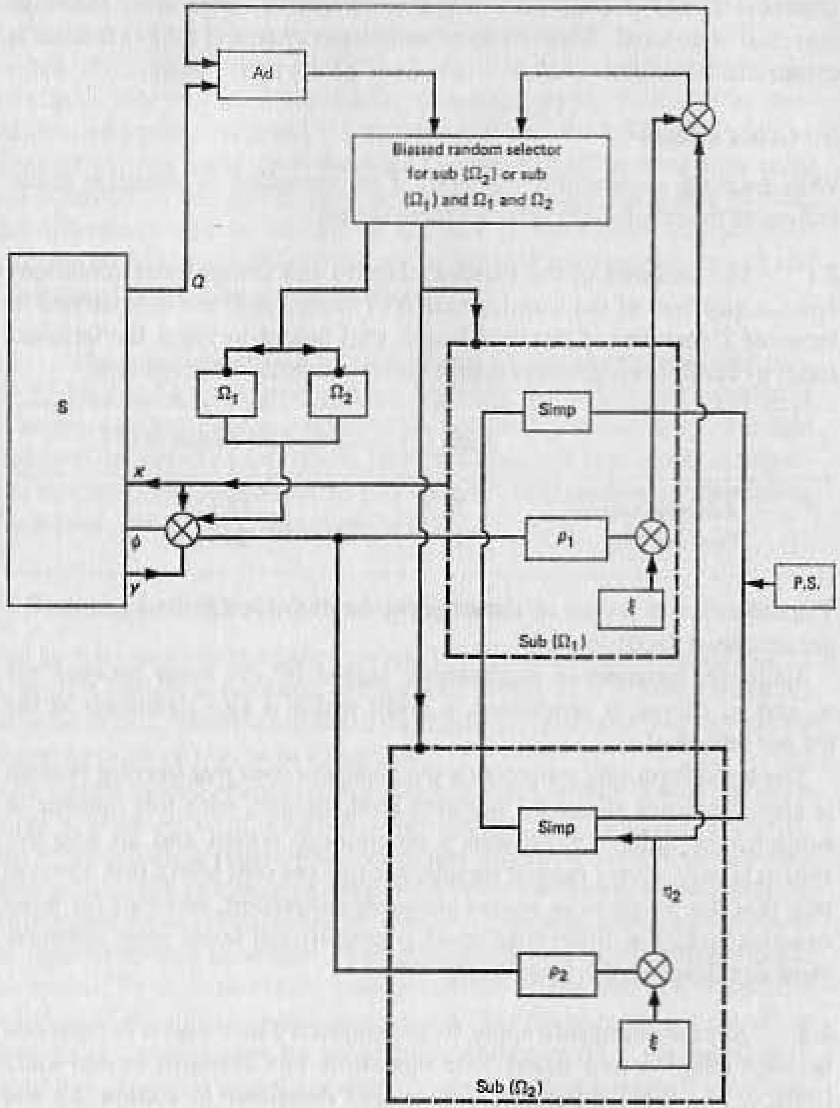


Figure 81 Adaptive control and compromise (or metasystem) solution: *Q*, student's selection; other symbols are as used in Fig. 79.

pressing a button or probabilistically by setting a confidence estimation scale expressing his belief that Ω_1 or Ω_2 should be rehearsed). The weight given to his proposal depends upon the prevailing value of *Domin*. If *Domin* $= 0$, the student's choice or his probabilistic bias is decisive; if

Domin = 1, the overall controller's bias; in the latter case, alternate rehearsal is ensured. Between these values the system (Fig. 81) reaches a compromise solution.

3 Gross Effects

With training applications in mind, it is interesting to compare crude indices of the relative efficiency of these systems.

3.1 The influence of the various adaptive and compromise conditions upon acquisition of the transformation or coding skill are summarised in terms of T (number of constant-length trial blocks to reach the terminal state) by the following figures (again for ten subjects in each group).

	Mean (T)	Standard deviation (T)
Adaptive system	76·2	13·14
Free learning	116·4	20·81

The difference in favour of the adaptive condition is significant at the 0·1 per cent level ($0{\cdot}001 > p$).

Similarly, interference suppression, judged by the areas between the η_1 and η_2 curves, is suppressed, a result which is also significant at the 0·1 per cent level.

The trend favouring *compromise* over *adaptive* over *free learning* systems is also significant at the 0·1 per cent level, though, with this number of subjects, the difference between a *compromise* system and an adaptive system is only of very modest significance (0·5 per cent level). It is however true that the *compromise* system enhances motivation, and that for more complex tasks the differential result is unequivocal (even gross measures yield significances of 0·1 per cent).

3.2 Similar comments apply to the trajectory task and a comparison between adaptive and steady-state operation. For example, in one study (Pask *et al.*, 1967) using the arrangement described in section 2.3 and groups of eighteen and twenty-five subjects (with n = number of trials) we obtained the figures shown below.

	Mean (T)	Standard deviation (T)
Steady-state	558	86
Adaptive system	1225	215

The difference in favour of the adaptive process is significant at the 0·1 per cent level on a non-parametric test (Mann–Whitney U).

Here, there is also data on long-term retention (over a few weeks) which is substantially perfect. Very similar comments apply to the other procedures and conditions noted in the discussion, and the difference in favour of the adaptive mode is rarely less than 1 per cent significant, even for small and relatively *ad hoc* studies that have been carried out from time to time. The differences tend to increase as the task is made more complex (for example, by introducing predictive and positional cueing information) and if individual differences are pronounced.

3.3 The transformation and code-learning tasks closely resemble real-life keyload skills (shift positions or duplicate keyboards are analogous to multiple codes). Not surprisingly, the laboratory results tally with field studies of devices like SAKI (Pask, 1958, 1961) when it is properly incorporated in a training scheme and to very detailed field studies of teleprinting (Pask *et al.* 1969a; Scott and Pask, 1973).

4 Some Inherent Limitations of the System

Just as the steady-state control system (or the steady-state technique, in general) is subject to the limits summarised in Fig. 32 of Chapter 4; so the *adaptive* control system is subject to the limits of Fig. 33 and the *compromise* system to those of Fig. 34 in Chapter 4.

At a more mundane level (though the argument reflects exactly the same ideas) a single variable regulator relies upon a rank ordering of inputs; that is, the problems engendered by a higher value of η must be consistently more difficult than those engendered by a lower value of η. Consistency, in the required sense, is established by checking that there are no inversions of the type mentioned in section 1 and intuition is fairly often misleading in this matter. By the same token, a many-variable system relies upon a partial ordering of difficulties (in the same sense). The primary criterion may be taken as an ordering over the problems of each subskill in respect of each variable, η_i, together with the possibility of embedding subsets of problems, characteristic of the integrated skill, in a further ordering system (Gaines, 1972). The chief difficulty occurs due to the fact that a skill can be assembled in many legitimate ways out of its subskills; different methods can be adopted by different students or the same student on different occasions.

If a skill does have these very restricted properties it is called a structured skill (Pask, 1964b; Lewis and Pask, 1965) and it turns out that the most interesting tasks do not belong to this category. For all that, the multi-variable adaptive system and the compromise system have a flexibility that a steady-state control system does not have; the regulator is *not* confined to compensating for changes of attention that are localised, effectively, in one η related universe (like the regulator in Chapter 4, Fig. 32). In a non-trivial, though still very special sense, the regulator may deal with changes of attention over several distinct universes (Chapter 4, Fig. 34).

5 Self-Organisation and Uncertainty Regulation

This section is devoted to the following theoretical notions.

(a) The adaptive regulator (including the *compromise* case of section 2.5), when it is interacting tutorially with a student, is an hierarchically organised system with *Lev 1* and *Lev 0* as levels of control. It is also part of a learning and teaching system in which the learner cannot be fully disentangled from the teacher (Chapters 5 and 6).

(b) A steady-state regulator and the system it controls fit into a degenerate form of this paradigm.

(c) Performance indices and difficulty indices may be expressed as selective uncertainty/information indices.

(d) As a result of this, the adaptive/compromise systems are seen to be self-organising systems in Von Foerster's (1960) sense (Appendix A).

(e) The steady-state system is a degenerate form of self-organising system.

(f) It is possible to interpret the selective uncertainty/information indices as indirect estimates of subjective uncertainties and to replace them (if the sampling conditions of Chapter 1 are satisfied) by direct estimates. Due to the nature of perceptual motor skills, it is usually impracticable to institute a change to direct estimation. But the possibility of doing so is important since direct estimates underlie the uncertainty regulation procedures, apposite for the intellectual and strategic learning to be discussed in Chapter 11.

5.1 The constructions promised in (a) are shown in Figs. 82 and 83, where the hierarchical components may be conceived as finite-function machines (Chapter 2) or, in the lower boxes, aggregates of these. A more realistic interpretation, and a slightly more liberal one, is obtained (as promised in Chapter 5) by regarding the constituent finite automata as fuzzy automata or fuzzy algorithms (under execution in a generally non-numericised manner).

5.2 For the steady-state system, the last clause (fuzzy non-numericised execution) is excluded; that is, such a process cannot be *observed* or incorporated into the regulator design.

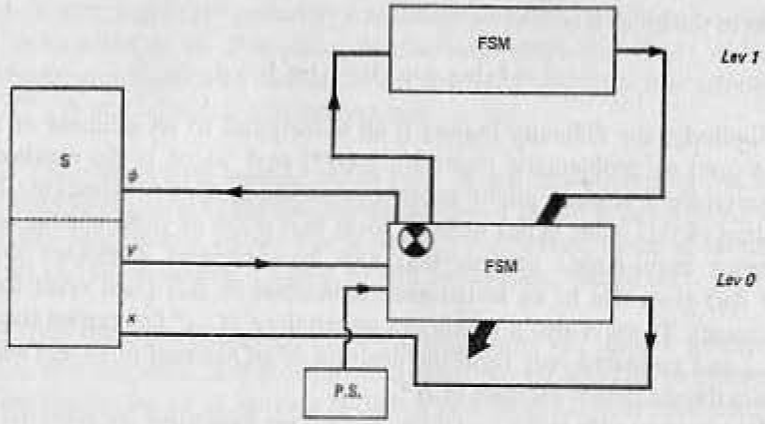


Figure 82 A finite-function machine construction for the adaptive teaching/learning system. Symbols and other notation as used in Chapter 2. *Lev 0*, *Lev 1*, levels in control hierarchy; FSM, finite-state machine; P.S., problem source shown as input to lower FSM. Dotted line across student indicates that he may be (minimally) imaged as a pair of FSMs.

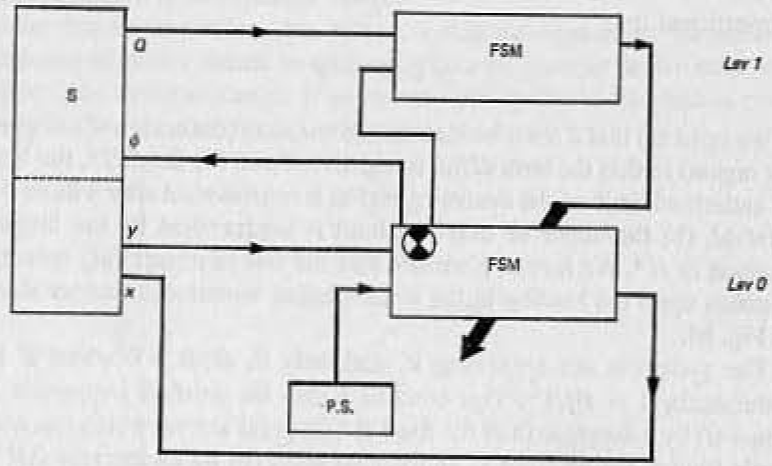


Figure 83 A finite-function machine construction for the compromise teaching/learning system: *Q*, student selections; *Lev 0*, *Lev 1*, levels in control hierarchy. Other notation is as used in Fig. 82.

5.3 A common feature of all the performance indices ρ is that, with respect to a *given goal* (to bring about $\mathscr{R}$, for example), they correspond to selective uncertainty/information indices (as in Chapter 1). The general form of the index is:

$$H = (\rho \log \rho) + (1 - \rho \log 1 - \rho).$$

Similarly, the difficulty indices η all correspond to an estimate of the structural or problematic uncertainty (H^*, say) which is the maximum uncertainty a student might experience at the instant in question. The scaling of this index is not defined, from that point of view, but the consistency requirement ensures that any instantaneous increment ΔH^* (or $\Delta\eta$) gives rise to an instantaneous increase in ΔH (vice versa for a decrease). There is also a maximum uncertainty H_{max}^* (corresponding to η_{max} and evaluated only from the position of an external observer) which limits the maximum increase in H^*.

For systems with many variables (ρ_i, η_i) these quantities are vectorial (as in multivariate information theory: Chapter 1).

Let J represent a transmission (estimate in terms of selective work: Chapter 1) so that J is interpretable as the time integral, over a unit such as a trial block, of an adaptation rate $-dH/dt$. On response to a transient ΔH^*, or for fixed H, the student's loading (Pask and Mallen, 1969), and an *indirect* index of the subjective uncertainty a student experiences, is proportional to

$$H = H^* - J$$

We posit (a) that J must be maintained constant (the notion of an operating region) so that the term dH/dt is negative. Thus, for fixed H^*, the lower or underload limit on the operating region is contravened after a finite time interval. (b) the upper or overload limit is contravened by too large an increase in H^*. We further postulate that the rate of uncertainty reduction depends upon the loading in the single-peaked monotonic manner shown in Fig. 84.

The system is self-organising if, and only if, $dz/dt > 0$ where Z is a redundancy $1 - H/H^*$. This condition may be satisfied (Appendix A), either (i) by a decrease in H for fixed H^* (case (a) above) which has necessarily limited tenure since H approaches zero, (b) by an increase ΔH^* in H^* with H constant (which corresponds to an increase in η). However, provided that ΔH^* is not too large, this increase is compensated by learning and represented as an increase in the rate integral J, or (c) a system may be self organising due to the coordinated or coupled regulation of H and H^* (H changing as a result of the student's *goal-directed* adaptation and H^* due to the tutor or regulator).

This is a macroscopic picture of the tutorial operation in an adaptive system which is designed to maintain a constant loading or a constant value

of H, by a variation in H^* guided by a model in the subcontroller that represents the student as an H-reducing mechanism (within the limits of his operating region), and a further model in the overall controller that adjusts η (or H^*) to maximise the H reduction rate.

5.4 The steady-state system, on its own, is still self-organising but is degenerate to the extent that the overall controller model is missing. As a result, self-organisation occurs due to the stepwise juxtaposition of operations (a) and (b) in section 5.3 (not, except by accident, due to (c)).

5.5 One difficulty is that all those identifications rest upon a presupposition that responses are goal-directed and that, on *average*, reduction in uncertainty is due to an increase in *correct* certainty. Another difficulty is that, though the sense (+ or −) of the change in the estimated variables (H, H^*) is veridical, their absolute values are only calculable in respect of an external observer, not in respect of the student.

Both difficulties can be surmounted if the quantities H and H^* are regarded as *indirect* estimates of underlying subjective uncertainty/information/indices and if the values of these subjective indices are used in the control equations or in regulator design.

Under this interpretation, the system is still self-organising. However, the indirect estimates should be replaced whenever *possible* (in the sense of Chapter 1) by direct estimates. If so, values appropriate to the student can

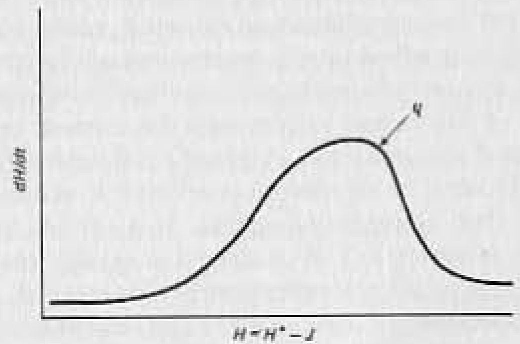


Figure 84 Graph showing the form of the postulated relation between uncertainty reduction rate and the student's level of uncertainty. η is a point corresponding to the optimal steady-state criterion expressed in terms of uncertainty/information/index and thus to a desired loading. This form of relation is supported by loading studies and, more convincingly, by the stable operation of an adaptive control mechanism, designed on the hypothesis that such a relation exists.

be attached to the terms cited and it is also possible, under these circumstances to obtain an index of *correct belief* (Chapter 1). The reality or the relevance of the self-organisation is guaranteed by any scheme that renders any decrease in H, which is used as a *control signal* by the regulator, contingent upon either an increase in correct belief or a constancy of correct belief.

In practice, schemes of this kind work very effectively. They have a much wider field of tutorial application than the domain of structured skills discussed up to this point.

9 Descriptions of Processes

A description, of any sort, entails a language in which the description is expressed and which can be interpreted by someone or something. For instance, one can describe, in natural language, the events going on in one's room or one's mind, in the street, in an aquarium or in a motor car's engine. The widespread idea that a static skeleton of things must first be described (e.g. the pistons of the engine, its cylinders, crank shaft, and so on), and that events become obtrusive later as happenings to *things* is a misconception.[1] Often this approach is convenient and logicians have achieved formal elegance by adopting it. But there is nothing in man or nature to enforce this world view and (as particle physicists, amongst others, have found) it is sometimes better to notice events first and to build things, either real or fictional, around them.

The outstanding distinction between a static description and the description of a process lies in the capabilities assumed to exist in a user (recall, from Chapter 1, that there is only *potential* information in a description, i.e. literally it *has* a pattern, it does not *have* an information). Both static patterns and processes may represent classes ordered by the writing and connection rules of a language or functions and relations. For example, a table specifies a relation and abstract automata may be represented by a table (for instance, *Ru* of section 1.2 in Chapter 2). But if a pattern is depicted then the processor that uses it must be equipped with an order of execution or a specially operated clock (a seeking-out program) that reconstitutes the table in its own storage and refers to it. If a process is represented (the specification *Rul* in contrast to the specification *Ru*, for example), then order of operation is carried as part of the pattern; it prescribes the process to be executed directly and it may prescribe or permit many execution orders.

1. Little or nothing is said, in this volume, about the description of static objects and patterns; still less concerning the important topic of description, *efficiency*. Without an *efficient* description, most problem solving operations and control operations are utterly impracticable. The reader is referred to Banerji (1970, 1971) for a comprehensive as well as elegant discussion of the field and the most lucid available account of what a description *is*. Perhaps the best technical accounts of pattern recognition, learning, and search are found in Ivahnenko and Lappa (1967) and Fu (1968).

Phrased differently, any function represented in a description may, or may not, have an imperative tag and any relation may, or may not, have a subjunctive/imperative tag. For a process description, functions are to be executed as verbs in the imperative or indicative mood; relations are to be executed as verbs in the subjunctive mood; a pattern description is not to be executed at all but referred to by a process undergoing execution. A process which may, of course, assemble it into a program, just as a reproductive automaton does in accepting a tape description and producing a machine.

1 Process Description

A process description can be inscribed (as a picture on paper, as printed words, or in the storage position of a computing machine). It is a *latent* or *potential* process and not the process itself.

Further, if you or the machine have a time sense with respect to past and future there is a sense (*this* sense) in which the process description is an account of *what has happened.* Of course, being equipped with a time sense you or the machine may describe *that* also and by qualifying a process description of *what was* into *what will be* transmute it into a *plan.*

The essential intermediary, required in order to use a plan, is a command and question language; minimally a language that is interpreted and capable of encompassing not only statements but imperatives and interrogations.

1.1 Suppose, you *intend* to use a plan. If you use it yourself, that plan is a description of your intention, but you might also intend it as a formula for action by some other agent acting in the capacity of an executive. If so, you address the transmuted process-description to the agent as a *prescription.* The agent/addressee may be a person, a group, a machine or an organisation; provided only that the agent is able to interpret the language in which the prescription is written and either to *have* the capability to execute the plan or to be in a position to *obtain* this capability. (In the special case mentioned, you first of all address the plan to yourself.) Equally well, you might use the inscribed process-description, without transmutation, as a *permission giving* structure addressed to an agent or recipient, and allow him to do certain actions or to make certain 'grammatical' utterances in the language (the process description is used as a grammar but this does not mean a syntactic grammar only; it is a semantic and pragmatic 'grammar').

1.2 The prescription can be represented in various ways. The preferred canonical representation is a command or a set of commands. This usage is uncommonly liberal and requires some qualification. The connotation of 'command' is a statement in the pertinent language of the type.

(a) Addressee ! Do P/Given X

or (b) Addressee ! Bring about Y/Given X

Where P is a process, and X is a precondition (the command to do P being obeyed by the addressee if and only if X holds); Y is a further condition often called the goal condition. In fact (a) and (b) are equivalent in so far as doing P brings about some Y and achieving Y involves the execution of some P.

Henceforward, we shall treat (a) and (b) as equivalent with a bias towards (b). The form (a) is a specific *instruction*, a very special type of command, like 'add' or 'move on' and it says nothing about the result of obeying the command. It is characteristic of run-of-the-mill, particular-to-general, programming languages. In contrast, type (b) invokes the achievement of Y (given X) by any means at the addressees disposal, and it presupposes the existence or constructability of P for this purpose. Type (b) is characteristic of general-to-particular programming languages like PLANNER or MICRO-PLANNER (Hewitt, 1969, 1971; Winograd, 1972).

1.3 By comprehending the complement of the precondition, the command form becomes the conditional statement with fixed addressee

'If X then Y if not X then nothing'

which is readily extended using a command disjunction of Rescher (1966) (one, and one only, of Y_1 or Y_2 depending upon data to be supplied) to yield the standard conditional form

(c) 'If X then Y_1: if not X then Y_2' $\triangleq$ 'If X then Y_1, else Y_2'

or in general to yield the complex conditionals

(d) 'If X_1 then Y_1 else if X_2 then Y_2 else . . . if X_n then Y_n*'

However, the command form is open to fuzzy interpretation, which is essential in the developments of the next volume, as are any of Zadeh's (1973) fuzzy conditionals.

1.4 New commands may be composed and decomposed (ultimately to instructions) using either Rescher's (1966) calculus or a mild extension of it in the coverage of which command graphs are introduced as well as Rescher's command chains. In this sense any prescription for a process is representable as a command, or a set of commands, but certain variants are very important.

1. A command (large or small) may, as a special case, be *unconditional* (⟨Addressee ! Do P/for any precondition⟩ or ⟨Addressee ! Do P/One *state*⟩ and *that* state holds).

2. If the relation between *X* and *Y* is a *function* then the command is interpretable only as a direct *imperative*. It gives a value in the range of the function for a value in its domain.

3. If the relation between *X* and *Y* is not a function, then the command is only interpretable as a subjunctive imperative.

4. A command that is obeyed is a *goal*, in the sense of an *intention*. It may be that a *goal condition* (*Y*) is itself a goal (this is always true, for example, in my theory of conversation and learning).

5. A command to make an utterance of a given form in the language, *as though* a condition held, is a *question*.

6. A question may demand an explanation; if so, the explanation is a description of the execution of the original command (to bring about a goal condition) in a context that is generally left free.

Given (4) above, the execution of a command *models* a relation and the *explanation* is the description of a certain modelling operation (as required for example, by Loefgren, 1972).

7. Either the command or its question variant may be qualified; the qualification restricts the context in which an explanation is given.

8. If the linguistic form of the answer to a question is a statement satisfying an exclusive disjunct, it is a *whether* or multiple-choice question; if several such statements are permitted (given a list of disjuncts) or if there is an open disjunction (asking for a *constructed* response in reply), then the question is a *which* question. The logic of the syntax of *whether* questions and *which* questions is developed by Harrah (1966). His recent treatment of more general questions 'Erotetic Logic' (Harrah, 1973) differs from the present approach but is compatible with it.

1.5 In overview, a process description may be converted into a plan. Again, it may be addressed to an executive agent; if so the process may be *prescribed* or *permitted*. In either case its execution, or possible execution, may be imperative/subjunctive, imperative or interrogative. The verbal summary is shown as a graphic condensation in the chart.

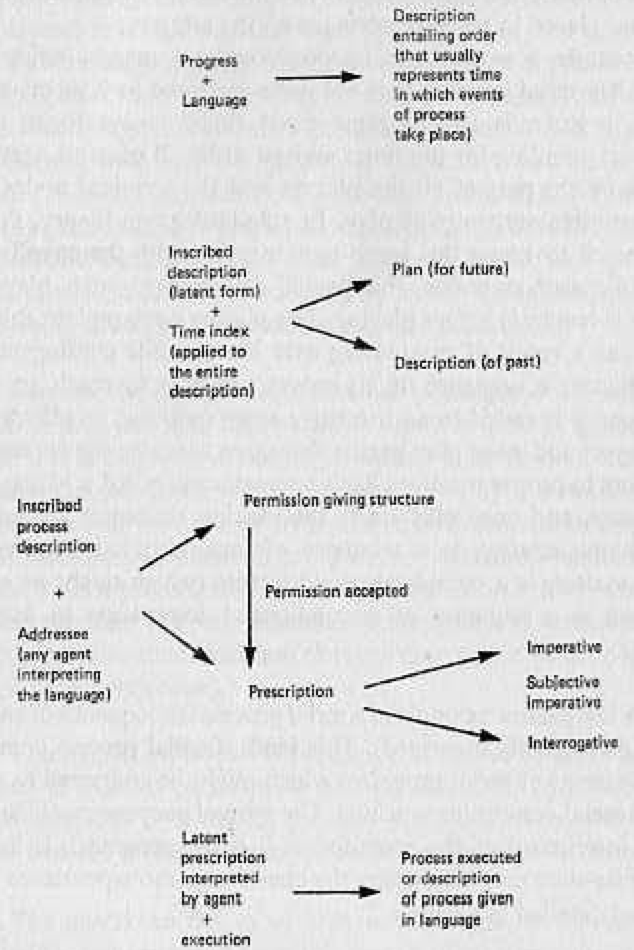
In this picture it is also supposed that the agent *has* the wherewithal to execute the commands, but all we require is that an agent should either *have* the wherewithal or should be able to obtain or construct it. Hence, two further distinctions are needed, namely:

(α) If an agent has the wherewithal and obeys a command, then it is a command to *do*.

(β) If an agent does not have the wherewithal and obeys a command, the agent may or may not interpret it as a command to *learn* to do. Certain agents, notably human beings, act like this by design.

(γ) A command to do certain operations that bring about the wherewithal for obeying (other, as a rule unrelated, commands) is a command to learn.

As before, the command may be an entire prescription or any part of it.



2 Execution Order

Any process whatever entails the notion of execution *order* (that is, what distinguishes a process description from a static description). The following kinds of process order, though not exhaustive, are particularly important; namely, serial processes and non-serial processes.

2.1 A serial prescription is a chain or list of commands, conditional or unconditional. The process obtained on executing a serial prescription, i.e.

a *serial process*, is exhibited using one token which resides in one, and only one, position at once (a 'locus of control' or the command currently undergoing execution). Thus a record of the conditions brought about is a list or sequence of items that can be tagged, like the successive positions of the one token, and placed in correspondence with the integers.

For example, a sequence of unconditional commands (either to *do* or to *learn*) is a serial process. It is not quite so trivial as it seems to be. For example, a game is usually represented (in extensive form) as a tree, with nodes standing for the states arrived at by all possible combinations of moves on the part of all the players and the terminal nodes standing for the possible outcomes of play. In standard game theory, the players are supposed to know this game tree, together with the payoff function; that is (for each outcome, the payoff delivered to each player *if* that outcome is reached) before playing. The players contemplate this description and, as a result of ruminating over the possible contingencies, each player chooses a sequence of *his* moves sufficient to reach an outcome. This sequence is stated to an executive agent (without revealing it to any other player) and once play begins the agent executes the sequences of all the players in proper rotation. Such a sequence is called a *playing strategy* in the game, and once play starts the playing strategies are immutable, i.e. a playing strategy is a sequence of unconditional commands; so, for that matter, is a completely rigid lecture (which might be charitably interpreted as a sequence of unconditional commands to *learn* rather than *do*).

2.2 A less narrow example of a serial process is a sequence of conditional commands (possibly questions). This kind of serial process comprehends the whole range of serial programs which might be addressed to a (general purpose) serial-computing machine. The general purpose machine, together with an inscription of the prescription (i.e. the program), is isomorphic with a finite-state machine, under the one-to-one correspondence '⇔' given below and detailed in Chapter 2.

Initial state, input state sequence	⇔ Fixed programme data
Internal states	⇔ Values of programme variables
Output states	⇔ Values of output variables
Steps in execution	⇔ Steps of sequencing shift register

2.3 If the program data is not fixed and is delivered, in an appropriately synchronised manner, during execution, then the finite-state machine has a free input and the computation need not be fully determined. In general, however, a fixed input is assumed and, if so, the program is equivalent, in the abstract, to a finite automaton.

This equivalence is usefully referred to the game formulation. Whereas standard game theory represents playing strategies as sequences of unconditional commands, the development of game theory by Banerji (1970) encompasses the notion of a strategy that is conditional upon the moves of other players. It is still true that once playing begins, the players (having chosen their strategies) are no longer in control. But they have specified, as strategies, a set of finite-state machines (serial programmes) that are executed by an executive agent. The entire game is represented by a directed graph (not usually a *tree*) and the playing is the action of a large finite-state machine. The individual, strategic, finite-state machines may thus be cognisant of any game states the players have anticipated before the play and may utilise *feedback* data. In fact the Banerji game is derived as a special case of *feedback* control (a control operation *is* a playing strategy) whereas standard game-playing strategies mediate feed-forward control only.

2.4 Suppose the conditional commands in a sequence are subjunctive in form. If it happens that the executive agent is a *serial* general-purpose machine (i.e. it can only deal with one locus of control at once) then the permissions, possibilities, or enablements given by the subjunctive interpretation can only be resolved by (more or less complex and precondition biassed) random selections. Hence the serial execution of the subjunctive imperatives in a prescription is biassed-random and the general-purpose machine, together with an inscribed prescription is isomorphic with a probabilistic finite-state machine (of a type determined by the complexity of the biassing dependencies).

2.5 Finally, an *algorithm* (Markov, 1961 is the original and clearest reference) is a serial prescription and its execution by a serial general-purpose machine is a serial process; the general-purpose computer, together with the inscribed prescription is isomorphic to a Turing machine.

2.6 The moral, so far, is as follows. Prescriptions contain a certain order; in the cases examined up to this point it either *is* or *it can be interpreted as* a serial order. Any agent (e.g. a processor) that executes the prescription also has an ordering principle; for example, that it executes one command at once (the command on which our single token is placed at that instant). Phrased alternatively, the engine has one locus of control at once. Moreover, its *state* is always well specified. There are many reasons why an executive agent may be unable to obey a prescription; for instance, because it cannot interpret the language in which the prescription is written. A consideration that overrides all of these minutiae is that the order in the prescription shall correspond to the ordering principle in the executive.

2.7 The other category of processes, non-serial processes, is a mixed bag. A non-serial process contains junctures at which the order of execution is irrelevant. If it happens that simultaneity involves no *conflict* (that is, the execution of one command does not preclude the execution of another) then execution of the process can be represented by placing several tokens on positions that stand for commands and moving these tokens *independently* as the commands are obeyed. Holt (1968) refers to non-serial processes of this type as 'occurence systems' which is the best canonical form for representing 'parallel computation' of one kind or another. The prescription for a non-serial process of *this* type can be accepted by any executive agent that comprehends the language; provided that it is made up from sufficient small processors, which act independently but are fully synchronised. Because of that, the combination of processor and inscribed prescription is characterised by as many loci of control as there are marker tokens but it has a well defined state at any instant of execution. Further, the loci of control do not interact though the supervisory synchronising mechanism will combine the separately computed data at instants determined by the prescription. The performance of such systems has a general interest; it represents reality quite well and parallel processing saves time and occasionally space. But nothing very new emerges whilst the control loci remain independent. The most penetrating analysis of parallel systems is given by Minsky and Papert (1969), in the context of a special class of pattern recognising machines (the more elementary versions of Rosenblett's 1961 'Perceptrons', Chapter 2, Fig. 196).

2.8 This type of non-serial process devoid of conflict, has an execution *tree* rather than an execution *chain*. Several tokens could be moved down the tree to represent simultaneous command executions. For example, several property tests can be evaluated at once and variables may be assigned several values, thereafter isolated and separately treated. Perhaps the most significant difference between the serial and the (restricted) non-serial process is that a subjunctive imperative command may lead to several results (studied as distinct, however, unless there is a directive to coalesce them in a specific way). Whereas, in serial execution, these commands are necessarily resolved as one value by a metalinguistic ranking or maximum finding operation (Shimura, 1973; Zadeh, 1973) or to yield one alternative (by a biassed random selection).

Non-deterministic programmes (in the sense of Manna, 1970) are prescriptions of this type. So are the fuzzy algorithms of Zadeh (1973). If executed by a serial device, the conditionals lead to a numerical solution signifying grade of membership in a fuzzy set; whereas, if executed in parallel the fuzzy members of the fuzzy set, or its extremum, form a non-numericised solution).

More interesting possibilities appear if a suitable non-serial prescription is addressed to an executive agent having several loci of control that are not necessarily independent. These loci are most conveniently regarded as distinct but initially asynchronous processors that can be drawn into synchronisation in so far as the execution of a command by one of them interacts with the command being executed by another. Since the processors are not synchronous, there is not always a *state* of the entire system. Moreover, the cooperative and synchronising interactions between loci of control amount to (non-trivial) information transfer. Subjunctive imperative commands, in particular, give rise to this type of interaction (the simultaneous execution of fuzzy conditionals, where the fuzzy solution from one control locus is the input to a computation at another locus). Let us call the process obtained by executing this type of non-serial prescription a *concurrent* process, as was the case for machine processes of Chapter 2.

3 Some Defects

The following concurrent process descriptions all have certain defects (to be discussed in a further volume) as they are tailored to suit very different applications. Combinatonic networks (Barralt-Torrijos and Chiarviglio, 1971; Chiarviglio, Poore and Barralt-Torrijos, 1971) derived from a combinator logic approach to computation, represent both parallel, serial and concurrent processes. Petri nets (Petri, 1965; Holt, 1968) are slightly more general but correspondingly intractable. Dienes (1972) has discussed the matter.

One concealed (some say 'obvious') assumption underlies all of these prescriptions and processes. Whether serial, *parallel*, or *concurrent*, the process has a *beginning* and an *end*; at these points any processor does have a well defined *state*.

Personally, I do not find it at all obvious that all processes *do* have beginnings and ends. 'Process' could be defined as *having* this property, of course, but many applications of the term 'process' to real activities would be prohibited as a result of such a definition. I shall try to justify this point as well as illustrating the other distinctions by reference to executive agents less restricted than most general purpose computers; namely, human beings who compute with complex- and variable-order principles.

10 Processes and Prescriptions for Action and Learning

This chapter describes some problems of human performance and learning from a process-oriented viewpoint. The concepts stem, in part, from studies of industrial and commercial organisations and from studies of the teaching process in the areas of programmed learning and computer assisted instruction.

Section 1 deals with prescriptive, descriptive and permissive structures made up from commands to *do*; section 2 with rather narrow prescriptive structures made up from commands to *learn*, and section 3 with a detailed case history. The theme is continued into the next chapter on strategy in which it is possible to introduce permissive and descriptive structures and to voice a need for very general representations of the *learning* process.

1 Process Instituted by Commands to *Do*

In various branches of industry, processes such as clerical and inspection routines or even the operation of a whole department are represented by flow charts, decision tables, and condition charts (including critical path and PERT meshes). These instruments are used descriptively to give an account (e.g. in work study) of how people do jobs; prescriptively, to induce or monitor performance and, in the case of the meshes, permissively. (A flow chart depicts one process; variations are induced by external conditions only).

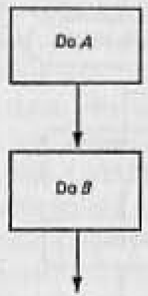
It is provident to concentrate upon the prescriptive application as the most frequent and successful one and to comment, where necessary, upon description and permission-giving. In this case, the addressee (the executive agent) is a human being and the commands (including questions) are humanly interpretable. Throughout this section all commands are commands to *do*, but it is possible that an addressee who is unable to obey these commands will interpret some of them as commands to learn how to do.

There is no evidence of a strictly behaviouristic identification of the type which is very common (section 2) in the context of commands to *learn*. The idea that a *prescription* for a process might be a series of semantically-

neutral stimuli evoking (causally-related) responses has little currency and is pretty obviously untenable against the background of a process governed by rules and regulations.

1.1 Instruction Sequences Any linear arrangement of unconditional commands (such as a check-list or a rudimentary manual) may act as a process prescription, on the assumption that the addressee can interpret the commands and that he is able, as well, to interpret the order in which they are written (that is, a serial order; man is asked to use his brain as a serial processor). Such a prescription can be executed but is only of value in very static situations, where its execution is likely to produce the desired result only because the environment or input data is *not* variable. If the prescription is regarded as a prescription for control and the executive agent as a *controller*, then he is a *feedforward* controller and is liable to the frailities (inability to remedy unappropriate actions) of all such devices.

1.2 Flow-Chart Representations A series of unconditional commands could be represented as a 'flow-chart'; it would consist in a connected series of operation or assignment boxes



and so on. Generally, and non-trivially, flow-charts are used to represent conditional processes and to admit of feedback control as well as feed-forward control. Consequently, the notation includes a condition test symbol (usually shown as '▭') and such 'test boxes' are combined freely with unconditional operation boxes. Some caution is needed in so far as the flow-chart may either represent a process (like an unmonitored serial computer program) in which all of the data is available within the system or it may represent interaction with other systems. For example, some operations may be searches or enquiries that are not guaranteed to terminate; some of the conditions tested may depend upon the answer to genuine questions (rather than look up operations within a fixed data structure). The crucial point is not, of course, where the operational data is located but whether it is available to the system (i.e. the operation is do-able). You cannot drive to Manchester from the garage at your office starting at once (on the face of it a reasonable command), if there is no motor car in the garage at this moment, nor can you check stock prices from the telex, if the line is out of action. If the availability of motor cars, telex data, or whatever, is aleatory then the system receives an input. More realistically, 'if the best restaurant in town is full tonight, book him a table at his hotel; otherwise book a table for five at the best restaurant' generally does demand an enquiry (a non-trivial one; 'which is the best restaurant?' and 'is there such a thing?'). So far as prescribing a process is concerned, the existence of genuine input as against the existence of guaranteed data-values is a matter of what is assumed (the *use* of a prescription or the manufacture of a *description* is a very different matter). Unless the contrary is stated, we shall assume that the objects used to perform operations (motor cars, slide-rules,

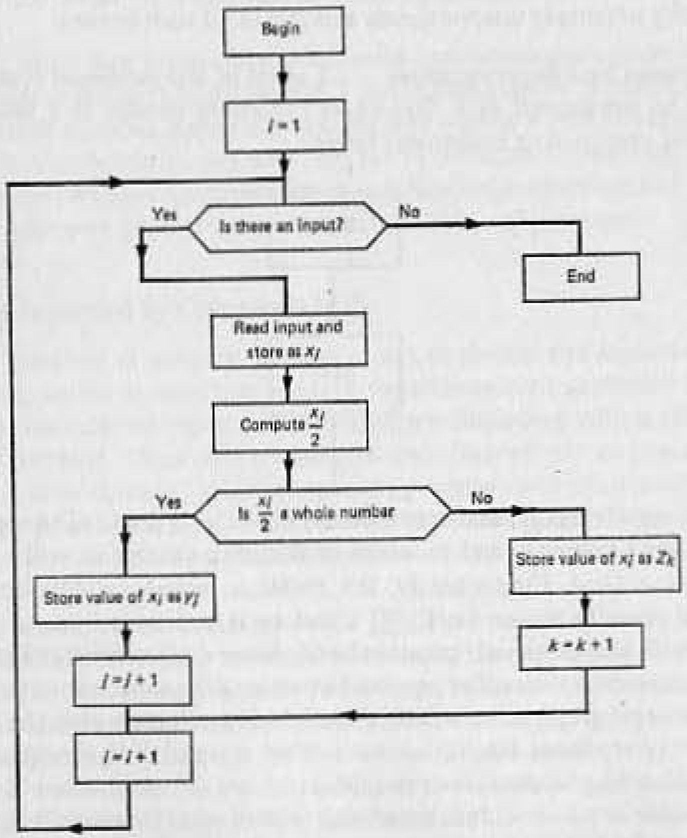


Figure 85 A performance algorithm flow-chart for the process of taking a list (x_i) of numbers and producing a list (y_j) of even numbers and a list (Z_k) of odd numbers.

aeroplanes) are available, that there are no genuine inputs, and that for any condition test the posited conditions exist, so that a value can be determined. In other words, we are making the temporary and very dubious assumptions (a) that the boundaries of a system are determined (whatever that means; it may mean many things depending upon how the 'system' is dissected out from the flux of events as '*the* process') and (b) the system is a *closed* system (of these (b) is meaningless unless (a) is affirmed and the affirmation or denial of (a) is actually a very subtle issue).

With these caveats in mind it is useful to distinguish several varieties of flow-charted process; performance algorithms, decision flow-charts (condition test charts) and subroutine charts.

A *performance algorithm* flow-chart (for example, Lewis, Horabin and Cane, 1967; Lewis, 1970) consists in conditional test boxes and operation boxes linked by directed arcs. It has a beginning and an end (Fig. 85). It may contain loops (iterated conditional operations). Execution of a task can be represented by placing *one* marker on the beginning and moving it along the directed arcs. On reaching an operation box, the stipulated operation is carried out. It may be any imperative (an assignment statement like 'give a variable a freshly computed value', or a physical action like 'place the form on a pile'). On reaching a test box, the test is carried out and an appropriate exit is made. Strictly, the tests are confined to evaluating variables that stand for predicates with the values 'true' and 'false' (or '1' and '0') and to tests for equality on numbers. But these tests may be combined in a single test box by the logical connectives 'and', 'not', 'or', in which case the test box is called a limited entry decision table. Further, the value □ (no value assigned) is admitted with respect to undetermined exits.

The addressee is credited with the ability to interpret and carry out all the instructions (both tests and operations) that appear in the algorithm.

A decision table flow-chart has the same connection rules and the same 'one *marker* at once' restriction as the algorithm. But the test boxes may be Decision Tables (Fig. 86) rather than limited entry decision tables. In these boxes, various conditions can be tested, for example, inequalities between numerically valued variables like 'greater than' or 'less than' or 'greater than or equal to' or 'more than 100 and less than 250'. These conditions indicate the holding or not holding of relations. Though the relations cited so far hold between two variables it is possible to extend the notation in order to encompass many-variable relations such as '$A=B=C=D=E=F$' ('the books are balanced', or 'the aircraft is symmetrically loaded'). Once again the addressee is credited with the ability to interpret and carry out all of the instructions.

Several difficulties arise in connection with the extended form of decision chart. Under the broader interpretation of a condition it is not clear what

Determination of Tax Liability	*Column or 'Rule' Number* 1	2	3	4	5	6
Q1 Selling Price greater than Market Value?	Y	Y	Y	N	N	N
Q2 Market value greater than Cost price?	Y	N	N	Y	Y	N
Q3 Selling Price greater than Cost price?		Y	N	Y	N	
D1 Tax charged on Selling Price less the Cost Price, less Expenses	X					
D2 Tax charged on Selling Price less the Cost Price, less Expenses		X				
D3 No Tax either charged or allowed			X	X		
D4 Tax allowed on Cost Price less the Selling Price, plus Expenses					X	
D5 Tax allowed on Market Value less the Selling Price, plus Expenses						X

Figure 86 A decision logic table for determining tax liability (after Lewis, 1970).

can be compared with what. Whereas the addressee or user may be expected to compare *numbers*, does he have the ability to recognise more obtrusive properties; either abstract properties like *similarity* or concrete properties like shape and colour that are used in many ordering schemes?

The dilemma is usually resolved (or simply inched around) by appeal to 'well learned skills'. The solution is legitimate enough if training in these skills is provided before the user is asked to interpret the chart; failing that, he is almost bound to interpret any command to *do* (not just the command associated with the decision table) as a command to learn. If so, the interpretation of the entire chart becomes equivocal; various operations coalesce into 'more or less well learned skills' and the chart becomes virtually meaningless (it might as well be represented as a big decision table).

Suppose, for one reason or another, that a process *is* serially executed. It is thus possible to represent it as a flow-chart. The chart may not only be complex but repetitive. Like any other serial programme, it usually contains parts (i.e. subroutines) that represent the same kind of computation carried out in different contexts or with different data. A 'subroutine chart' is a representation of this structure. It has elements that may be either simple conditional commands or subroutines made up from many; the only

restriction is that each subroutine has been delineated, before execution, *in extenso* (as a segment of simple conditional commands, duly surrounded by a marker to indicate its integrity and named or labelled). Such a chart is shown in Fig. 87 and it can be employed practically using instruments for accessing and assigning parameters to representations of the subroutines. Plate 8 shows one of them.

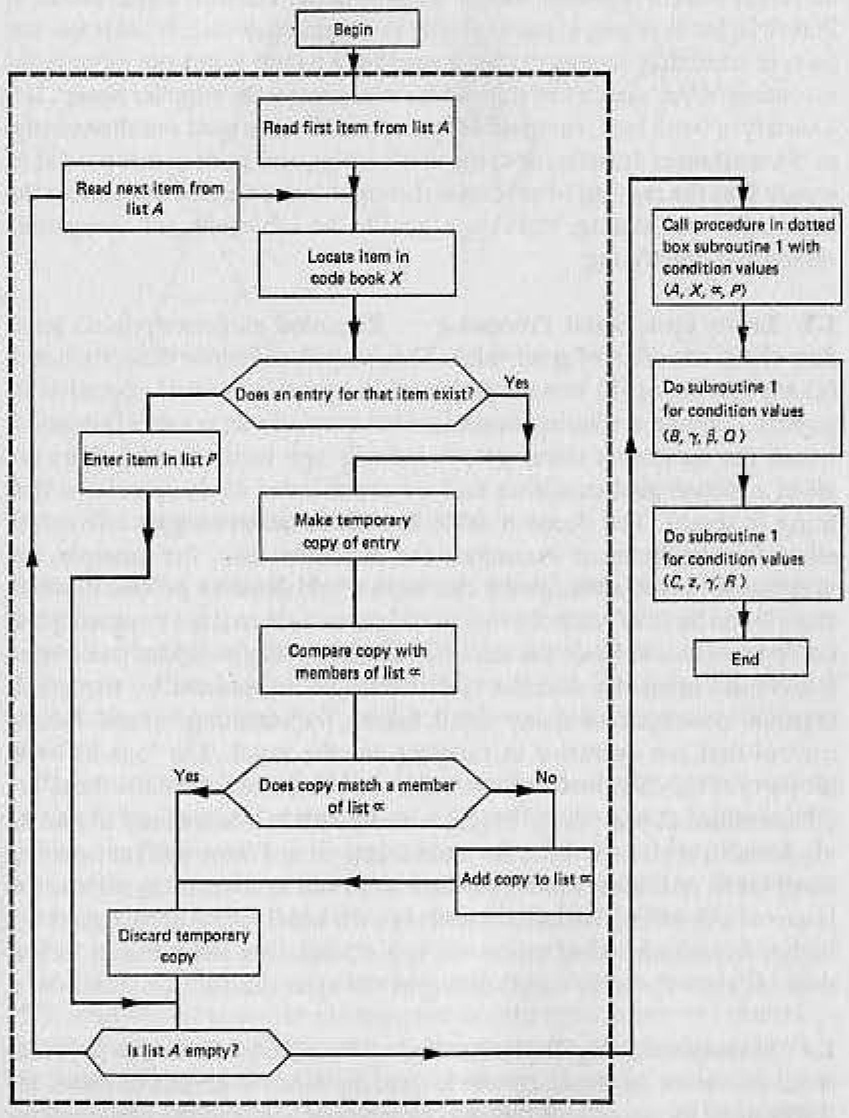


Figure 87 Example of a subroutine chart.

Man's habit of interpreting commands to *do* as commands to *learn* is contingent upon naming and describing what is to be done, i.e. upon a goal condition being specified. But the naming of a subroutine goes together with a contextual statement of what it achieves or satisfies. Consequently, subroutines (especially the simpler ones like index searches) become well learned skills. That is harmless and may be desirable provided the chart is no longer held to represent what is actually done. The instrument shown in Plate 8 is, for example, in use explicitly as a training device. What is learned (say, in searching documents for a supplier's name) is not the more rapid execution of the 'search for' subroutine with parameter 'supplier name'. It is a variety of (with luck, compatible) processes that are used simultaneously, as circumstances demand, or at the user's whim, and some care is needed to ensure that the rag-bag of processes (brought into existence by naming the subroutine and stating, thereby, a goal to be achieved), are compatible rather than interfering.

1.3 Limits upon Serial Processing Regarded as prescriptions, serial flow-charts are often of great value. They may also *describe* the activities of (a) any novice, or (b) some experienced but peculiarly serial operative. In general, however, the human brain does not normally act as a serial processor (recall the comments about people *learning* new methods when they are given a named goal condition and are commanded to *do* operations that bring it about). The decision table flow-chart constitutes an attempt to allow for simultaneous execution; the condition tests, for example, are unordered. Hence, although the chart as a whole depicts a process in which there is one locus of control at once (which can be simulated by moving one token from box to box), the activities actually going on whilst this (main) token rests upon the decision table would be represented by the simultaneous movement of many small tokens (representing several loci of control that are operative in carrying out the tests). The 'one at once' property of the chart itself is preserved by insisting that the 'small tokens' are all assembled at one point, 'begin', when the main token comes to rest on the decision table box. Also, the main token cannot move until at least one small token resides on a point labelled 'end', and as soon as the main token is moved away from the decision table box the small tokens are assigned to a limbo, from which they return to 'begin', and that returning to 'begin' depends upon the main token coming to rest upon the decision table box.

1.4 Permission-Giving Representation of Non-Serial Processes Non-serial processes are generally represented by directed graphs in which the nodes stand for conditions that may or may not hold and the arcs stand for relations of precedence between the conditions and their holdings. A con-

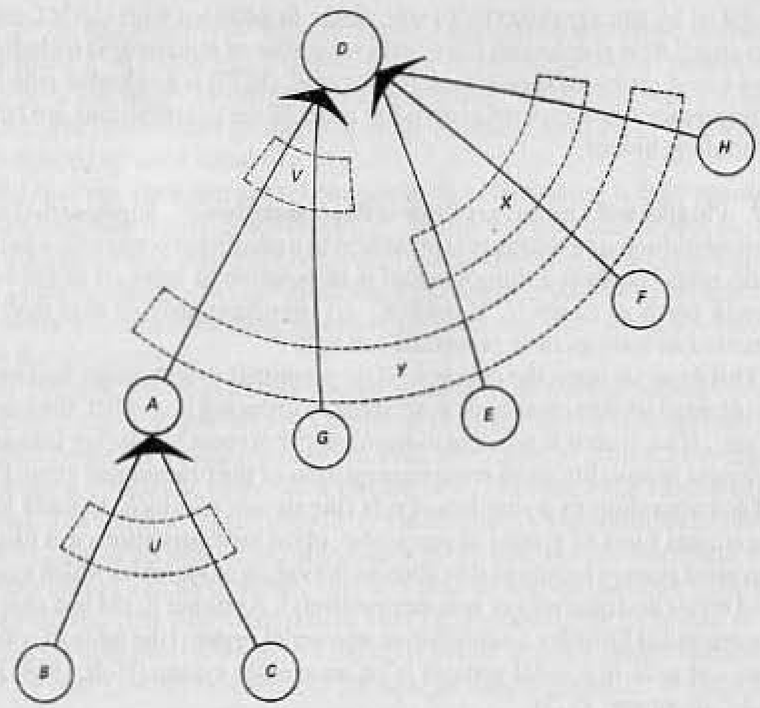


Figure 88 Fragment of a precedence network.

dition is itself a relation (not a state). A typical part of such a graph or *precedence network* is shown in Fig. 88 where there are two notations; condition *A* may hold if and only if conditions *B* and *C* both hold; condition *D* may hold if and only if either conditions *E* and *F* and *H* all hold, or if conditions *A* and *G* both hold, or if both of these complex preconditions are satisfied.

The precedence network is clearly permissive as all of the transitions are prefaced by 'may'. The operations that bring about condition-holding (which may be executed if appropriate commands are issued and the command preconditions hold) are usually represented as labels over clusters of arcs in the graphs (for example *U*, *V*, *X*, *Y*, in Fig. 88).

Pictures of a similar kind are familiar enough. For example, they are used in critical path analysis (for representing production processes) and in PERT networks (where the durations of a critical path chart are refined into expected durations and other statistical parameters, such as the variance or the means of expected value). In fact, these are all rather specialised forms of process representation (so is Fig. 88 which would be known in computational circles as an and/or network). The whole subject is comprehensively reviewed by Elmagrahby (1970) who deals, in addition with GANT charts and signal flow graphs and introduces a number of innovations including a very useful *restricted conditional*. Hobbs *et al.* (1970) is a valuable collation of papers for the computer expert, but most of the contributions are rather technology bound.

1.5 Parallel and Concurrent Process Representation Suppose that such a permission-giving structure is presented to a parallel processor (the human brain might be such a thing), would it be possible to issue all of the commands (such as to do U, V, and X, Y,) simultaneously so that they are executed as soon as their preconditions hold?

This depends upon the network. If no permitted subprocesses that might be executed simultaneously or in an arbitary order are in conflict, the answer is 'yes'. If so, then it is possible to simulate the process by placing tokens on sufficient preconditions to ensure completion of the process and move these tokens according to a simulation rule (the nature of which depends upon the chosen form of graph; in particular, upon how conditions are used to represent storage locations that contain the values of variables which may be read either destructively or non-destructively). As noted in the last chapter, the canonical form for a conflict-free non-serial system (the issue of conflict does not arise in a serial system) is an occurrence system (Holt, 1968; Holt and Commoner, 1972).

There is also a sense in which viable systems must be closed, i.e. that 'alien tokens' do not intrude, except as transients, into a simulation. The tacit suggestion is that 'a process' or 'a system' is identified by the token type (colour, perhaps) used to simulate its execution. For a process with input, some preconditions are determined by 'other' processes with different types of token and the systems must not get mixed up. Though there is a kernel of truth in this notion it is replaced, in the next volume, by a much less artificial identification of a process's individuality based upon essentially immunological criteria abstracted from biology. The kind of 'other recognition' (hence, in a limited way, 'self recognition', which is a ubiquitous characteristic of any biological immune response process) appears to be essential to the integrity of all processes and systems (it is the key to what is meant by 'begin' and 'end' as well).

1.6 Resolution Suppose there is conflict. How would it arise?

(a) It may be due to any process that looses specificity or makes data ambiguous (the same thing). If the computation is fully specified in advance, this kind of conflict may be avoided by a precedence network that prohibits it. For example, in computing the value of

$$(a + b) \times (c + d)$$

the '+' of (a,b) and the '+' of (c,d) may be computed in any order, provided the results $e = a + b$ and $f = c + d$ are stored for later reference (if the storage is available) or the '+' operations might be carried out simultaneously. But both values (e and f) must be available before the '×' of (e,f) is computed to yield value $= e \times f$.

But it is not always true that the required computation is fully specified (in the requisite sense) in advance (e.g. any open-ended search process).

(b) Conflict may also arise because of a command to perform incompatible physical operations; for example, you cannot brake and accelerate at the same moment (assuming you have two feet and one of them is resting over the clutch pedal).

(c) Perhaps the commonest source of conflict is storage allocation (as mooted in the example of (a)). Computer users are apt to take storage for granted (machine designers have no such illusions). If man carries out a computing operation, typically a clerical one, then the issue of storage is crucial. The storage is a resource. Some storage is located in external media such as files, cards and so on. Some storage is located in his own brain and conflict occurs due to processes that can find no storage position for computed values, that overwrite values, that fail to create free locations or to extend the (external) storage to a requisite extent before a task is started.

(d) Finally, conflict can occur because of communication. Though the fact was not stressed, all of the examples so far cited are due to unwanted communication between loci of control-executing (incompatible) operations. If these loci are conceived as different people (in a team, for instance) there is no difficulty in seeing the point. But the loci of control may just as well coexist in the same one brain processor.

The situation is greatly complicated as soon as there is a genuine interaction between *our* process and some other process; whenever *we* are in genuine ignorance about what the other process will do. For example, the sanitary prohibitions of (a) cannot be realised.

1.7 Communication and Cooperation Communication is a mixed blessing and the last couple of paragraphs emphasised its negative aspect. On the other side of the coin, communication between loci of control (they could be in separate people but need not be) gives rise to cooperative effects which enable otherwise impossible computations. Similarly, a mix of internal and external communication is responsible for the 'redundancy of potential command' of McCulloch (1965). Of various potentially dominant loci of control, the one that is currently best informed about the prevailing situation becomes immediately dominant. In all cases, the communication depends upon a partial or complete synchronisation between the loci of control.

In a very fundamental sense, this synchronisation is information transfer, and the communication is about that information transfer (or is a symptom of it). For later reference I have shown in Fig. 89 Holt's most elegant example of the distinction between 'information transfer' in this sense (far removed from selective information theory) and 'control'. The caption is self explanatory.

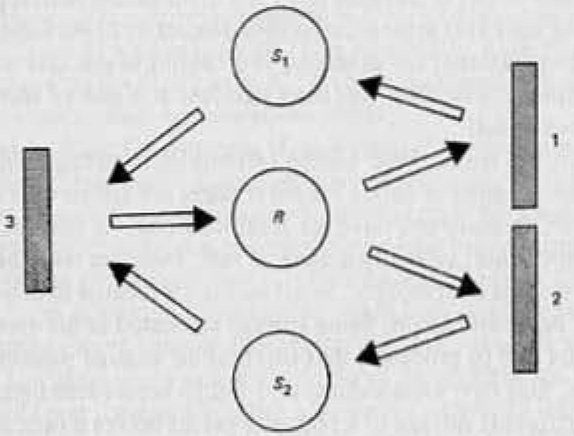


Figure 89 Simplest case of a Petri net showing conflict (after Holt *in* Bateson, 1972): circles, system conditions; hatched bars, transitions. Starting at condition *R*, two different transitions might occur (1 or 2). A determination of which transition occurs requires *information transfer* from *outside* the system.

1.8 Some Comments on Human Processing The possibility of conflict occurs with the possibility of cooperation; competition is a special case of conflict. Processes that entail all of these possibilities are concurrent (partial synchronisation of otherwise asynchronous loci of control is permitted).

It looks as though the human brain, taken as a processor, computes concurrently; that is, even if presented with serial commands to *do* these are interpreted at the least provocation (perhaps always) as commands to *learn*, and, in fact, how to learn how to perform the same task as executing a process.

As noted in the last chapter, the canonical representation for concurrent systems is a Petri net. The difficulty is that token assignment simulation rules are hard to formulate for a general case, though in special cases there is no great difficulty. On the other hand, for special cases that are tractable, it is often just as convenient to use the simpler notation of Fig. 88 in a slightly augmented form. If a human process-prescription is to have descriptive pretentions (or even wide utility as a prescription) then a

representation of this type is mandatory as a minimum. An example is shown in Plate 9. The networks are complex because human processes are complex. The representations are the least able to furnish a workable approximation to reality.

2 Tutorial Programmes

Training routines are typically organised as temporal sequences or programmes which often contain conditional operations and test points. The teaching of academic subject matter is sometimes represented in the same manner. The executive agent may be a teacher, a student himself, or a machine; the last case is emphasised because the limitations are more clearly expressed, not because machine administration is 'best' or even 'most frequently employed'. Unlike task performance specifications, the instructions in the tutorial programme are intended to bring about changes in a pupil or student. They may, incidentally, evoke performances (either exercises or tests) but they are not so much directives to *do* as directives to *learn*. Tutorial programmes are used as prescriptions and (locally only) as permission-giving structures that enable certain autonomous activities and disallow others. They are not used to *describe* learning though any defensible programme is based upon a descriptive (and, if possible predictive) model for how students do learn. This model justifies the programme design.

The minimal capabilities of the executive agent depend, first of all, upon the programme type.

The simplest (serial and *linear*) tutorial programmes are simply chains of instructions. Logically they have the calibre of *game-playing strategies*. Whoever designed the programme had in mind a 'game' in which his executive agent was one player and a member of the student population the other player. Before play, the designer set out all possible joint moves leading to an outcome and selected one sequence as the playing strategy to be adopted by his executive agent (which need thus be no more than a counting shift-register moved from one address to the next either by an internal clock-pulse or a 'self pacing' clock-pulse received from the student). Alternatively, the executive agent, together with the tutorial programme, may be envisaged as a simple feedforward of predictive control mechanism.

The next kind of tutorial programme (a so called 'branching' programme) is a *Banerji game playing strategy*: that is, it contains conditional tests and conditional assignment or execution statements. It can be represented by an algorithmic flow-chart and thus as the specification of a finite automaton. The designer is able to stipulate contingencies (what is to be done *if* such and such occurs) and the minimal executive agent is more sophisticated; a finite-state machine of whatever complexity is required. Alternatively, the executive agent, together with the tutorial programme, may be envisaged as a feedback controller (perhaps with feedforward or predictive subsystems).

The 'liberalised' algorithm or condition chart calls for a still more elaborate executive agent since the order of operations is no longer always determined (one class of examples is cited; namely, the structural communication method). It is quite possible to conceive tutorial programmes that correspond to Petri nets and execution graphs as well, but none of them are called 'tutorial programmes' (these structures are characteristic of conversational teaching systems, for example, as discussed later in the book).

Form apart, tutorial programmes differ in respect of the operations they are intended to evoke; that is, there are differences in the intended interpretation. For example, from a *strictly* behaviouristic point of view a human student is regarded as a *target* organism. The prescriptive parts of a tutorial programme are seen as stimuli or combinations of stimuli that *cause* responses and the permissive parts as constraint relaxations that *enable* autonomous events which are otherwise prohibited. In contrast, if human students are seen, even in the limited context of instruction, as sentient beings able to interpret commands, questions, and the like, then the prescriptive instructions either figure as commands to learn (for exercise material and tests, local commands to *do*) or questions that elicit answers. The permissive instructions remove some of the sanctions imposed upon the student by a tutorial contract in which he agrees to interpret and obey the tutorial progamme. The major difficulty is that though *explanations* may be elicited by *questions*, an explanatory reply cannot usually be evaluated except by the student; in contrast the answer to a multiple-choice or a which question can be evaluated by quite a simple executive agent. As a result, nearly all the conditional tests involve degenerate evaluation (done by the executive agent) and the tutorial efficacy of the scheme is thereby restricted.

It is worth pursuing this point if only to justify an otherwise pernickety insistence that an algorithm *is* an algorithm (or a tutorial programme *is* a tutorial programme) only if its interpretation can be properly specified. The fact that explanatory replies cannot (in general) be interpreted, brings out an important limitation which is obfuscated if the tutorial programme is conceived naively as a 'plan for making things happen'.

None of the tutorial programmes we consider are adaptive with respect to an individual student and only one of them has an explicit mechanism for adaptation with respect to a student population.

2.1 Tutorial Linear Programmes A serial and 'linear' teaching programme can be administered by programmed texts, flash cards, or the like.

Suppose the programmed materials are presented to the student by a simple machine. The sequence of operations carried out by the machine is as follows.

1. Some descriptive or explanatory material is presented in writing, pictorially, as 'frame' or 'item' number, n, in a linear sequence.

2. A question is asked about the material in frame n. This poses a problem.

3. The student tries to solve the problem or respond to the question either (i) by constructing a written response (filling in within a blank on the frame itself) or (ii) by selecting one of the several multiple-choice alternatives; for example, by pressing a button corresponding to the chosen alternative. Generally, the student is allowed to respond at his own pace, but occasionally a mild form of time constraint is imposed to prevent dawdling or downright inattention to the material.

4. The student indicates that he has completed his response (pulling a lever, pressing a button).

5. The machine presents the correct response to frame n so that the student can compare his actual response with the correct response.

6. The student is asked to contemplate the difference, if any, between the actual and the correct response and to remedy any misconceptions that led to this deviation.

7. The student indicates (for example, by pressing a button) that his comparison is complete and that he has taken the necessary steps to put his mental house in order.

8. The machine moves frame n, the nth response, and the correct nth response from view and exposes frame $n + 1$ in the programme.

Machines of this sort and the techniques that go with them have been used and developed by Pressey (1960), Skinner (1968), Holland and Skinner (1961), Glaser (1962), Markle (1960), Gilbert (1962) and many others. These operations do not dictate a programming principle any more than the pages and paragraphs of a book dictate its content and style. Nevertheless *some* programming principle *must* be adopted. Historically, most linear tutorial programmes are based upon the idea that behaviour can be shaped according to the principals of operant conditioning and upon the side assumption that knowledge or skill can be built up sequentially, in steps corresponding, roughly, to the frames. Since the programme is linear (frame n is always followed by frame $n + 1$) there is also a supposition that one good sequence of presentation of the material can be chosen for all individuals in a target population.

My own extension and criticism of this view of learning is given, at some length in Chapter 11 and the following account also provides background data for what is being modelled, criticised, and (in part) rejected.

The precepts of operant conditioning are that operant responses are established by contingent reinforcement and that a successful response is inherently reinforcing. Hence, linear programmes are generally written to secure an expected correct response percentage in the order of 80 to 90 per cent. Apart from the establishment of correct operants, there are four main processes involved in the underlying learning model.

(1) Chaining of responses into sequences wherein the response to an item evokes the discriminating stimulus eliciting the next response (the discriminating stimulus signals an occasion on which the response may be reinforced. It thus also serves as a conditioned reinforcing event). (2) The development of 'higher level' reinforcers. (3) Discrimination, whereby responses become more selective. Finally, (4) 'generalisation', whereby responses are extended to classes of similar stimuli.

Using a 'classical' technique, the programmer specifies a terminal behaviour he would like the student to exhibit (commonly he designs a post-test to determine whether or not this terminal behaviour is achieved) and writes a sequence of frames $1, \ldots, n, n+1, \ldots, N$ to shape the student's current behaviour so that the terminal behaviour is approximated. The subject matter is thus broken down into segments and the size of the segment that goes into a frame depends upon the expected value of the student's operant-span, i.e. the size of gap he is able to fill in answering one of the questions whilst having an 80 or 90 per cent probability of being right. Some of the frames, interpolated at suitable points in the sequence, are intended to instrument chaining operations, other to carry on discriminations, and others to induce generalisations. To secure these ends, the programmer has a repertoire of standard tricks at his disposal which refer to the minutiae of the conditioning process. Perhaps the most important are (a) the provision of cueing or 'prompting' information which partially specifies the correct response, and (b) the converse operation of stimulus 'fading'. By the provision or withdrawal of cues, it is possible to adjust the correct response-probability with respect to a given type of material. Of course, the self-paced presentation allows the individual student to adjust his load within quite wide limits. He can go rapidly through the frames he finds simple, and slowly through the difficult ones.

If behaviour-shaping is taken literally, then contingent reinforcement means associating a behaviour that normally occurs autonomously and with high probability, that is, 'because the target organism likes to behave in that way' with an initially low probability behaviour that is to be established. The protoganists of behaviour-shaping are alive to the underlying difficulty; no one knows what behaviours are reinforcing and *a priori* probable.

In the past, the dilemma was resolved (or simply pushed aside) by appeal

to physiology (certain brain centres, if stimulated *do* produce pleasure and are stimulated by high probability autonomous behaviours). Whilst true, this kind of argument (which is, incidentally, part of a physiological metatheory, not part of behaviourism itself) fails to account for obvious individual differences due, at least, to previous and uncontrolled conditioning of the target organism. Various schemes have been used to solve the joint problem posed by ignorance of what *is* reinforcing and the existence of individual differences due to esoteric experience. All of them go beyond the strict behaviouristic framework, though this point is often glossed. The most lucid scheme is the 'reinforcement menu' of Homme (1966). Students exit, periodically, from the programmed or scheduled activity, and (if they are to be reinforced) choose from a 'menu' of possible rewards before returning to work. The menu, however, is chosen and updated according to criteria that lie outside the learning model and are the responsibility of a 'reinforcement manager' (for example, the teacher in charge of an infant school).

The same procedures are justifiable, for very limited types of material, with respect to a systemic model in which teaching is assumed to establish goal-directed or problem-solving systems in the student's mind. At first sight, only the jargon is modified. The corrective signal (comparison of the actual response and the correct response) provides 'knowledge of results' information rather than reinforcement and the external goal-achievement test implied by the provision of the corrective knowledge of results becomes internalised as learning proceeds. Cueing information partially solves the problems posed by the stimuli and may reduce the overall goal to subgoals, but some care is needed (Chapters 5 and 8). Beyond very restricted situations and learning conditions this reinterpretation entails much *more* than a change of jargon and notation scheme.

The 'classical' technique is supplemented by a more systematic behavioural method called mathetics, chiefly developed by Gilbert (1962) and his associates. As before, a terminal behaviour is stated, but this is systematically transformed into a 'synthetic prescription' that states how the student's current behaviour is to be modified in order to attain the terminal criteria. The basic units of the programme are no longer frames but 'exercises', and the small step-concept is de-emphasised or even discarded. The exercise brings about a specific behavioural change, and it entails instructing, cueing and observing stimuli over and above the discriminating stimulus that elicits the main or 'mastery' response. In other words, each exercise is a pattern of behaviour-changing operations; typically, it involves a sequence of frames in which a mastery response is demonstrated, prompted (or cued) and released. Further, in the construction of an exercise, the programme writer takes systematic account of the symbolic representation of the subject matter (the theory behind the behaviour). Mathetics is the best developed and most elegant programming technique, but I believe its nomenclature is positively misleading. An 'attention-directing stimulus', for example, acts within the theory, as a command (not as a stimulus); a 'questioning stimulus' *is* a real *question* (again not a stimulus).

Whichever model is adopted, it is evident that the serial programme is a feedforward control process. The programme instructions control learning in so far as they are based upon a predictive model for how the student will react to them. The onus for the intimate control of learning is placed firmly on the subject's shoulders. (This comment applies very strongly to the mathetics programmes.) The student is responsible for pacing the trials and he acts both as comparator and corrective agent. True, a feedback signal is received by way of knowledge of results information, but it is used, in comparison and correction, by the *student*, and not by the *machine*. (The programme sets up conditions for internal, student-mediated feedback, but it does not itself exert feedback control over the learning process.)

It will be clear that not every sequence of frames constitutes a programme. To qualify for this title, it must be possible for the programmer to say what each frame is intended to do, and why (with reference to the criterial or terminal behaviour) it has been introduced at a given point (i.e. frames should establish the conditioned response '*x*' or 'make the discrimination *y*' or whatever). So far as behaviour shaping is concerned, only the *programmer* is required to exercise this much perspicacity. The pupil, subject or student is regarded (theoretically speaking) as a malleable entity on which the programme acts; albeit, an entity able to emit the operant responses that allow it to act. The linear format does not, of course, dictate a behaviour shaping view of the student (or vice versa). It is however well suited to this view.

2.2 Tutorial Branching Programmes A branching programme (again, imagine it is administered by a teaching machine) is an instrument capable of mediating feedback control over the learning process. Because of this the learning model employed in constructing a branching programme can be much less specific than the model for a linear programme addressed to the same subject matter and student population. Since it *is* feedback regulated, unknowns can be filled in during the execution of the programme; for example, the programme may prescribe a test for a condition and certain subsequent operations that are to be performed *if* certain test values are in fact obtained. By way of contrast, the model for a feedforward (linear) programme must, if it is to exert the same modicum of control, incorporate estimates of or assumptions about the values that will be obtained. The psychological background is correspondingly eclectic (behaviourism, cognitive psychology, motivational psychology, and so on).

The distinction between linear and branching programmes is illustrated in Fig. 90, where (*a*) represents the arrangement of frames (nodes, in the network) for a linear programme and (*b*), (*c*), (*d*), (*e*) correspond to various types of branching sequences.

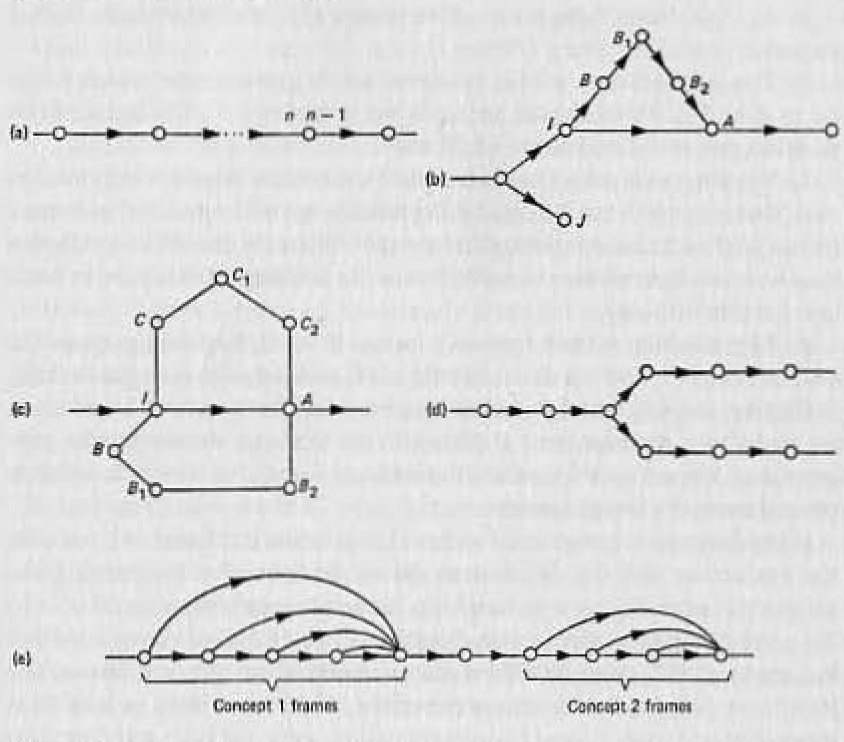


Figure 90 Forms of programme.

Suppose, for example, the student finds himself, on trial *n* in a teaching process, at frame *I* in Fig. 90(*b*). (It will be evident from the figure that being located at frame *I* on trial *n* depends upon a previous path through the network, i.e. the student could have reached frame *J* on trial *n*.) The following operations are performed by the teaching machine or the student.

1. As in a linear programme, the frame *I* material is presented to the student.
2. A question is asked about frame *I*.
3. The student is required to respond either constructively or selectively.
4. The response is evaluated. If the response is selective and mechanised (for example, button selection) then it is evaluated by the machine relative to some criterion. In the least elaborate case the evaluation consists in a comparison between the response made and a set of responses which, the

programmer has predicted, might have been made. This set certainly contains the correct response. But it may also contain response possibilities which, if they are selected, indicate the existence of misconceptions about the material in frame *I*.

5. As in the linear case, the machine presents the student with the correct response to trial number *n* (Frame *I*).

6. The machine uses a built-in decision rule to determine which frame (*A* or *B* in Fig. 90(*b*)) should be displayed at trial $n + 1$. The input of this decision rule is the evaluation of (4) above.

7. The student is asked to contemplate the correct response information and (if necessary) to use it in modifying his concept of the material in frame *I* (trial *n*). When he has considered the matter sufficiently, the student indicates that he is ready to receive the next frame (by pressing a button, or in some comparable fashion).

8. The machine selects the next frame *A* or *B*, depending upon the evaluation of (5) and the decision rule of (6) and presents it to the student.

Clearly, steps 4, 6 and 8 mediate external feedback-control based upon an evaluation function and a decision rule that are chosen by the programmer. Steps 5 and 6 mediate the internal corrective feedback which is present even in a linear system.

If the response is constructed rather than selected it is usual to hand over the evaluation and the decision to the student, i.e. the student is given certain rules (saying how to weigh up his constructed response relative to the correct response) and certain directives (if you feel that your evaluation is *a*, go to *A*, if *b*, go to *B*). The student himself selects the next frame. The important point is that a subsystem of the student has been isolated as a supervisor and controller of his own education, using the rules and directives that are given. This may or may not be a good thing to do depending upon the student and the subject matter. The expedient certainly does impose a real cognitive burden.

Several different sorts of programme network may be appropriate according to the intention behind the decision rules. The case just considered in Fig. 90 (*b*) is a network structure containing a mainstream linear programme with remedial loops. In the simplest possible conditions the response to frame *i* is evaluated as right or wrong and a decision is made to select *A* (for presentation at trial $n + 1$) if the *n*th response (to frame *I*) is right, to select *B* if it is wrong. Frame *A* is part of the main stream of instruction; in a linear programme it would be the next item after frame *I*. Frame *B*, on the other hand, is the start of a remedial loop B, B_1, B_2 which contains frames intended to iron out the misconceptions about frame *i* manifest by the evaluation 'wrong'. The student is returned to the main instructional stream at frame *A* only when he has shown evidence that his

misconceptions are put right. If the evaluation and the decision rules are more elaborate, the remedial action can be more discriminating and the programme structure may, with advantage, be more detailed. For example, in Fig. 90(*c*) there are two different sorts of remedial loop (B, B_1, B_2 and C, C_1, C_2) one of which is selected when the subject is 'wrong' at frame *I*; *which* one is selected depends upon *how* he is wrong. The use of remedial loops is akin to a cueing procedure extended over several frames. Because it is extended, it can be a controlled cueing procedure.

The evaluation and decision process can also be used as a mental testing facility to determine what sort of individual each student is, and to prescribe essentially different forms of instruction for different sorts of individual. A typical programme network is shown in Fig. 90(*d*), where the first part is a test sequence for sorting students into classes and the remainder serve to administer several more or less linearly arranged courses contingent upon class membership. It is usual (as in this network) to provide retest facilities for reassigning a student from time to time.

In section 3 we present a tutorial control process founded on the skip linear-programme network of Fig. 90(*e*). It is a series of linear programmes in which each one is addressed to the inculcation of one concept. From time to time, a test of concept mastery is made and, if the concept has been mastered, the student is directed to skip over the remaining frames dealing with this concept. Hence the name 'skip linear'.

The first branching programmes were written by Crowder (1960) who also designed the first teaching machines to administer them. Many people have subsequently been active in this field: Galanter (1959), Lumsdaine (1963), and Stohlurow (1961), to mention only a few of them.

2.3 Structural Communication One special programming technique is the structural communication method of Hodgeson and Bennet (1967) and Hodgeson (1968). Any study unit (corresponding roughly to a programme segment) consists of the following sections:

1. A statement of the author's intention, that is, the underlying topic and goal.

2. A viewpoint which orients the student and leads him to explore the field.

3. A presentation of the background material relevant to the concept of the study unit.

4. An investigation in which the student is posed problems, say five of them.

5. A solution phase involving a response indicator. This is a page containing (say twenty) statements, all of which refer to some of the problems.

The student tackles each problem in turn and selects statements that he believes to be true.

6. A discussion guide. Contingent upon the student's selecting and not selecting certain combinations of statements in the response indicator; he is directed, through a number of stages of analysis, to study some of the 'discussion comments' which are described below.

7. 'Discussion comments' which remedy misconceptions, resolve ambiguities, make statements and encourage the student to justify his point of view. The comments also direct the student to do something by going back to the problems, the viewpoint, or the intention.

Progress through this system is logical and systematic enough. Superficially, the student adopts a serial path which is recordable. He is not held in the straitjacket of sequential presentation and, broadly speaking, the wiser he is the less restricted he will be. Whereas the majority of branching programmes are algorithmic in the sense that they correspond to serial flow-charts, a structural communication programme can only be represented by a condition chart. For example, step (5) above is a condition test in which the required evaluations can be carried out in any order (usually with interruption and stacking of data permitted) or even simultaneously. By the same token step (7) contains unordered execution or assignment statements. Whether or not the student uses the freedom permitted varies a great deal but he is certainly, and rightly, encouraged to do so.

2.4 Discovery Learning Structural communication is a procedural reification of a doctrine called 'discovery learning' or (since undiluted discovery appears to be a pretty haphazard activity) 'guided discovery' and is a compromise between free rein and the authoritarian approach. The basic idea is excellent. Moreover, it is empirically supportable. Learning takes place if and only if the student is impelled to discover a connection between already learned concepts or to construct a concept *de novo*.

As often stated, however, 'discovery learning' is a sloppily made concession that learning is chiefly a matter of cognitive processing rather than 'behaviour shaping'. A few advocates of the principle take the trouble to be more precise, mainly in operational terms describing what tutorial acts should be performed to induce discoveries, rather than in terms of the mechanism involved. Thus, the work of Bennet and Hodgeson (cited previously) is one outstanding exception to a generally cavalier treatment of what on earth 'discovery' is, and the work of Belbin (1969) is another.

From a cybernetic point of view it is hard to condone an assumption that 'discovery' is a property of the mind without delving into the systems involved. This acid criticism is spurred on by a real hazard. There is no doubt that the cognitive operations called 'discovery' are very important

and that they are much more difficult to comprehend than 'adaptation' (induced by behaviour shaping). The majority of writings on the subject of discovery are intended to ease the difficulty, so that discovery is fictionally comprehensible as a popular tagword. In the process, 'discovery' is used to name a scholastic tautology (a rose smells sweet because it has sweetness; a mind discovers because it has the discovery faculty) as a result of which the word falls into disrepute and the excellent concepts, which actually underpin the discovery doctrine, fall with it.

2.5 Behavioural Objectives During the last decade or so one movement (above all others perhaps) has had a salutory influence in leading people to think clearly about training and education. The underlying dogma is 'behavioural objectives' (Gagne, 1962). As a terse and insufficient precis, the dogma holds, as follows.

(a) That the terminal criteria of training and teaching can be expressed in terms of behaviours (often very complex ones; for example 'multiply numbers *A* to *B* using a slide rule' or 'multiply numbers *A* to *B* using a table of logarithms' or 'multiply numbers *A* to *C*'). If these behaviours are correctly performed a student 'knows' whatever (multiplication, for example) is required of him. The word 'knows' is placed in inverted commas because of a contention that the galaxy of behaviours is all that an instructor can know (without inverted commas) about the student. (b) The complex terminal behaviours can be broken into segments and the correct performance of these smaller pieces of behaviour, usually induced by behaviour shaping, is a sufficient entry criteria for a behaviour shaping operation that inculcates the terminal behaviour (obviously (a) and (b) can be iterated with respect to each behavioural segment until very small pieces of behaviour remain as primitives and, in principle, until there is no residue).

The whole gamut of methods and models (with the possible exception of structural communication) are popularly adumbrated under the rubric, 'behavioural objectives'. Training programmes are certainly devised on that assumption and often more or less work (though how effectively it is hard to tell). The plain fact is, however, *if* discovery is taken seriously then neither the contention of (a) 'terminal behaviours are all *we* can know' nor the contention of (b) 'it is possible to segment any skill' hold water. If discovery does mean a cognitive event then we *can* and *must* know, in order to support this view, how a student comprehends a subject matter, describes it, and explains it. True, the data may come from objective records of what the student does but they are not interpreted as behavioural records. For example, physically and operationally there is a great deal of difference between 'multiplying' and 'explaining multiplication' though it may be the case that the *explanation* is physically manifest as making a *model* for the multiplication process. The latter activity is neither interpreted as a 'multiplication behaviour' (it is, of course, a behaviour of some kind), nor is it viewed by the student as an elaborate multiplying response.

By the same token but in respect of (b) there is every difference between behaving and knowing. Some confusion is engendered by the fact that both of them are species of *doing*; in behaviour, overt operations are *done*; in knowing, operations are done to concepts, in order to combine them or to construct them. With rare exceptions, it is not possible to express *what may be known* in the same way as *what may be done* (if discovery is taken in earnest, that is). Given genuine discovery, not all states of knowing can be induced by a concatenation of behaviour segments or (as stated under (b)) inferred from the correct performance of a complex terminal behaviour.

These comments are not intended to dismiss the doctrine of behavioural objectives out of hand. Within a certain compass the doctrine has great and well attested utility. But either the definition of 'objective', needs revision, or (better, for 'objective' is sound enough in its proper domain) the limits of the dogma need to be recognised.

3 A Restricted Tutorial Programming Scheme

The following scheme was used for several industrial training application and, in collaboration with Xavier Salazar Resines (1966), at the National University of Mexico, UNAM, for academic instruction[1]. It is a better scheme than most, but the chief reason for dwelling upon it is to exhibit the defects which became apparent when it was put into practice. These defects are liable to beset any scheme which is based, as this one is, upon the ideas discussed earlier in the chapter. The defects are especially obtrusive if the scheme is intended (as this one is) to give substance to the notion of discovery.

The basic unit of mentation is called a concept and this is imaged as a goal-directed or control process. That is, a *concept* is an organisation which forms an hypothesis, acts to satisfy the hypothesis, and tests for its confirmation or denial. It may thus also be interpreted as a 'problem solver'. As in the last chapter, concepts exist at certain levels of control (parts of a hierarchy of control, designated *Lev 0*, *Lev 1* and so on. The *Lev 0* processes act upon an environment; the *Lev 1* processes act to construct or modify concepts placed at *Lev 0*; the *Lev 2* processes (if they exist) act upon *Lev 1* processes.

1. The work was strongly influenced by Roger Diaz De Cossio and Juan Cassias, also of UNAM.

Given the identification 'problem solver', learning is a problem solving *of* problem solving, i.e. learning, which is a *Lev 1* process, either constructs concepts, or remedies a deficiency in a repertoire of *Lev 0* processes.

A tutorial programme operates in concert *with* the *Lev 1* processes in a student's repertoire or *upon* the *Lev 0* processes in his repertoire (on a par with his own *Lev 1* processes). Hence, if the student's hierarchy of problem solvers (alias *concepts*) are placed in register with the tutorial programme, this programme prescribes *Lev 1* processes.

A discovery is defined as the construction of any *Lev 0* concept (in fact, various kinds of discovery are postulated since various kinds of construction are distinguished). A guided discovery is a discovery involving the cooperative interaction of a *Lev 1* process prescribed (for example) by a tutorial programme.

In order to compose *concepts* (goal-directed, problem solving processes) into aggregates, they are represented in a common manner. The chosen idiom was the 'TOTE' or 'Test operate test exit' unit (Fig. 91) introduced by Miller, Galanter, and Pribram (1960) rather than my own 'control units' (Pask, 1968, 1970b) which were less familiar to the users. Such elements (really 'If. . . then. . . else' statements) may be concatenated into sequences and formed into an hierarchy (Fig. 91). Any of these composite structures is also a concept. The representation is extremely restrictive (a) because it is confined to serially executed procedures and (b) because it is tailored to fit a *post hoc* account of execution in which the structure is obscured. However, with the augmenting notation given in the captions to the figures, it suffices and is peculiarly well suited to the representation of the operations induced in an executive agent by a tutorial programme.

Manifestly a TOTE unit (since it is really an 'If . . . then . . . else' statement) must be embodied in data storage and executed by some processor. Two storage and execution loci are postulated (both of them functionally rather than psychologically distinguished), namely, the 'intermediate storage' and the 'long-term storage' of the brain. Intermediate storage is a computing mill and is the seat of interference phenomena, and the like; long-term storage is essentially indelible. The other category of data storage, usually distinguished in the literature, is short-term storage (in which, for example, lists of data are rehearsed); it will be evident that the embodiment of any TOTE unit (hence, in this image, any concept) incorporates and organises a short term storage operation.

The primary optimising criteria has to do with intermediate storage from which *concepts*, if shown to be useful or valid, can be transferred into long-term storage. The criterion is as follows. Intermediate storage has a finite data capacity and may be over taxed. Any concepts coexisting in intermediary storage are liable to interfere under execution and thus to be demolished.

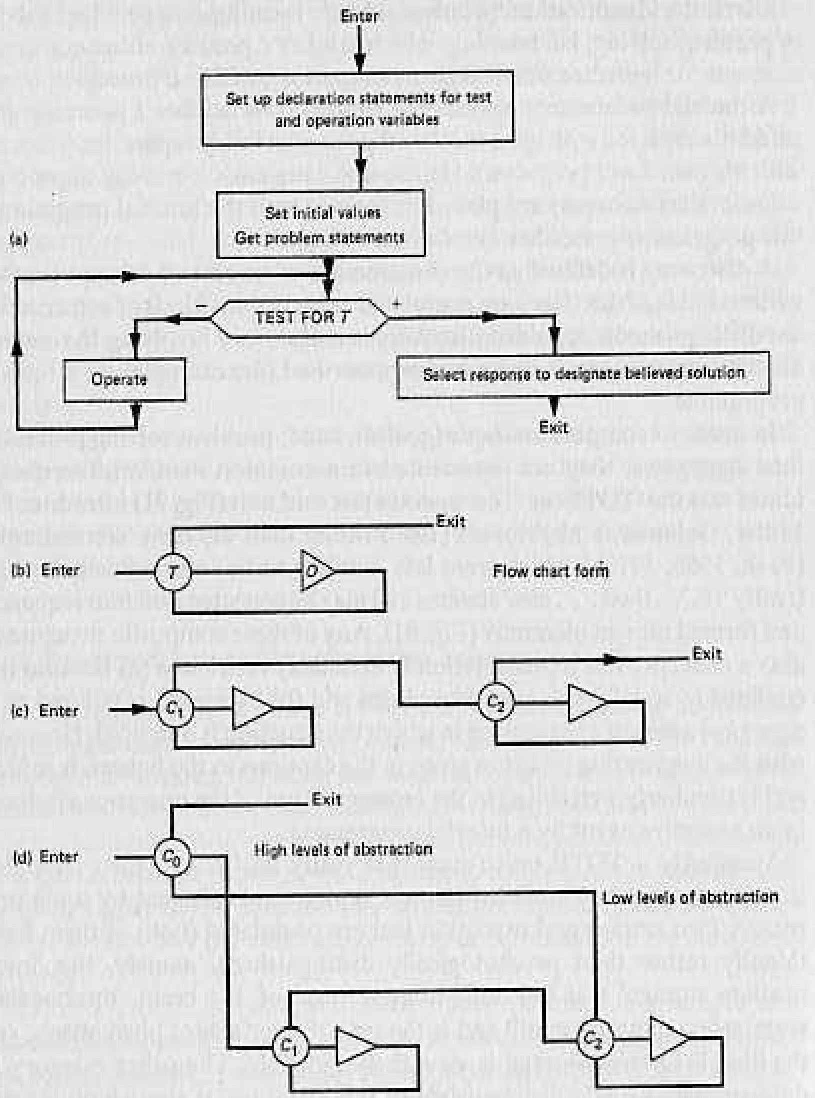


Figure 91 The TOTE (test, operate, test, exit, unit) notation and permitted compositions of TOTE units: (*a*) meaning of TOTE as an organisation; *T*, test for goal and *O*, operation performed upon information in immediate memory to approximate *C*; (*b*) shorthand form. All subsequent units are labelled in their test part either by *C* (the test for hypothesis under a concept) or *G*, *g* (educational goals and subgoals); (*c*) concatenation of TOTE units; (*d*) hierarchical structure of TOTE units. Concept C_0 is placed (very loosely) at a higher level of abstraction than C_1 and C_2.

Teaching 'sets up' a large number of *concepts* that are embodied in intermediary storage locations. If concepts are regarded in isolation any one of them is 'set up' by the cyclic tutorial operation shown in Fig. 92. A serial tutorial-programme thus loads intermediary storage with concepts; some of those established previously were needed as components in the next construction step. Of the previously established concepts, needed to form a particular concept at a given step, some may be retrieved from long-term storage where they are immutably inscribed but others have not yet entered long-term storage. These (and the concept being constructed) may be destroyed either by overload or interference and an effective tutorial programme should (by hypothesis) be designed to avoid either of these possibilities.

Tutorial operations are also conceived as TOTE units, set up by the tutorial programme, and intended to bring about the *tutorial goals* (*G*) of establishing a concept (*C*) for each topic in a subject matter instructed by the tutorial programme. A nested hierarchy of tutorial TOTE units is a *format*; in this case based on the *skip linear* scheme (the simplest branching network). This approach is founded upon two propositions.

1. Any long sequence (in principle, an indefinitely long sequence) of frames that are addressed to a single topic and are intended to inculcate a concept for this topic, may be regarded as a single TOTE unit (with the tutorial goal of teaching a concept for the topic in question). The validity of this proposition is a simple consequence of the fact that a tutorial cycle of the type discussed in section 2 *is* a 'test operate, test, exit' sequence provided that the data it operates with are interpretable by the student. As noted earlier, a tutorial TOTE unit exists at a level of control *Lev 1* and it *acts upon* concepts in the student's *Lev 0* repertoire though it also *interacts with* processes in the student's *Lev 1* repertoire.

2. Any subject matter has a main topic which can be broken down into subsidiary topics (a most dubious proposal).

However, if (2) holds, then, within the constraints of the *format* a tutorial programme consists in a prescription for an hierarchy of tutorial TOTE units that inculcate concepts for subsidiary topics (as tutorial subgoals), and that ultimately teach a concept for the main topic.

3.1 Construction Procedure 1. The main topic to be taught is given by the subject matter.

2. The main topic is reduced to constituent topics in the following manner (which closely parallels the arguments of Gagne, 1962, 1969). First observe that the main goal (to establish a concept of the main topic) could be achieved if (i) the student has a number of subsidiary concepts $C_1, \ldots, C_r$ and (ii) if he is provided with an appropriate sequence of performance

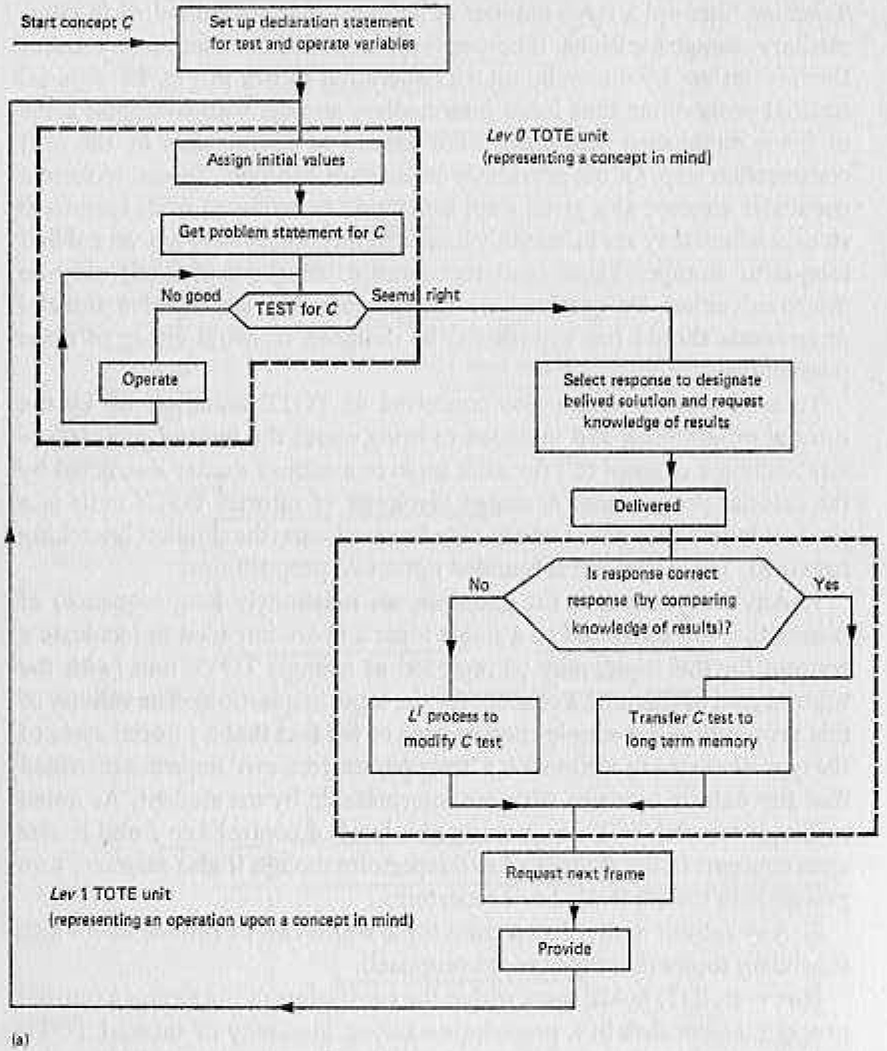
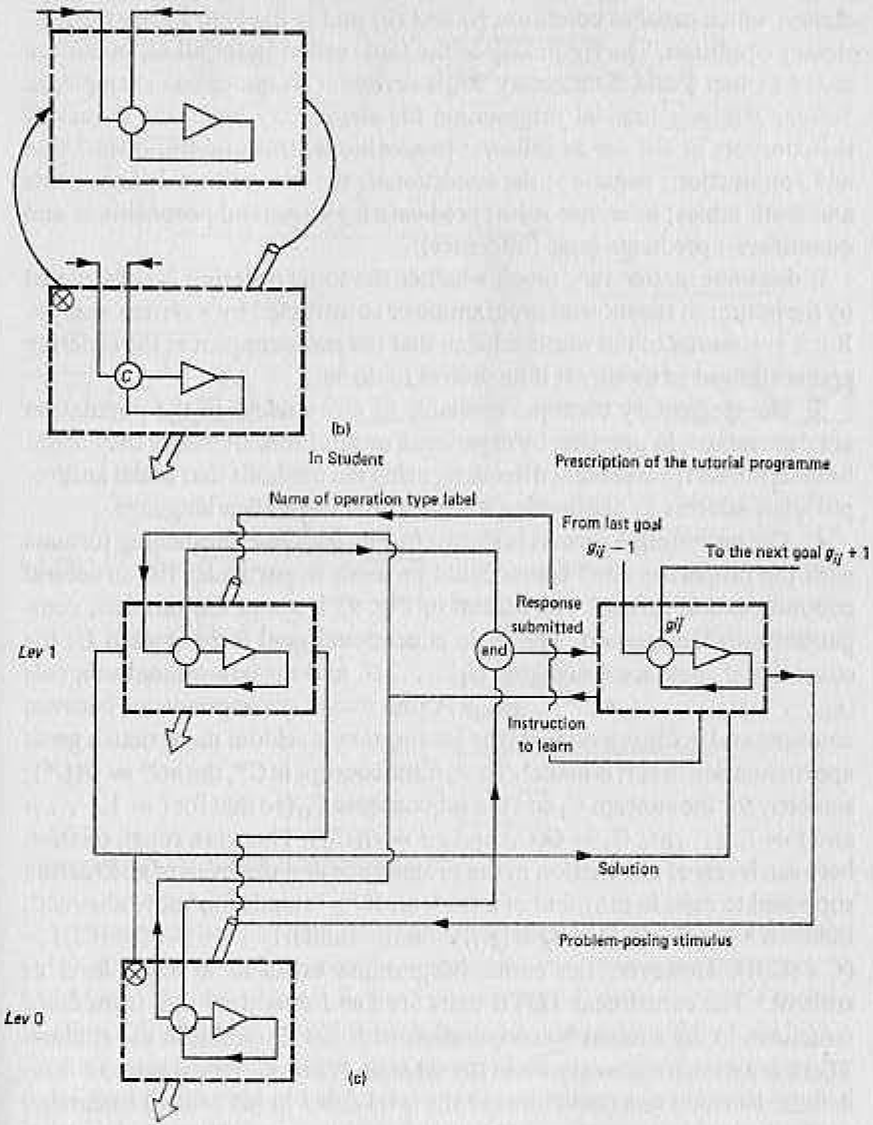


Figure 92 Representation of a tutorial cycle: (*a*) notation and meaning for any process; (*b*) shorthand form. A parametric arrow penetrating any dotted line indicates concept learning on student; an arrowed box delineates a TOTE unit which changes the data on which a TOTE unit tests or operates; a description specifies the TOTE units enclosed in the dotted-line box. (*c*) Relation between a tutorial TOTE unit (with goal *G*) prescribed as part of a tutorial programme and a TOTE unit (representing a concept *C*) in a student's mind. Each dotted box within the student may contain any composition of units which, for brevity, are represented as one unit in this chart.



instructions; concisely, a suitable algorithm that refers to the C_k; $k = 1, \ldots, r$. The problem is to select a breakdown of the main concept $BR = C_1, \ldots, C_r$ for which an algorithm is available. In general, this breakdown is not unique. There is a set, BR^* of breakdowns BR, from which one must be chosen. To narrow the choice, a further condition is introduced; namely, that for any BR in BR^* all of the C_k in BR must be associated with a legitimate concept ordering in the sense of step 6 (below). In practice some BR is chosen which satisfies condition (i) and (ii) and is likely to satisfy the legitimacy condition. The legitimacy of the topic ordering for all C_k in *BR* are tested at step 6 and if necessary *BR* is revised. (To quote one example, in Salazar Resine's tutorial programme for elementary logic, the names of the concepts in *BR* are as follows: propositions, truth tables; disjunction and conjunction; negation; the conditional; the biconditional; arguments and truth tables; inference rules; predicate logic (general propositions and quantifiers); predicate logic (inference)).

It does not matter very much whether the topic ordering is constructed by the author of the tutorial programme or constructed by a system analyst. But it is *essential* to the whole scheme that the *author* approves the ordering and is allowed to modify it if he desires to do so.

3. The elementary concepts available to any student in the population are determined, in practice, by experience or intuition. In theory they might be determined from student discourse, using the methods that social anthropologists address to discovering salient words in an alien language.

4. The programme format is shown in Fig. 93. Other branching formats with the properties cited below could be used, in particular the structural communication format. The format of Fig. 93 is about the simplest, compatible with the method. The main educational goal is designated *G*; the educational goals are designated $G_1, \ldots, G_n$ and the educational subgoals $(g_i, \ldots, gi_{m_i}) \ldots (gn_j, \ldots, gn_{m_j})$. A one to one correspondence between concepts and goals is assumed (the assumption is seldom more than a gross approximation, but it is made). Thus, if the concept is C^*, the $nG^* = G(C^*)$; similarly for the concept C_i and the subconcepts C_{ij} (so that for $i = 1, \ldots, n$ and $j = 1, \ldots, m_i$, $G_i = G(C_i)$ and $gij = G(Cij)$. There is a rough relation between levels of abstraction in the programme and the levels of abstraction supposed to exist in the mind of a student. If '$>$' stands for 'more abstract', then $G^* > (G \in \{G_i\}) > (g \in \{gij\})$ and similarly $C^* > (C \in \{Ci\}) > (C \in \{Cij\})$. However, the entire programme exists in a single level of control.[2] The constituent TOTE units are *Lev 1* operators that form *Lev 0* structures in the student by cooperation with *Lev 1* systems in the student. There is an inherent weakness in the scheme. Whereas the student can form definite learning sets (structures at the level *Lev 1* in his control hierarchy) the programme cannot do so. If it could, then there would be time variable relationships between the G_i and between *gij* (not merely an ordering $1, \ldots, n$ or $i_j, \ldots, i_{m_j}$). Further any programme with this property would be real-time and on-line adaptive; the adaptive process being formalised as a *Lev 2* box in the teaching system. The adaptive process of steps 8, 9 and 10 (below) is isomorphic with such a *Lev 2* box. But it does not act at the individual level, i.e. the programme is adapted only to a population of students.

2. Apart from cueing operators which may be regarded as *Lev 0* objects.

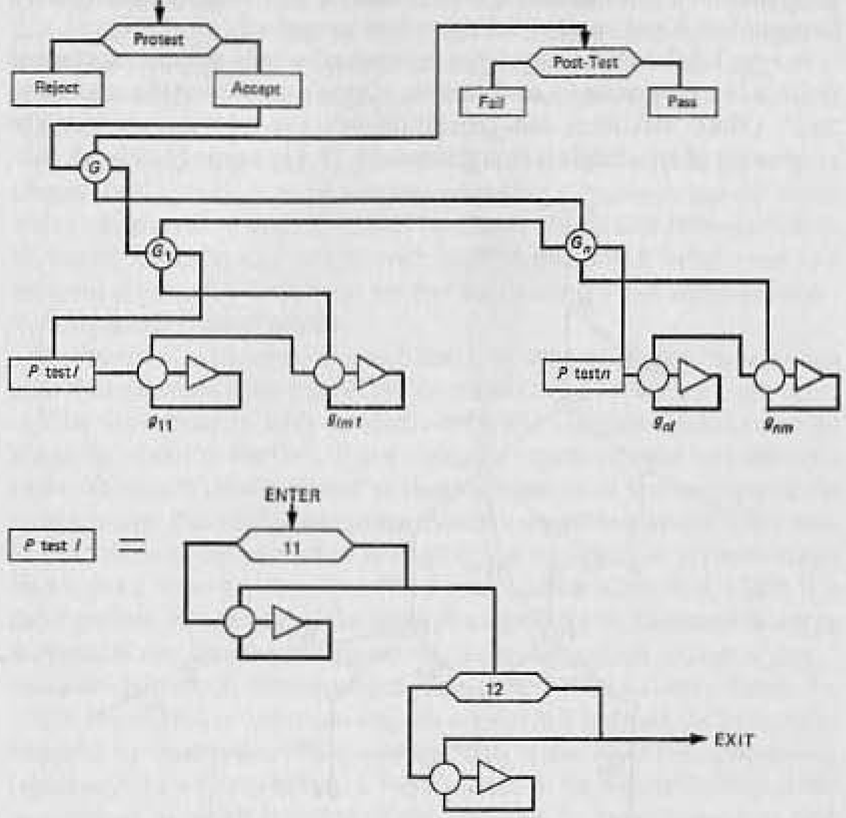


Figure 93 A branching programme format.

5. Consider the first topic of the main topic stated in step (1). It is made of subtopics. Some of these may be elementary concepts and others may be introduced by definitions. Form an ordering of these topics (Fig. 94), starting with the first, in which each subtopic is represented by a node in a directed graph and in which *A* is above *B*, if and only if, a concept for topic *B* must be attained before *A* is instructed, either a concept for topic *B* is part of a concept for topic *A* or the attainment of a concept for topic *B* is a necessary precondition for attaining a concept for topic *A*. To each directed edge (directed arc) of the graph, and to each terminal node, assign a type label according to the type of process used in building a concept for the topic at the node where the edge terminates.

So far as the teaching is concerned, a type name is the name of an instruction which the student is asked to carry out, on the assumption he has a *Lev 1* mental operation for doing so. The list of type names is restricted to generalisation, concatenation, discrimination and definition. *Discovery* (unqualified) is any or all of the operations so named.

A type label may be any 'type expression' where a 'type expression' consists in a type name or combination of type names using the connective 'and'. (Thus 'definition and generalisation' is a type expression.) The assignment of type labels is straightforward. If *A* is uniquely above *B*, then

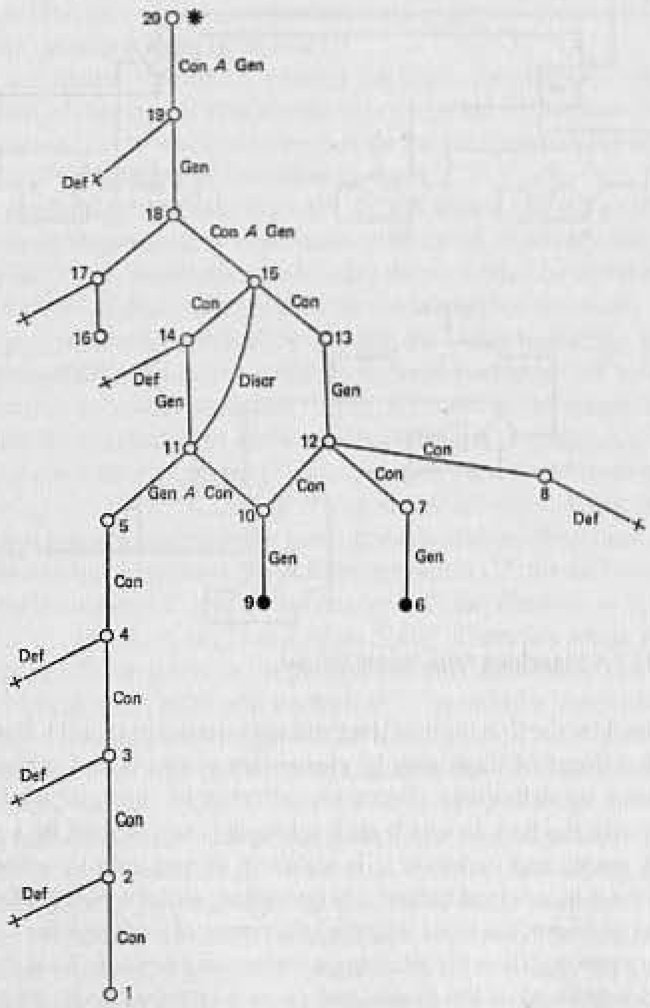


Figure 94 A topic ordering for part of an elementary logic a tutorial programme. The numbers are assigned by a heuristic to indicate sequence of instruction. Type names: Def, definition; Con, concatenation; Gen, generalisation; Discr, discrimination; ○, subconcept node; ● terminal, node; *, concept at head of ordering; ×, terminal branch for introducing definition.

the edge connecting *A* to *B* is labelled by the process whereby a concept of *A* is obtained from a concept of *B*. If *A* is above *B* and *C*, then the edge *AB* is labelled by the process whereby a concept of *B* is used in building (part or all of) a concept of *A*, and the edge *AC* by the process whereby a concept of *C* is used in building (part or all of) a concept of *A*.

The topic ordering is closely related to an ordering of skills in the sense of Gagne (1962, 1969). It is also closely related to a 'concept matrix' in the Ruleg programming system devised by Glaser (1962) and Homme (1966). However, a rectangular matrix with topic names which label rows and columns is unwieldy for a large programme, since it is not often necessary to compare all pairs of names.

If subtopic *A* is above subtopics *B* and *C* in a topic ordering, then we can infer that a student having concepts for *B* and *C* could achieve a concept for *A* if he were given suitable instructions (this is Gagne's condition for an hierarchy of skills). Further, at first sight, the organisation of subtopics in a topic ordering is closely related to the organisation of the concepts of the main concept. But on closer scrutiny, there are several important differences.

The 'instructions' referred to in step (2) (the organisation of the concepts into a main concept) form part of a performance algorithm which is a *Lev 0* process. In contrast, if the name of a type label or the names in a type expression are issued as instructions, each instruction brings a *Lev 1* operation into effect; it is an instruction to learn, not just an instruction to do.

The instruction to learn can only be obeyed if it respects the limitations imposed by intermediate data storage. This is the basis for determining legitimacy. In step (6) below, a topic is said to be legitimate only if the instructions to which it will give rise are likely to prove consonant with storage capacity limitations.

6. Suppose it is necessary to learn one concept at once. If so, several different subconcepts must be kept in mind (and by hypothesis, kept available in intermediate memory) whilst others are instanced. It is possible to choose a path (a sequence in which the subtopics are to be instructed) that will minimise the loading by minimising the number of stacked prerequisite concepts that will be created by the tutorial programme based on any one path. In case there is no unique path, a set of 'best' paths is determined. Several heuristics are available for numbering the subtopics and thus delineating the path. They differ slightly according to the assumptions made about interaction in intermediate storage. One of the heuristics is applied and the 'optimally' numbered topic ordering is tested to see whether a numerically valued property, μ, exceeds a critical limit interpreted as the estimated capacity of intermediate storage. If estimated capacity $> \mu$ then the numbered topic ordering is deemed legitimate. If not, then either (i) the choice of topics in the main topic (the choice of *BR* in *BR**) made at

step 2 must be revised or (ii) it is necessary to introduce repetitions in the instruction of a single topic (which is usually self-defeating; most of them increase the value of μ).

7. Suppose that the numbered topic ordering is legitimate. Using the programme format of Fig. 93, the author (i) writes a test for concepts of elementary topics and (ii) writes a topic statement (goal statement) for topic 1. This does not describe the topic (still less teach a concept for the topic). It must, however, allow the student to recognise whether or not he has mastered the topic after suitable instruction, i.e. whether or not he has achieved the goal of learning a concept for this topic. Further, it shows the student what he and the programme are aiming for. (iii) Write a precondition test, *P*-test 1, covering all the topics other than definitions that are represented by terminal nodes in the numbered topics ordering. (The precondition test contained remedial loops, as indicated in Fig. 93. If the student does not have one of the subconcepts, he is instructed (remedially) until he does have it.) (iv) Taking subtopic 1 in topic 1 the author, using the type labels, writes eight or more sequentially cued, skip linear-frames (possibly interpolated with linear frames) on examples dealing with this subtopic; having stated the subtopic in the first frame. This is essentially the recommendation given in Glaser and Homme's Ruleg programming system augmented by the more recent comments of Davis and Hartley (1972). The process is repeated for subtopic 2 and so on, until the last. (v) A test is written for topic 1 as a whole (by definition of a topic ordering this is a test for concept number m_n). If the student fails in this test, the present format returns him to the start of the programme segment for concept 1. It would be advantageous to return him to remedial material.

8. Groups of students from the target population are matched with respect to verbal abilities and speed of programme reading. Each group of students should exhibit variations characteristic of the target population and the groups are labelled for future reference as *A*, *B*, . . . and so on.

The programme segment for topic 1 is administered to all members of group *A* and their performance is recorded, individually on a record sheet, together with their comments about the material. These comments are of the type 'bored' and 'cannot see where the argument is going' and are of a standard form. Data for the entire group of students is conveniently summarised by a tabulation (Fig. 95) which relates the programme format directly to the topic ordering.

Various criteria can be applied (for example, to secure a mean correct-response frequency or a mean correct-response latency). In Mexico, we regarded the data as a *clinical history* and most of the *diagnosis* of why a *group* did not learn was done by a physician, supported with analytic facilities for searching back along the records and for assessing the records

	Concept name Cl_i																		
	Subconcept name C_i1									Subconcept name C_i2									
Frame number	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88
Frame type	*B*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*D*	*C*	*A*	*A*	*B*	*A*	*A*	*A*	*A*	*A*
Mean latency	2	4	4	7	4	6	3	2	0	1	2	3	5	4	5	2			0
Mean correct responses	8	5	7	6	4	2	2	1	0	8	7	5	6	1	2	4	2	0	0
Mean mistakes	0	3	1	2	4	6	6	0	0	0	1	3	2	7	6	4	6	0	0
Mean omissions	0	0	0	2	4	4	7	7	8	0	0	0	0	0	5	7	7	8	8
Comment 1 frequency	0	1	2	0	1	0	0	1	0	0	8	0	0	0	1	0	0	0	0
Comment 2 frequency	0	2	0	0	0	0	0	1	0	0	2	0	0	0	1	0	0	0	0
Comment 3 frequency	0	5	7	0	0	1	0	0	0	0	3	2	0	0	1	0	1	0	0
Comment 4 frequency	0	7	8	0	1	0	0	0	0	0	5	4	0	5	0	1	0	0	0
Concept latency					4										3				
Concept mistakes					22										29				
Mean frames concept					40										45				
Rate of increase of mistakes			0·43			0·57							0·55			0·44			
Comment 1 frequency					5										9				
Comment 2 frequency					3										3				
Comment 3 frequency					3										7				
Comment 4 frequency					16										15				

Figure 95 Recording sheet for tutorial programme data from student group.

of individual students in an attempt to pinpoint specific pathologies. The output from the clinician (or, less effectively, from an analysing heuristic) is returned to the programme author who is required to modify any frames that cause difficulty.

The trick in the scheme is as follows. As a crucial part of step 2, the author approved the topic ordering (either he constructed it or selected and modified an independently constructed ordering). Whether he writes according to this structure is a different matter. But, unless he does so, the data (whatever the judgment criteria employed) will not support his work. Hence, the author is given liberty, at this stage, to change the topic ordering (though he is naturally loath to undertake such an onerous task unless it is essential). As a result *authors* seem to learn to take the topic orderings they have approved quite seriously.

9. The programme segment is rewritten (in rare cases the topic ordering may also be changed). The rewritten programme is now administered to group *B* of the evaluationary study students and step 8 is repeated. The adaptive process goes on (in practice, not beyond group *C* or *D*) until the basic criteria are satisfied.

10. When the programme segment for topic 1 is complete, steps 6, 7, 8, 9, are repeated, for topic 2 and so on, until all the topics in the main concept have been exhausted.

11. A post test is devised covering the concepts of the main topic as a whole. The only significant difference (as against topic 1) occurs in step 7. Here the precondition test for a topic i, $i > 1$, includes tests for all those concepts established in previous segments that are concepts for topics labelled by terminal nodes of the ith concept ordering.

This adaptive technique is a tutorial operation at *Lev 2*. It is a heuristic. Certain features of the heuristic are readily mechanised (for example, the numbering of the concept ordering) and others can be (for example, the housekeeping details of programme writing could be handled by co-operative interaction with a computer). But some steps, notably writing the frames and possibly finding a good breakdown into subtopics and certainly the clinical analysis, are in the human domain.

3.2 Criticisms of the Programming Scheme The tutorial programming scheme operates quite successfully and, judged by gross criteria, yields effective tutorial programme segments. It does not follow that combinations of these segments will be equally effective because no account is taken of 'long distance' interactions. Consequently when these are important (as they are in learning many subject matters) the result of segment combination is unpredictable.

The main criticisms of the scheme that emerged from using it are more fundamental. As a matter of fact, they apply to all of the schemes based on comparable premises but happen to be well exhibited in the context of this one.

(a) The tutorial programme is a teaching strategy of a specially restricted type. There is only one such teaching strategy and that one may be (empirically often is) out of kilter with the student's innate learning strategies. That is, a student is not, even at the strategic level, a *tabula rasa* open to the inscription of *any* plan for learning, however efficient it may be. Left without guidance he learns using his own strategies, which compete with *some* imposed by the teaching system though they interact cooperatively

with others. The competition can hinder the learning process, and, as judged from our records, may even altogether inhibit it.

(b) The topic ordering is a slightly liberalised form of noun tree; a 'taxonomic ordering' in a very narrow sense. It might be conveniently tidy if knowledge did have such a structure but (fortunately to my mind) there is no real reason to believe it does so, and every reason to believe it does not. It seems that knowledge is structured as a *verb network* or relational network not as a *noun tree*.[3] If so, only *descriptions* of knowledge are hierarchical. For example, though some of our students found it easy to follow the partial orderings of topics, or even the particular sequence delineated by the tutorial programme, others found the task quite impossible. They could only conceive 'truth tables', for example, in many ways simultaneously. To succeed in the learning task they would have needed a large number of (quasi-hierarchical) descriptions of the underlying cyclic structure.

(c) As noted already there is no easy way of evaluating explanations. The multiple choice and constructed response questions that are asked constitute, in the sense of Harrah (1966), *whether* or *which* questions. These can easily be machine evaluated, which is why they are asked in the first place. Our records strongly indicated that the selective replies to these questions (lists of possible selections of one answer from a set of alternatives) furnish inadequate indices of comprehension and that the position is not greatly improved by averaging over responses (it might be by considering patterns of response to an appropriately patterned collection of items).

On the other hand there are firm grounds, theoretical as well as empirical, for believing that a novel or invented correct explanation *does* furnish good evidence for the existence of a concept and that a derivation from, or explanation of, the original correct explanation is evidence for the existence of a concept in long-term storage (my current jargon is different; namely, that the explanation can be reconstructed as a *memory*, preferably in many ways).

3. Take a dictionary and follow through the definition of a *noun*: the successive definitions are nested in a *tree* (the most general the highest and the most specific the lowest). On the contrary the definitions of a *verb* form a network; any verb is defined in terms of other verbs and at some point the original verb is used in the definition of a derived verb. Thus; Push is to Shove *or* Move or . . .; Shove is to Thrust *or* Lean against *or* push . . . (and so on for all branches). The structure is a *verb network*.

11 Learning Strategies, Teaching Strategies, Matching and Mismatching

Learning depends upon the strategies used by a student in order to direct his attention and to partition the educational goal into separate subgoals. At one extreme, these strategies may stem from the student himself; they are *learning strategies* which he brings to bear upon an otherwise unstructured situation. At the other extreme, the strategies may be imposed upon the student as *teaching strategies* by a programme, a teaching device, or a training routine. Many real situations lie between these extremes. One of them is a tutorial conversation in which methods of learning are open to discussion and in which the strategy is selected as a result of a compromise between the student and the teacher.

Though strategies are important, it is also necessary to consider the student's competence in executing strategies of a given class. For example, an individual student may be good at 'seeing things as parts of a whole', or conversely, he may have a special aptitude for 'stringing subproblems into sequences', which (on resolution) lead to the solution of a large problem. There are, of course, many other types of competence; for example, competence demarcated along the visual/verbal dimension or determined by the relative efficiency of 'short-term' and 'working' memory.

Certain types of strategy call for certain types of competence. For instance, some strategies can best be (or ultimately can only be) executed by the 'holist' student, whereas the execution of other strategies rests upon a 'serialist' competence. By hypothesis, effective learning takes place if, and only if, the individually-selected (learning or teaching) strategy is matched to the student's existing competence.

Students are generally unaware of their competence, and a free learning student is unlikely to learn well. The combinatorial problem of mating the strategy choice to a (largely unknown) competence profile is considerable on its own, but there are also specific factors which militate against effective learning by positively encouraging mismatch. One of the most important is 'cognitive fixity', a tendency to adhere to an originally selected strategy even in the face of evidence showing it to be inappropriate.

The position is somewhat different if a teacher or a teaching programme

is introduced into the situation, because, so far as he (it) is obeyed, the teacher (programme or machine) usurps the role of the student's internal attention directing mechanism and guides his learning according to a teaching strategy. In the simplest case, the teaching system is designed on the basis of a single strategy. This is a provident design provided there are grounds for holding that a single strategy is, in some sense, apposite; for example, 'there is only one possible strategy' (rarely, if ever, true), 'one strategy is better than the others, i.e. there is an optimal strategy' (true in a few cases, note, for instance, Matheson (1964) on list learning), or 'one strategy is far more economically administered than the others, and individual differences have little influence upon the rate or effectiveness of learning' (quite often true and generally used to justify an instructional program that is neither branching nor adaptive).

Generally, however, such grounds for uniqueness do not exist, and in this case, the design of an effective teaching system must be founded upon a class of strategies. Sometimes, one member of this class can be chosen for use on a given occasion with a given individual by an adaptive procedure that employs feedback from performance measures, and the like. If so, the most important process in reaching such a decision is a matching process whereby the student's competence is weighed up and a strategy is selected to suit it. More often, the set of strategies is not given *a priori* and its members must be determined by a (human or mechanised) tutorial conversation of the sort alluded to earlier. In practice, also, there are other factors, to do with the regulation of uncertainty and the student's motivation, which often dictate the use of a conversational teaching system. But, within the conversational format, there must still be a tendency and even an authority to maintain matching. The compromise solutions accepted by the teaching system must be those that engender matching even if they do not completely secure it.

1 General Learning Situations

As a preliminary, it is useful to distinguish between performance strategies concerned with the execution of a skill and the learning (or teaching) strategies that build up performance strategies (Pask, 1969b). The arbitrary character of this distinction is stressed in Chapter 6, where it appears as a distinction between levels of control; namely *Lev 0* and *Lev 1*.

1.1 Performance Strategies Performance strategies have been studied with respect to a great many tasks. In the context of a perceptual motor skill, such as typewriting or vehicle control, the performance strategy is an operational or imperative interpretation of the hierarchical organisation in which subskills, associated with the achievement of subgoals, are integrated into the skill as a whole. Skill organisation, in particular the strategies involved in even the simplest of tasks such as compensatory tracking, is discussed in detail by various authors, for example, Pew (1966), Gaines (1968), Angel and Bekey (1968), and Pask *et al.* (1969b). Skill organisations differ appreciably between individuals, although detailed scrutiny reveals that the idiosyncratic performance strategies cluster around no more than four or five basic types.

Over a wide range of more intellectual tasks, performance strategies are manifest in the hierarchical organisation of problem solving procedures. This is clear, for example, in the protocols produced by the subjects of Newell, Shaw and Simon (1962) or Reitman (1965) (subjects who addressed themselves to logical demonstration; in a weak sense 'theorem proving').

Another instance is provided by the work of Bruner, Goodenow and Austin (1956) on concept acquisition. Their subjects adopted several strategies which were externalised, in the conditions of the experiment, as stretches of behaviour. Any strategy is a procedure for setting up and testing hypotheses about an unknown conjunctive category of the graphical exemplars in a finite universe of discourse, which is described in terms of four attributes (of the form 'shape of figure' or 'colour of figure'). Two main conditions were employed, receptive and selective. In the receptive condition, the experimenter has a conceptual category in mind (unknown to the subject) and he presents the subject with a sequence of exemplars (figures characterised by certain values of the descriptive attributes) furnishing, for each one, the information that it does or does not belong to the unknown class. From time to time, the subject submits an hypothesis about the unknown class and, if necessary, is corrected. In the selective condition, the subject (rather than the experimenter) has control over the evidence in so far as he is able to select exemplars which are submitted for tests of class membership. Subjects run in the receptive condition generally adopted one of two strategies, 'holist' and 'partist'; those run in the selective condition adopted one of four; 'successive scanning', 'part scanning', 'focusing' and 'focus gambling'. The nature of these strategies is detailed in the original work; the immediately important point is that though different individuals set about solving the problem in different ways, the variation is, in practice, quite limited. Although repeated performance of the task engenders a few novel strategies, most, if not all, of these are mixes of the basic ingredients noted above (Lewis and Pask, 1964).

As a final example, strategies occur at a level of cognitive activity which is generally deemed innovative or, at least, insightful. These strategies have been studied by Elshout and Elshout (1969), using the apparatus test as the experimental task (the subject is asked to provide two improved versions of

a given piece of 'apparatus'; to cite one of the author's examples, a 'chair'). Two strategies were distinguished: 'successive transformation' and 'locating problems'. Using the first, the subject reduces the problem of finding an improvement into (a) the subproblem of finding some property of the apparatus that needs improvement (usually culled from his own experience), and (b) the subproblem of transforming this property to yield an improved apparatus. Using the locating problems strategy, the subject (a) considers the set of properties of the given apparatus and selects one (e.g. the chair 'stands on the floor'); (b) replaces this by an imagined related property (another value of the same attribute, e.g. the chair 'lies on the floor'); (c) finds a problem encountered in connection with the original apparatus (a chair that stands on the floor) that would be solved if the apparatus were modified (i.e. if chairs in fact, lay on the floor), and (d) solves this problem by finding an innovation (some sort of collapsible chair, e.g. a deckchair) that can both lie on and stand on the floor.

Broadly, therefore, performance strategies characterise a very wide spectrum of the mental processes, which can be captured and objectified in laboratory studies. There is, of course, abundant discursive evidence that strategies are involved in all sorts of cognition and recollection (for example, in answering questions about a certain period of history or a certain field of mathematics).

1.2 Learning Strategies Given a realistically sized task (and assuming that he cannot already perform it), a student is unable to generate the required performance strategy all at once. Instead, he directs his attention to various facets or subtasks and musters subroutines that build up a performance strategy bit by bit. The process is carried out according to a learning strategy which, in the free learning subject, may be innate or acquired and which, for the student, is imposed externally by a teacher or a teaching system.

A learning strategy is comparable in kind with a performance strategy. Each sort of strategy entails decomposing goals into subgoals and applying mental subroutines to achieve the subgoals concerned. The necessary difference between learning strategies and performance is in the domain upon which they operate. Whereas the performance strategy solves problems posed by states of the (usually symbolic) environment, the learning strategy solves the problems posed (in the context of a goal like 'learn to solve apparatus test problems') by deficiencies in the current repertoire of relevant performance strategies; the solutions produced by a learning strategy are performance strategies.

In unmechanised and relatively uncontrolled situations there is naturally some confusion between learning and performance strategies, but the ambiguity disappears if the task is well defined and the extent to which learning is externalised in behaviour is fully stipulated. For example, are the concept acquisition strategies of Bruner, Goodenow and Austin (1956) one sort or the other? Well, it depends. If, as pictured in the last section, the subject is solving a problem, posed by the experimenter ('describe the unknown class') then the strategy is a performance strategy (as maintained). But the concept acquired in the process is not just a class description; it may be regarded as a procedure (in fact, just another name for 'performance strategy') designed to recognise members of this class. If so, then the Bruner, Goodenow and Austin strategies are learning strategies and the conduct of the experiment externalises a learning process. For instance, in some of the small group experiments carried out in my own laboratory (Lewis and Pask, 1964) subjects were required to use the concepts they attained; in this context the Bruner, Goodenow and Austin strategies are unequivocally learning strategies. Likewise, in the simulation of this situation by Hunt, Marin and Stone (1966), the 'artificial intelligence' program represents the student using (extended versions of) the Bruner, Goodenow and Austin strategies, as the learning strategy whereby it constructs a concept of the form 'performance strategy'.

So far as the other examples from the last section are concerned, there is little room for doubt about a reasonable interpretation. Without diffidence, the action of a typist in tackling subsets of the keyboard separately is assigned to the class of a learning strategy which builds up the motor programs (performance strategies) required in order to perform the skill. Nor is it too difficult to trace the modification of these programs as the skills develop (Pask *et al.*, 1969a). Again we have no hesitation in distinguishing the organised acquisition or inculcation of the 'successive transformation' or the 'locating problems' strategy from the performance itself.

In conclusion, the first aspect of a learning strategy is a contingent plan (i.e. the plan may depend upon indices of success, or the like) for selecting a field of attention. Thereby, the student directs his attention to different parts of the task, to different subgoals or subproblems. In this definition, the field of attention is the domain of some performance strategy and it may either be part of the external environment or an internal representation of it; the latter possibility is pertinent when learning involves rumination, covert rehearsal, and other processes which have no direct *behavioural* correlates. My early work is primarily concerned with situations in which precautions have been taken to ensure that the domain in question is related to the external environment, so that attention directing can be objectively scrutinised. But this condition is not required by the basic statement. Generally, the performance strategies, whose domains are selected by a learning

strategy, are incomplete or even embryonic entities (figuratively, 'boxes waiting to be filled in'). If so, then the learning strategy musters operations that act upon the form of the performance strategy; operations that remedy its defects or, in the limit, that construct it. This is the second aspect of a learning strategy. So, by way of a summary, a learning strategy is first of all a contingent plan for selecting performance strategy domains (fields of attention) and secondly a plan for building these strategies or for repairing them.

1.3 Individual Competence to Execute Strategies Learning strategies call for the execution of mental subroutines which are relatively permanent features of the mind; for example, the subroutines involved in abstraction, in concatenation or 'stringing together', in substitution (of one operation by another), and so on. By the same token, performance strategies make use of several relatively permanent subroutines, notably those that organise the short-term and working storage of the brain into coherent systems (as shown, for example, by Atkinson and Shiffrin, 1967). There is ample evidence that the efficiency of different subroutines, at either level, varies a great deal from person to person, perhaps also from day to day. The distribution of efficiency evaluations over the subroutines is the subject's competence (or competence profile).

A subject's competence is determined by presenting him with paradigm problems of a given type and finding how efficiently he solves them. There is nothing new in this. Multiple aptitude and ability tests provide the requisite data so far as performance strategies are concerned, and since it is maintained that there is no necessary distinction in kind (only in domain) between performance and learning strategies, this data is just as good with respect to the 'higher level' operations. For example, the styles or dispositions observed by Kagan (impulsive/reflective) or Witkin (field dependent/field independent); by Bruner, Luria and the Piaget school, who distinguish certain dominant or preferred modes of problem solving.

Nor is the interpretation of ability test results in terms of a competence profile altogether original. The 'structure of intellect' model of Guilford (1956) is proposed in a similar spirit. But Guilford has an essentially descriptive approach; he maintains that mental factors, by the factor analysis of test batteries, can be categorised and refined in several ways to yield the familiar three-dimensional figure with operations (evaluation, convergent thinking, divergent thinking, memory, cognition) along one side, with contents (figural, symbolic, semantic) along the next, and with products (units, classes, relations, systems and implications) along the last. It is possible to obtain specific estimates of differential competence in respect to many of the ninety cells so distinguished. In contrast, our own theory is process orientated and, consequently, words such as 'operation' have a different meaning. Any strategy is made up of operations which are akin to the TOTE units[1] of Miller, Galanter and Pribram (1960), and which are assembled into hierarchical or interactive structures. Such a structure is a possible strategy and, when executed, it gives rise to a process. The most elementary operations are thus of the same type and the subroutines made up from them differ from one another in respect to context; i.e. their differences rest upon the operations or processes upon which they depend.

In general, ability tests may be expected to sample the efficiency of common subroutines. But any 'operation' (in the present sense) entails entries, in many of Guilford's cells; indeed, it involves most if not all of Guilford's operations. The multiple reference is neither surprising nor disturbing but it does highlight the question of whether Guilford's taxonomy is appropriate for a process model. It is also pertinent to remark that the currently available ability tests are far from optimal devices for sampling the basic competence data when a process model is in mind. For example, there is no test that distinguishes between people's ability to build up association-discrimination processes and their ability to construct rule-like procedures (which is an outstanding distinction from a process point of view).

1.4 Matching between Competence and Strategy The criterion of matching employed in experimental work is based upon specific tests for an individual subject's competence in executing particular operations or learning about a miniature universe of knowledge (under conditions that are designed to make his approach to the matter objectively recordable). Tests of either kind can be carried out in the course of an experiment but are usefully buttressed (they might be replaced) by an appropriate mix of mental test-data. The result, after analysis, is a competence profile which can be compared with each of the (rather few) strategy classes that actually exist. A matching index is obtained by correlating the measure on each proficiency index of the competence profile with the occurrence of each corresponding operation in the strategy or (in case the subject has learned about the miniature universe) by direct pattern superposition. In particular, if a definite strategy is actually adopted by a subject for a learning task, it can be assessed as more or less matched to the competence profile of the individual subject who adopts it.

1. As in the last chapter. However, I am giving a much broader connotation than usual to the familiar notion of a TOTE. It is crucial that the test and the operations are usually non-deterministic programmes or fuzzy algorithms (rather than serial programmes equivalent to finite automata).

1.5 Misperception of Competence and Fixity In free learning conditions most subjects are bad at selecting matched strategies. As a result, they learn slowly if at all (Pask and Scott, 1971; Pask, 1969b). To some extent this may be due to ignorance of the strategies available, and it is certainly true that the imposition of any well defined strategy (for example, via a teaching programme) generally enhances learning efficiency.

But, on the other side of the coin, students commonly bring preconceived notions to bear upon the process of learning and these notions are the germs of strategies, even if they cannot be verbalised as plans. Generally speaking, any imposed strategy is in competition with strategies that already exist. Hence, in a teaching system, it is essential to discover what the competing strategies are by allowing the subject to assert them. This is one reason why the system is one in which the subject is encouraged to talk about the learning process.

Moreover, even if the student is fully aware of the strategic possibilities (for example, as a result of a preliminary discussion), he is still prone to adopt the 'wrong' strategy. An important factor in the decision process is undoubtedly the student's misperception of his own characteristics. On the whole, students are unaware of their competence and the situation is improved if appropriate competence-evaluating information is provided. But misperception is not the only factor. When given the requisite information, individuals are still apt to disregard it, or (putting the matter differently) they need to accumulate much evidence before they will autonomously change their minds and adopt a more appropriate strategy than the one they chose originally. This tendency is commonly manifest; as before, it is called 'cognitive fixity' since in many ways it resembles the 'cognitive dissonance' of Festinger (1957), i.e. a strong tendency to reject or pervert as affirming, any information that disconfirms an hypothesis which the individual has decided to adopt and in which he has invested resources or effort. For example, the purchaser of an expensive motor car is liable to reject evidence that it is in fact unsuited to his needs. Cognitive dissonance is a special case of cognitive fixity. To minimise cognitive fixity, it is thus not only necessary to provide competence evaluations but also to actively dissuade the subject from pursuing misguided lines of action. This is the other reason why a conversational design is often mandatory.

There is often a lot of cognitive fixity over writing a paper, for instance. The author starts with a goal of sorts and realises that the paper must satisfy limits of length and comprehensibility and must put across certain salient ideas. The goal generates a strategy; the paper has a general form and its parts are tackled in a given order. The fixity occurs as follows:

At one stage in writing the paper a great deal of effort is invested in a section developing, for example, the theoretical underpinning of an argument. This, of course, constrains the form and content of the rest of the paper and, if the theoretical argument happens to be inappropriate (not necessarily false), makes it virtually impossible to present an understandable case. Nevertheless, the effort has been spent and the author likes the style of the section. In fact, the paper gets nowhere (and is probably unwriteable in principle) until the author eventually collects sufficient evidence to make him expurgate the offending section. Moreover, a sort of trapping state is involved. The longer that the inappropriate strategy persists (in this instance the strategy engendered by the theoretical section); the more it colours the rest of the structure and the more difficult it becomes to dislodge. As a final point, cognitive fixity is operative even though the author knows all about it, and the snares in which he may be caught. The fact is, human beings are not very good at self-observation or self-control and cognitive fixity (which is perhaps the major obstacle in the path of effective learning) can only be reduced by an outside influence.

2 Specific Investigations (Code-Rule Learning)

Due to the complexity of real-life learning situations, it is quite difficult to obtain individualised data of a type that will illuminate these issues. The learning strategies germane to geography, history, or chemistry may extend over days or even weeks and regularities are obscured by interruption and extraneous activity. More definitive results stem from laboratory studies. On the other hand, brief and small-scale learning experiments do not allow for strategic diversity and a special choice of task is called for.

2.1 Suitable Experimental Tasks

1. The task is learnable but learning is a lengthy business (5–7 hours is perfectly practicable).

2. The problems and subproblems belong to well defined classes so that the experimenter and the subject are in agreement regarding which class a problem belongs to.

3. Although the subject is given the goal of learning to solve full problems (of the main class) these are far too difficult for the novice to tackle *in toto*. Thus, any subject is forced to adopt a strategy of partitioning the task by learning to deal with subproblem classes one after another (eventually becoming proficient with respect to the full problems).

4. Problem solution is rapid so that the experimenter can present the subject with representative sequences of problems drawn from each class.

5. It is possible to evaluate the subject's performance and to offer him indices of success and goal approximation.

6. The learning strategy (which any subject is forced to adopt, as in (3) above) can be externalised as a stretch of behaviour. It is convenient to concentrate upon the attention-directing aspect of a learning strategy, and to regard the strategy as externalised in so far as the subject attends to a sequence of subproblem classes, which terminate in the class of full problems.

7. The subject can be given reasons for externalising a process which is, from his point of view, more conveniently carried out in his head (when, of course, it is hidden from the experimenter). For example, he may be asked to physically select a class of subproblems by pushing buttons on a console, as a result of which a sequence of problems drawn from this class is presented to him. If so, the subject *could* partition the full problems mentally and solve part of each full problem in his head. To induce him to make objectively-detectable selections in a veridical fashion (rather than the capricious manner he might adopt following an arbitary fiat), the following expedients are used.

(a) The subject agrees to maintain at least a certain average score, over and above agreeing to aim for the educational goal. A score is computed with respect to the selected class only (note, if the novice selected the full problems the score would be so low that the subject could not keep his agreement).

(b) Knowledge of results is provided in respect to the selected class (note, this gives the student detailed as against macroscopic information).

(c) Facilities, such as evaluative data, are provided in the framework of the classes so far selected.

Externalising gambits like (a), (b) and (c) constitute a primitive form of cooperative externalisation technique or CET (Pask, 1969b; Pask and Scott, 1970, 1972b).

8. It is possible to determine the subject's competence by tests that are relevant to the task. Usually behavioural tests are called for; paper and pencil ability tests have been used but they are insufficiently specific and are inconvenient for on-line administration.

9. It is possible to ascertain the subject's performance strategy. Once again, behavioural measurements are preferred.

10. The investigation is greatly simplified if one learning strategy is used to build up just one performance strategy.

11. The task is related to a useful category of real-life tasks.

2.2 Free Learning Experiments Experiments have been carried out in a situation that approximates conditions (1)–(11) and is a microcosm embodying certain aspects of real-life relation learning. The task employed is an elaboration of the code learning tasks of Chapters 6 (simulation), 7 and

8. The display and response facilities are shown in Fig. 96. A full account of the experiments is given in Pask and Lewis (1968), Pask (1969b), and Pask and Scott (1971).

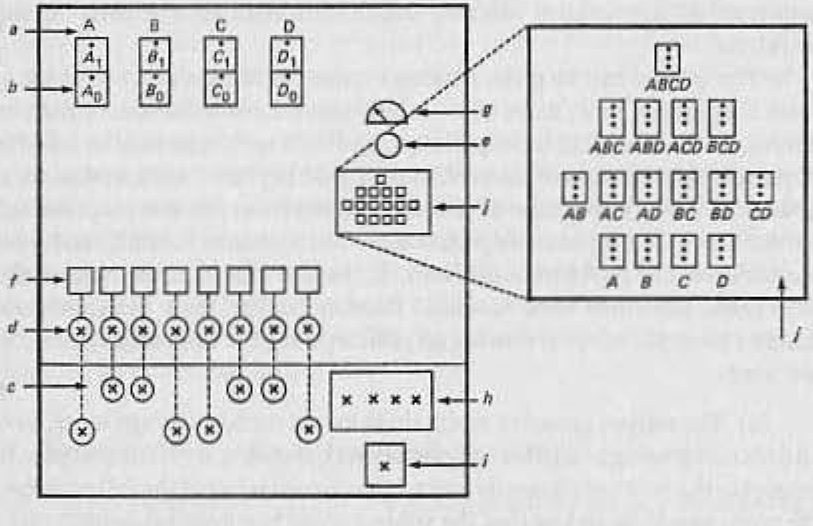


Figure 96 Display and response board: (*a*) signal variable names, *A*, *B*, *C*, *D*, one for each box; (*b*) problem lamps, A_1, A_0; B_1, B_0; C_1, C_0; D_1, D_0; in boxes. The lamp in the upper box indicates the value 1 of an attribute, and the lamp in the lower box, the value 0. Problems, *x*, are configurations of illuminated lamps (see text); (*c*) eight response keys, $y_1 \ldots y_8$; (*d*) external register, retaining key-pressed information until the end of any trial; (*e*) partial knowledge of results lamp (complete response is or is not correct, σ_0); (*f*) complete knowledge of results lamps indicating correct value for each component of the response; (*g*) score meter; (*h*) attribute selection buttons; (*i*), submit button (see text); (*j*) last value of score display. Three lamps for each subset: blue (upper), $\rho > \rho_2$; green (middle), $\rho_2 \geqslant \rho \geqslant \rho_0$; and red (lower), $\rho_0 > \rho$.

In the full problem situation (which must ultimately be mastered) the subject is presented with a sequence of four, variable, visual signals and must solve each problem by making an appropriately coded four-component response. Calling the signal variables *A*, *B*, *C* and *D*, the full problems belong to a class named (*ABCD*), i.e. its members present all of the variables at once. Under the temporal constraints imposed upon the task, a novice is quite unable to learn how to deal with problems in (*ABCD*) and is thus impelled to direct his attention to subproblem classes of the form (*A*), (*B*), . . ., (*AB*), (*BC*), . . ., (*ABC*), (*ABD*) . . . (there being fifteen classes in all). The structure of the problem environment is shown in Fig. 97. Clearly, the

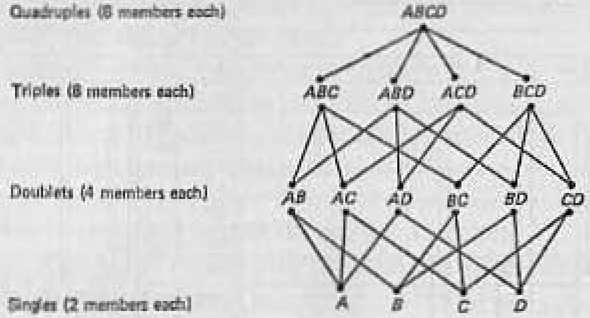


Figure 97 Structure of the problem environment.

terminal task is isomorphic with the terminal task for code learning described in Chapter 8, section 1, only, in this case, the environment has been partitioned so that there are many visually distinct ways of directing attention to different subproblems. The experiment is fully automated and the subject is asked to externalise his attention-directing strategy at the end of each block of trials (each sequence of problems) by keying into his console the name of the subproblem class which he will consider during the subsequent block. This selection determines the subproblem class from which stimuli (designating the problem) are actually drawn. A CET involving the provision of scores, and the evaluations and knowledge of results is used to secure reliable externalisation.[2] In these conditions a learning strategy is manifest; (a) As a sequence of subproblem class selections, and (b) Through information obtained, on subsequent interrogation, about how these selections depended upon the evaluation and scores.

Preliminary studies revealed the existence of four types of learning strategy (Fig. 98).

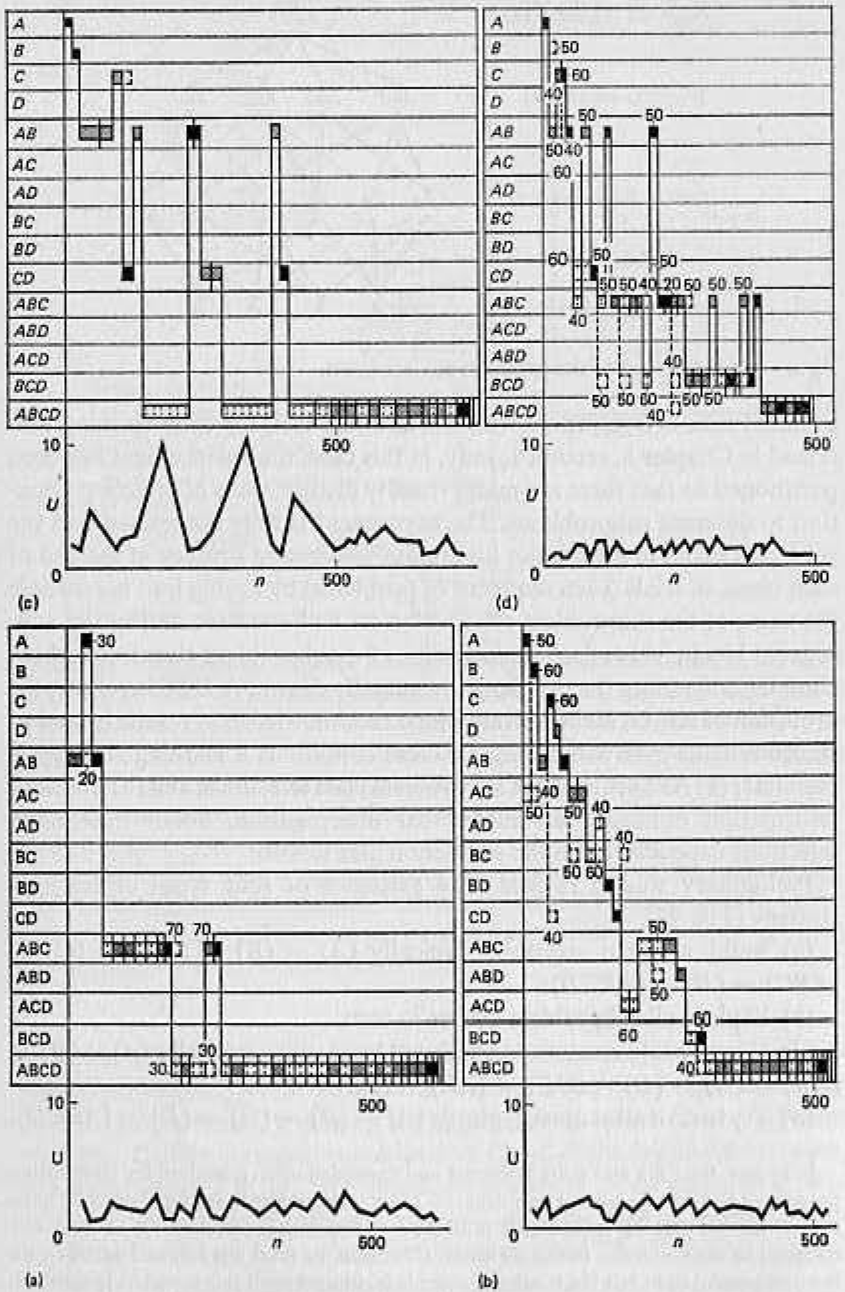
(*a*) Build up in a sequence, typically (*A*) → (*B*) → (*AB*) → (*C*) → (*ABC*) → (*D*) → (*ABCD*).

(*b*) Explore all subproblem classes in turn.

(*c*) Group problems into increasingly large units, typically (*A*) → (*B*) → (*AB*) → (*C*) → (*D*) → (*CD*) → (*AB*) → (*ABCD*).

(*d*) Try to do it all at once, typically (*A*) → (*B*) → (*C*) → (*D*) → (*ABCD*).

2. In fact, the CET was quite elaborate and depended upon a method for committing the subject to a certain level of performance. On entering a block, say (*AB*) or (*ABC*), the subject knows that he cannot exit until he has responded correctly on at least one occasion to each stimulus in the sequence. Of course, he need not respond correctly on the first presentation but the block sequence is continued until this criterion is satisfied. The score is the number of 'first presentation correct responses' (divided by the number of different stimuli). From the number of repetitions or guesses we derive a direct measure of the subject's uncertainty (i.e. Figures 98, 101, and 102).



One variant c^* of the type (*c*) strategy is noted in the source paper Pask and Scott (1971) but will not be discussed.

Of these possibilities, the performance strategy that goes with learning strategy type (*d*) leads to triviality in so far as the subject who uses it is failing to solve the entire problem. In contrast, he deals, at his leisure, with as many separate items as there are stimulus variables (it is doubtful whether, in this case, his behaviour is a problem solving behaviour). Hence, a time constraint was imposed to prohibit the offending performance strategy. In fact, the time allowed for solving each problem was separately adjusted for each individual (in a preliminary adaptively controlled session), so that no more than a pair of elementary operations could be carried out at any one trial. As a result, subjects were told and knew that learning strategy (*d*) was bound to be abortive. Moreover, they experienced this fact since before the main experiment (but after the adaptive period) each subject had practice in learning the skill with a different coding rule in force.

Two types of competence were observed as response processes, manifest in regular patterns amongst latencies of the several components of the complex response; (a) a serial competence which at the microscopic level is called stringing (of response producing operations) and (b) a holistic competence, interpreted as a grouping of operations.

The main findings of the free-learning experiments were that though a few subjects learned the skill successfully (in 400 trials or so) the great majority did not (some taking 750–1000 trials and a few hardly learning it at all). Those who failed did so for quite definite reasons.

1. The subject tries to use strategy (*d*) even though he knows that it cannot succeed. The lure of getting straight to the goal is remarkably seductive. Eventually, strategy (*d*) is abandoned in favour of some other plan and the student takes about as long to learn the skill as he would have done starting out this way.

2. Subjects who are good at stringing but bad at grouping try to use strategy (*c*) (which is fitted to grouping). Their learning is retarded and occasionally they are forced to change strategy.

Figure 98 Typical free-learning curves. Upper part of figure: problem class selections along the vertical coordinate; *n* (number of trials), along the horizontal. Proficiency, ρ, is shown by shading for each block of trials:

■ $\rho \geqslant 0{\cdot}75$; ▨ $0{\cdot}75 > \rho > 0{\cdot}25$; □ $0{\cdot}25 \geqslant \rho$.

Probabilistic problem class selections are shown as *P* numbers associated with bifurcating lines. Lower part of figure: *U* (behavioural uncertainty) along vertical coordinate, *n* along horizontal. (*a*) type *A* subject; (*b*) type *B* subject; (*c*) type *C* subject (also demonstrating unsuccessful attempts to use strategy *D*); (*d*) type C^* subject.

3. Subjects who are good at grouping but bad at stringing try to use strategy (*a*), which is adapted to stringing. Their learning is retarded and occasionally they are forced to change strategy.

4. Subjects adopt strategy (*b*). In most cases, it would be more accurate to say that these subjects have no global strategy. (Certain exceptions exist.) In later experiments where pre-experimental and post-experimental protocols were taken, one subject did express the intention of exploring and produced type (*b*) selections. But most of the apparent explorers turn out to be people who are in doubt about what to do. This contention is supported by subjective confidence estimates over the set of subproblem classes, which were obtained after each block.

In all, twenty-six subjects were run in the free-learning situation. Ten of these were run in conditions which are strictly comparable with those of subsequent experiments, and for this subgroup the mean number of trials to criterion was 574·5, σ (standard deviation) = 153.

2.3 Adaptive Machines The most obvious way to remedy some of these defects is to impose a teaching strategy upon the subject. Since there is great individual variation in how long it takes to achieve criterial proficiency, teaching must be a score sensitive procedure mediated by an adaptive machine, which like the controllers described in Chapter 8 was programmed on the special purpose computer shown in Plate 5. The machine is designed on the basis of just one strategy which is incorporated into the machine design in the same way as the skill structure of Chapter 8, and the machine (rather than the subject) makes all subproblem class selections.

Learning of the code learning and signal translation skills has been extensively computer simulated (see Chaper 6) and there are good grounds for supposing that strategy (*c*) is generally the best. It is not the best strategy for each individual, of course; strategy (*a*) is better for someone with a proclivity for stringing. But on average it is as good as any other single alternative and, in order to obtain a standard condition for comparison and reference, the experimental machine was based upon a type (*c*) strategy. The flow-chart of the adaptive system is shown in Fig. 99.

The subject is advanced to the next selection in the strategy if his score exceeds an upper threshold 'sent back' if his score is less than a lower threshold and 'left' where he is 'if his score lies between these limits'.

The results obtained from the single strategy adaptive experiments are quite unequivocal. This type of instruction is faster than free-learning. (Mean trials to criteria = 463, σ = 167, N = 10.)

Learning rate (adaptive) > learning rate (free-learning); the difference is significant at the 0·1 per cent level. Care was taken to check that the

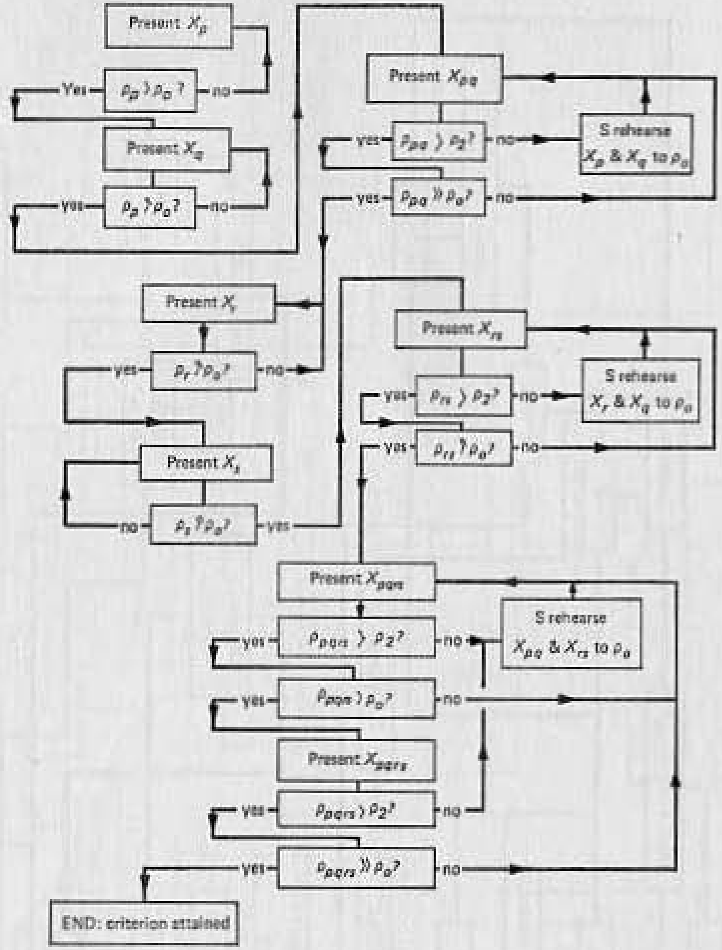
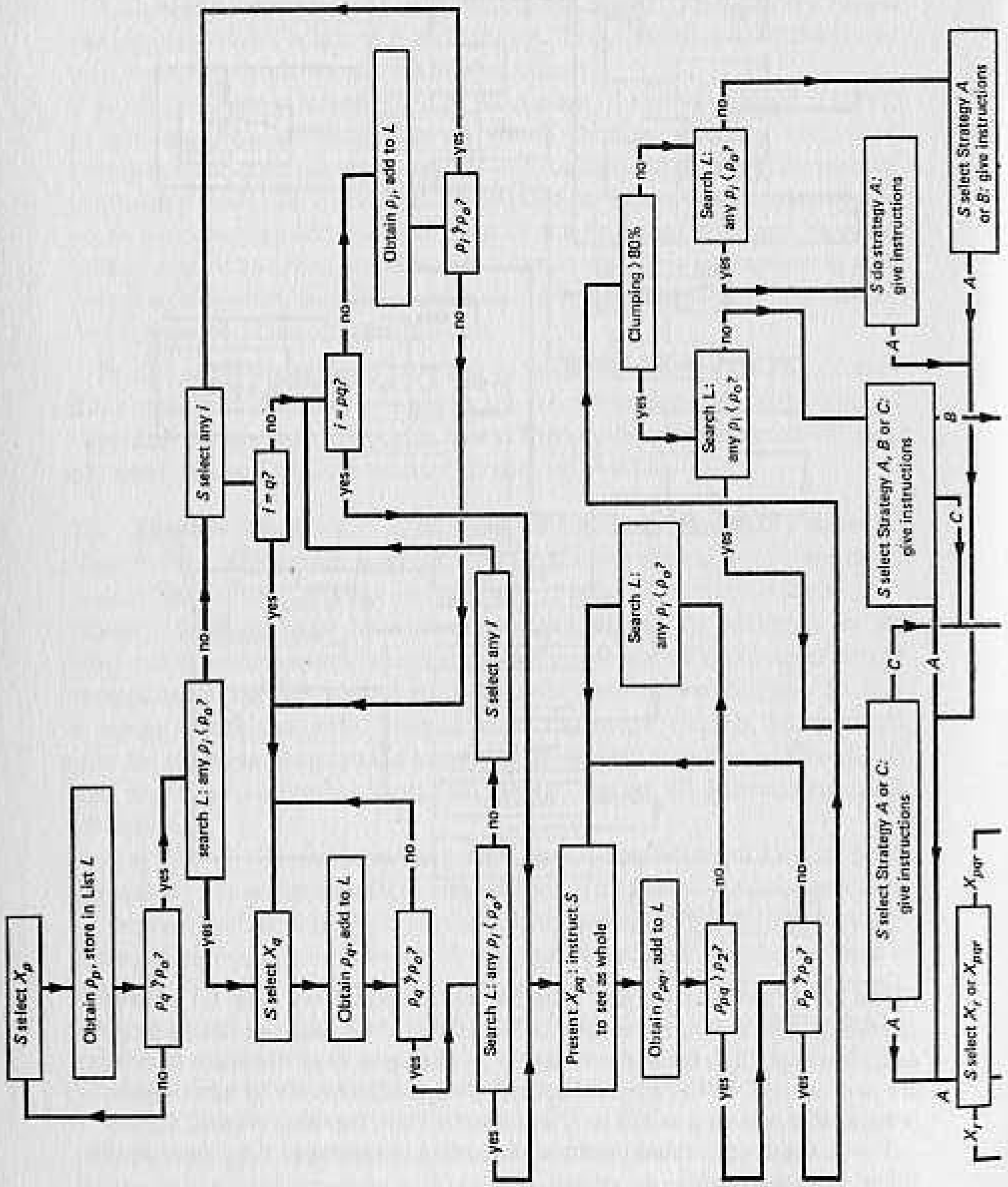


Figure 99 Flow-chart for the adaptive system based on type (*c*) strategy. *Explanation.* The X_i are the subproblem classes. The subscript values p, q, r, and s correspond to the signal variables, A, B, C, and D in the order that they are *first selected*. In the adaptive system, the machine makes all selections and is set so that $p = A, q = B, r = C$ and $s = D$. Thus, e.g. $X_{pqr} = (ABC)$.

The ρ_i are scores of first presentation correct responses to the signals in the subproblem class X_i.

ρ_0 and ρ_2 are, respectively, the upper and lower thresholds, where ρ_0 = 75 per cent, and ρ_2 = 25 per cent first presentation correct responses.

All commands are to the adaptive machine except for those prefaced by '*S*' which are commands to the subject.



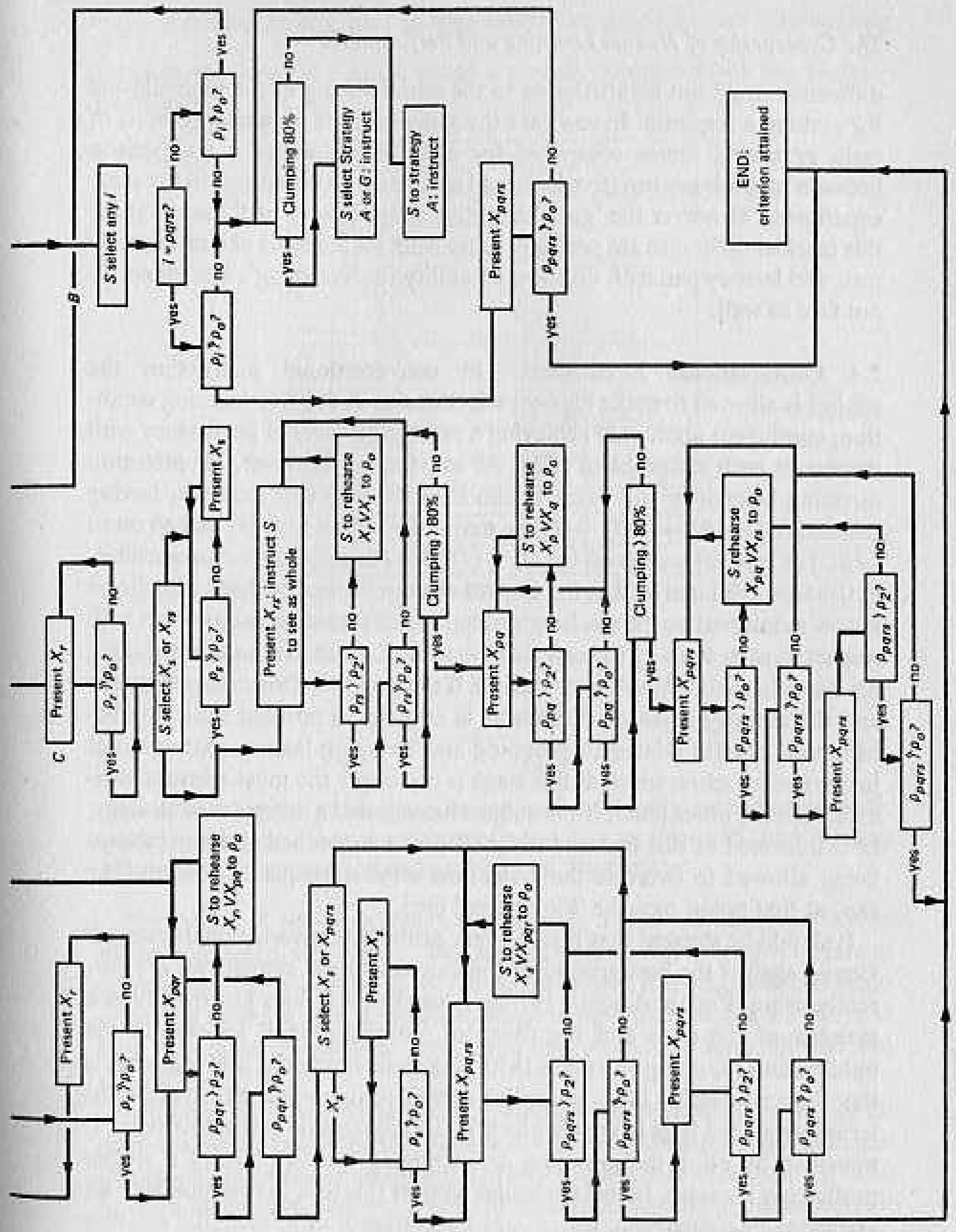


Figure 100 Flow-chart for the conversational system. *Explanation.* As for Figure 99 with one addition: the test of a subject's ability to make grouped responses ('clumps') is represented by the test box 'Clumps 80 per cent,' i.e. to pass the test at least 80 per cent of a subject's 'first presentation' responses must be correct *and* grouped.

difference could not be attributed to the initial setting of the time allowed for making a response. In any case the same result is obtained in terms of ratio or saving scores computed for each individual as a comparison between preliminary run (free learning) and main run (adaptive) in the same experiment. However the 'good' adaptive subjects who really account for this marked difference are precisely those who, on the basis of continuously recorded latency patterns, do show an ability for 'grouping'. The others do not fare so well.

2.4 Conversational Instruction In conversational instruction the subject is allowed to make his own selections as in the free-learning condition, contingent upon only achieving a reasonable level of proficiency with respect to each subproblem class. At some stage, however, his attention directing behaviour will commit him to a strategy (for example, having selected $(A)\rightarrow(B)\rightarrow(AB)\rightarrow(C)$ he may either adopt type (a) and go on to (ABC) or type (c) and go on to $(D)\rightarrow(DC)$. At this point the strategic alternatives are exhibited so that the student can opt for one of them. His choice is now monitored by the teaching system, which assess his competence with respect to each strategy (by carrying out a 'look ahead' test and determining his disposition to 'group' or to 'string'). If the subject's choice is appropriate and the strategy is matched, then he is allowed to proceed satisfying behavioural criteria related to grouping and stringing latency patterns (not just to overall score, which at this stage is no longer the most relevant variable). On the other hand, if the subject has selected a mismatched strategy, he is informed of this fact and diverted on to a matched strategy (always being allowed to override the system on some subsequent occasion if he can, at that point, pass the 'look ahead' test).

It should be stressed that this is a very primitive conversational system if viewed against the background of the conversational systems we are currently using (Pask and Scott, 1972b), nevertheless it does just qualify as a member of the class and the usage of 'conversational' appears in the publications for our group up to 1970. The main deficiency, a serious one, is that the conversation lacks an explicit criteria for *understanding* (a technical term introduced later in the chapter). To an appreciable extent this defect is remedied by using the grouping and stringing test in place of a simple proficiency measure. In this particular context this test, in common with an understanding condition, guarantees the existence of a 'memory' (section 2.6).

The design of a conversational system is based on a class of strategies (Type a, b, c) and not just one strategy. Its flow-chart is shown in Fig. 100 and allows for all sorts of remission and recycling which may complicate the picture sketched out above. Typical subject records for this type of

learning are shown in Fig. 101 which is usefully compared with Fig. 98 (free-learning).

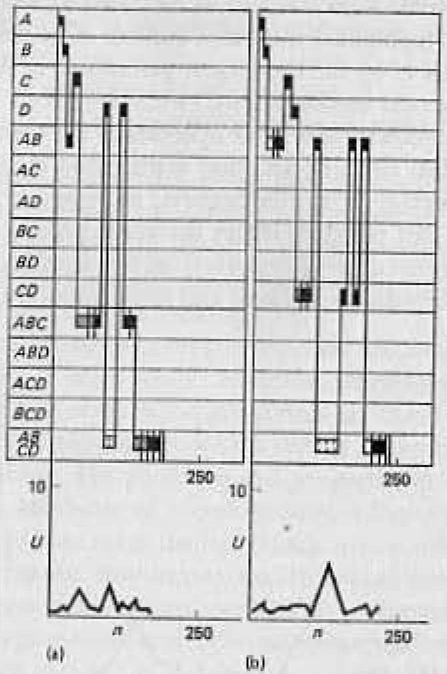


Figure 101 Typical learning curves for conversationally controlled subjects: (a) type-a strategy; (b) type-c strategy (in notation as Figure 98, each strategy adopted as a compromise between subject and the machine).

The results of the experiments demonstrate a clear advantage in favour of the conversational system. The results were as follows. Mean trials to criterion $= 220$, $\sigma = 40{\cdot}6$, $N = 10$; the difference, learning rate (conversational) $>$ learning rate (adaptive system) is significant at the 0·1 per cent level (the same precautions were taken with respect to time allowed and ratio scores).

Although the type (b) (exploratory) strategy was offered as an option, no conversational subject chose it and the confidence-estimation data indicates that this was so because no subject had any doubt about what he ought to do (though all subjects had the liberty to choose selections). By definition all strategies were matched to the subjects competence so, not surprisingly, 'grouping' subjects followed a type (c) plan and 'stringing' subjects a type (a).

The most dramatic difference between the free-learning and the conversational subjects is in the elimination of cognitive fixity, especially with respect to the type (*d*) strategy. On average the free-learning subjects spent 39·8 per cent of their subproblem-selections rehearsing problems in class (*ABCD*), and most of the trials were abortive as indicated by the high uncertainty over these blocks (computed from the 'number of guesses' data provided by the CET). This is by far the largest percentage of selections for any problem class; the next largest is (*ABC*) with 15·6 per cent. In contrast, the conversational subjects spent only 25·2 per cent of their selections rehearsing problems from class (*ABCD*) and they experienced relatively little uncertainty. This percentage is not the highest; more selections were spent rehearsing (*AB*) at 26·1 per cent. Hence the whole pattern of learning in the conversational system differs markedly from free-learning. In particular the students uncertainty decreases faster and more smoothly (Fig. 102).

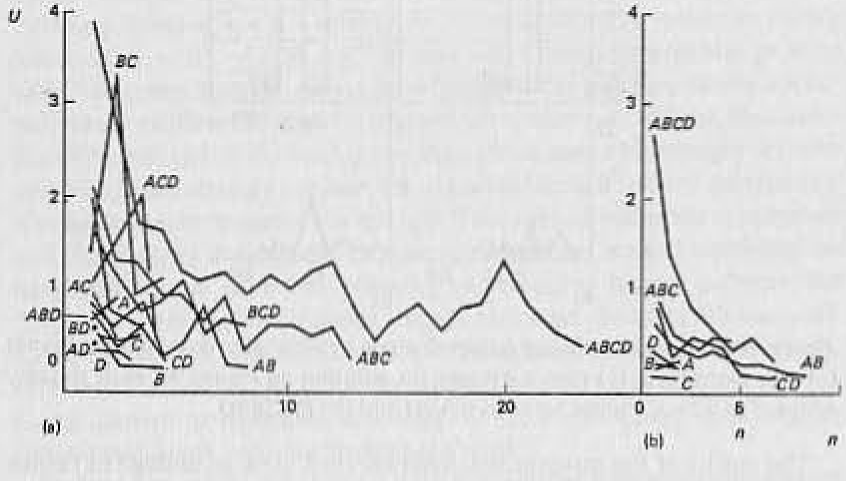


Figure 102 Plot of *U* (behavioural uncertainty), averaged over: (*a*) free-learning subjects; (*b*) conversational subjects with respect to separate problem classes plotted against number of trial blocks (*n*) spent dealing with each class.

2.5 Other Tasks, Isomorphism Between the Perceptual Motor and the Cognitive Domain The experimental desiderata of section 2.1 are satisfied by certain intellectual tasks; notably learning maps and layout schemes, moderately sized taxonomies, the operation of mechanical and biological systems (including cyclic processes such as the mammalian oestrous cycle and the operon/repressor control system). If the proper conditions are satisfied, then the distinctions manifest at the perceptual motor level of activity are recapitulated in the cognitive or intellectual domain; especially.

(a) Cognitive fixity.

(b) The existence of strategy types that are exclusive with respect to *one* experiment.

(c) The existence of a differential competence to execute learning strategies of different types.

(d) The effect of matching and mismatching.

(e) Interference and incompatibility of certain mental operations.

In fact, as hinted in section 1, it has been shown that these and other processes are characteristic of learning and cognition of a much more general kind; for example, in learning academic subjects; in design, in creative and innovative activity, and in the learning that accompanies and forms part of real-life decision making.

Hence it is argued, firstly, that certain processes are common to perceptual motor skill and to a small scale intellectual learning provided that the 'proper conditions' of observation are preserved. These conditions are, in fact, a conversational situation which may either approximate free learning or a type of teaching. The processes and properties in question are thus, strictly speaking, attributes of a conversation rather than an individual though clearly they bear upon the individuals who converse. In respect to these qualities (processes, distinctions, etc.) there is an *isomorphism* between the perceptual motor field and the cognitive or intellectual field.

Secondly, it is argued that the same isomorphism has a much wider scope; spanning education for certain, and perhaps extending further.

The second contention (like the first) has been empirically substantiated but it is necessary to look at the nature of the 'proper conditions' with some care, i.e. to spell out in detail how a 'conversation' is interpreted in this context and to surmount certain difficulties that become clear on examining the intermediary and laboratory sized learning situations.

2.6 Explanation Prior to considering the intermediate (small, intellectual learning) experiments it is essential to give a better account of understanding. In a perceptual motor skill, taught conversationally, the criterion for understanding is an ability to *reconstruct* whatever problem solving or goal directed procedure is employed to *construct* a complex response. Within the intellectual domain (for the moment, also, verbal conversations) the goal directed procedure is equated, as in the last chapter, with a *concept* which brings about or satisfies a relation *R*. The evidence for the existence of a *concept* of *R* is a verbal *explanation* of the relation in question.

Again, following the last chapter, suppose that the concept is assigned to a level of control, *Lev 0*. The entity which reconstructs a *concept* of *R* is a special case of an entity which constructs it in the first place; namely a *Lev 1* goal directed procedure. It might also be called a concept (for the distinction of levels is a matter of convenience in description) but to maintain this convenience and to follow ordinary usage[3] it will be called a *memory* of R; (that is, a constructive operation that recapitulates, reconstructs, or relearns a *concept* of R). The evidence for its existence is a verbal explanation of how to construct the companion *concept*. Hence, the evidence that a concept of something exists and is reconstructible, permanent, or memorable is an explanation *of* an explanation; which (if properly attested) is called an *understanding*.

This definition tallies with the perceptual motor paradigm in so far as the *Lev 0* explanation can be elicited non-verbally if there is an appropriate environment in which the relation it brings about can be modelled (literally as a physical model, a computer program for example, which, on execution, brings about this relation R). Any verbal explanation is a description of a model, either one that has been built in an environmental facility or one that does not, in fact, exist. For example, as Loefgren (1968, 1972) points out, a proof chain or a derivation of R (for *Lev 0* terms α, β; with α as an axiom) is the sequence

$$\pi_0(R) = \langle \alpha, \beta, \ldots, R \rangle$$

which is either the model of R or a description of it (the explanation of R) which is just a verbalisation of the *concept*, π_0, of R. Further, an explanation of this explanation (the verbal evidence for a *memory*) is a sequence (in *Lev 1* terms $a, b, \ldots$) like,

$$\pi_1(\pi_0) = \langle a, b, \ldots, \pi_0 \rangle$$

or, equally, this is a description of an 'internal-to-the-subject modelling operation' that reconstructs the concept, π_0, in the subject's *Lev 0* repertoire (and π_0, on execution, brings about R).

This imagery is stilted in so far as it represents serial operations (π_1 and π_0 are Turing machines, and π_1 is a reproductive automaton that replicates π_0). In general (hence our continual insistence upon the non-serial character of most mental procedures), there are many explanations and ways of modelling. This contingency is quite easily encompassed in the theory (Pask, 1973). However, the 'proof-chain' image makes it clear that the perceptual-motor paradigm and the intellectual paradigm, are in register at *Lev 1*. A *memory* is the same in either case; similarly, in either case, the constructive operations that build concepts to be remembered are described as the *learning strategy*. In contrast, at *Lev 0* all of the concepts (alias *strings* or *groups*) consist in models *composed in the subject's brain* and generally not described by the subject (though they are described by the experimenter when he talks of 'strings' or 'groups' or contrives a 'learning model' of the type discussed in Chapter 6; *not* to be confused with the *subject's* model). This internal-to-the-subject-model, though not described by the subject, is executed autonomously to produce a complex response and, if it were described, that description would be a *Lev 0* explanation (in the experiments under discussion an explanation of Ω or of a part of Ω).

3. Not in the run of psychological literature where 'memory' is often regarded as a storage.

Hence, the general criterion for *understanding* is to elicit in respect of each relation R (each part of Ω in the perceptual motor task and each topic in the intellectual task) an explanation and an explanation of this explanation. Several methods may be used to do so, all of them constituting means for externalising one or (generally) many chains, like,

$$\pi_0(R) = \langle \alpha, \beta, \ldots, R \rangle;\ \pi_1(\pi_0) = \langle \alpha, \beta, \ldots, \pi_0 \rangle$$

in the conversational dialogue, using a CET for this purpose. One method is called 'teachback'.

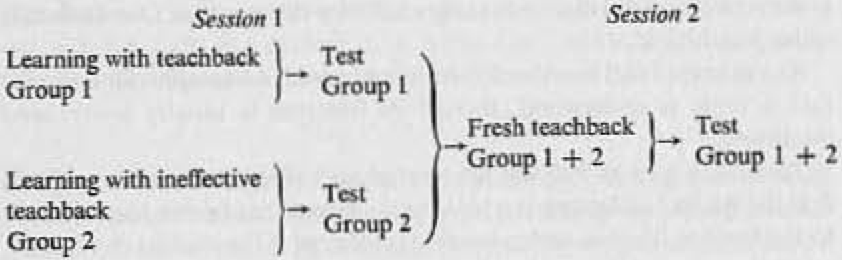
As a matter of fact teachback is commonly used in education for ensuring that a topic is understood, though its function is usually appreciated intuitively.

Teachback goes as follows: the teacher says of the student (or 'subject') that the student understands a topic to the extent that he can teach it back to the teacher. That is, understanding is inferred if the student can furnish an explanation of the previously discussed topic and can also explain why he gave that explanation or how he constructed it. The crucial point is that the student's explanation and the teacher's explanation need not be, and usually are not, identical. The student invents an explanation of his own and justifies it by an explanation of how he arrived at it (in fact an identical explanation is generally rejected unless the student can give a reason *why* the teacher's explanation was particularly good). It is required, of course, that the explanations (the *Lev 0* explanations) due to the student and the teacher explain the same topic R; moreover, that the *Lev 1* explanations produce a concept of R. But, in general, an indefinite number of explanations have this property (though there is also an indefinite number of psuedo 'explanations' that do not).

2.7 The Validity of Teachback Operations This technique (teachback) has been partially formalised using a restricted form of natural language dialogue, checked out by content analysis. The technique is crucial to the argument and is validated by experiments of the following kind.

In asserting that teachback (explanation of explanation of what has been explained in teaching) is an indication of understanding, we are positing that if a topic R is taught back then R will be retained at any rate whilst the student is learning the same material. Conversely, if he does not have a *memory* and a *concept* of R, the student has not (in the present sense) learned R. But the existence of a concept of R and its memory (though it may be inferred from an externalised explanation) does not imply that an explanation will be given. That depends upon the quality of the CET (if it were perfect, all concepts/memories would be externalised as explanations; otherwise there is no guarantee to this effect). Hence, we may expect to show, experimentally, that *any* topic R that is 'Taught Back' *is* retained; of those that are not taught back, some *may* be retained.

2.8 A Study This hypothesis was tested in studies involving over fifty students learning about thirty-five *related* topics (Pask and Scott, 1972a) using the following design.



'Dummy' or 'ineffective' teachback (session 1) is a procedure in which the requirement to explain is replaced by a requirement to give *correct* responses; up to a criterion of perfectly correct response, which is obtained by appropriate cueing and remedial action. The test in session 1 contains questions that demand explanatory replies. The test in session 2 contains some multiple-choice questions but mostly questions that demand explanations in reply. The latter are of primary interest (the multiple-choice questions act as a control). The session 2 'fresh teachback' is introduced to provide additional data about how explanations are changed and perverted by interfering events the student has experienced in the week or more between sessions.

First, every taught-back topic is recalled, with explanation, in the session 1 test; though many correctly answered though unexplored topics are not. The score difference between group 1 and 2 at session 1 is significant at the 0·1 per cent level ($0{\cdot}001 > p$). Next, the taught-back topics are nearly all retained until session 2; rather few of the unexplained topics are retained. Using the explanation-demanding question score, the difference between group 1 and group 2 at session 2 is enhanced, though the same level of

statistical significance is achieved ($0{\cdot}001 > p$). The difference is also significant in respect of the multiple-choice test component at session 2 but to a lesser extent (1 per cent significance, $0{\cdot}01 > p$). The fresh teachback data from session 2 is chiefly of interest in connection with a variation upon the experiment. It can be argued that although retention of taught-back items will be perfect *within* one session (it is), the resilience of a memory will depend upon the number of *explanations* produced in teachback; for example, that a student impelled to give *many* explanatons will fare better at session 2 than a student required to give only one. He has many ways of reconstructing a *concept* and this redundancy will combat the effect of interfering and incompatible learning experiences during the intervening week. This result is valid, and, in experiments designed to exhibit the effect by deliberately interpolating destructive learning experiences, it is highly significant (Pask and Scott, 1970). But in the study described, the effect is best exhibited by comparing and contrasting the detailed teachback records from session 1 and session 2 as the differences reveal both distortions and the operation of the reconstructive procedures that remedy them.

3 Intermediary Scale Experiments (Taxomnoy Learning)

The intermediary scale experiments were all concerned with modes or styles of learning and were designed to determine the free-learning strategies adopted by individual subjects and to assess the efficiency, for individuals with differing mental makeup, of different teaching strategies. In order to externalise the free-learning process (in which a student is at liberty to ask questions of his choice and direct his attention to different subproblems), the design embodied a conversational technique. For some of the experiments, the dialogue took place in a formalised language, via an interface (as in section 2) with a partly mechanised tutorial system. In others, the subject conversed directly with the experimenter who was, however, preprogrammed by fairly rigid rules.

All of the experiments can be conceived as experiments in relation learning and the relations in question are complex and redundantly specified. For example, one task in the series involved the relations inherent in a taxonomy; another concerned the relations describing a cyclic system. Embedded in each redundantly specified set of relations there is a kernel of essential relations which a subject has to learn and recapitulate. Though the subjects were aware of the essential relations, they generally learned and used redundant relations in order to access and manipulate essential relations while performing the test task which succeeded each experiment.

These are clearly combined experiments in learning and teaching. The design was partly engendered by educational objectives, but also by the general contention that theories of learning and theories of teaching do not exist as separate entities. There are theories of learning and teaching; that is all.

3.1 Relations and Types of Strategy If a relation R is described in the context of its domain, then it has an apparent dimension, n, equal to the number of properties in (descriptors of) its domain and an order or adicity m equal to the number of properties that R necessarily unites. The terms apparent dimension and order are easily interpretable (in the case that R is stated extensionally as a subset of the product of variables (U_i) indexing the related properties), i.e. as $R \sqsubseteq U_1 \times \ldots, \times U_n = U^n$ when an n-tuple $\langle u1, \ldots, un \rangle \in R \sqsubset U^n$ satisfies R. Ashby (1964a) provides an algorithm for determining the order m of an arbitary subset of U^n (his 'cylindrance'), and generally, m is less than n. In particular, following Ashby's line of argument:

(a) if $n > m$ (as usual), then R is the intersection of subrelations (cylinder subsets) of generally lower order and dimension;

(b) there will often be several families of 'cylinder subsets' that intersect on R; any one family specifies R; the set of families provides a redundant specification of R.

Relations may also be specified in a language where they are named by words or described by phrases, for example, 'father of' or '$>$' or 'the T.C.A. cycle' or 'A allows B to do C' or 'happiness' or 'antelope' (a class of animals defined by relations between behavioural or physical properties; the taxonomies we actually employed classified mythical, Martian, creatures since these were necessarily unfamiliar to the subjects). Sometimes the order of the relation and the dimension of its domain are explicit; for instance '$>$' has order 2 and is irredundantly specified (for numerical domains of dimension 2; similarly, 'A allows B to do C' has order 3, though the dimension is not explicit). But, linguistically stated relations can be quite respectably under-specified. It is possible to list situations that engender happiness or relations supporting the antelope character *ad infinitum*.

There is nothing mysterious about the idea of an underspecified relation. On the other hand it is an important and interesting idea. An individual who understands the language in which the relation is stated can extensionally specify the relation, at a given instant, by ostending the n-tuples which belong to R and separating them from n-tuples that do not. But this specification is tied to the individual and to the instant; if R is underspecified, then n (and possibly the value of m) is variable in any general statement that is independent of instants and individuals. Hence, it may be deemed prudent, and it is certainly quite harmless, to regard underspecified relations as the open

union of fully specified but redundant subrelations of possibly variable dimension. For example, Banerji (1970) distinguishes between input properties (in a fine-structure family) and non-input properties. Phrased in these terms an individual need not specify all non-input properties at the outset. But, if his goal is underspecified, he will use them and could specify them at some later stage.

It is profitable to distinguish between two major kinds of problem-solving computation. At one extreme, problems are solved by the serial application or basic routine, produced by a type (a) strategy (section 2). At the other extreme, problems are solved by mustering these routines concurrently (an effectively parallel computation produced by type (c) of section 2). Amarel (1969), taking the position of a formalist, conceives the first type of problem solver as a macro-assembler and the second as a director and compiler combined. If we make the same distinction in terms of artificial intelligence and psychology, we can say that the first sort of problem solving resembles the operation of a programme such as Newell, Shaw and Simon's 'logic theorist', whereas the second sort of problem solving resembles the operation of an associative network like Quillian's (1969) 'TLC'. When R is redundantly specified (as it is) then the second sort of problem solver aims for several subgoals at once and is likely to achieve R as the intersection of various families of subrelations.

The crucial point and the reason for making this distinction, rather than many others, is that (in common with some other pairs of processes) these two are incompatible. Data used by one process cannot be used without complete recoding and reconstruction, by the other.

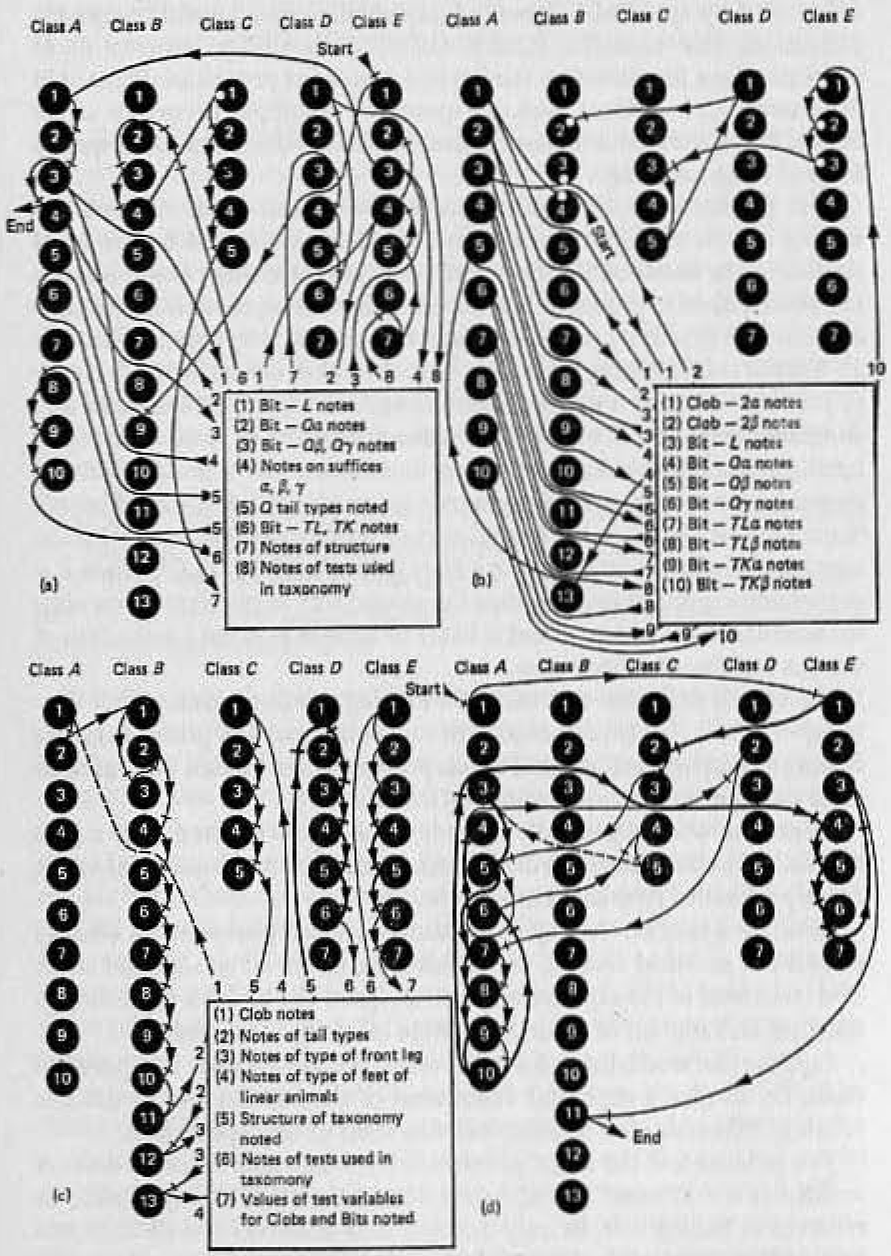
If, as is nowadays generally agreed, a concept is equated with a goal directed procedure, then the distinction demarcates two conceptual types, namely a serialist type and a holist type.

Consider a task involving a redundantly-specified goal relation. (Nearly all tasks of practical concern are of this type, as are all educational tasks. The tasks used in the experiments were designed to be of this type, though the form and amount of redundancy were carefully controlled.)

Suppose that a serialist and a holist solver both know what is required of them, i.e. to give a canonical description of the relation and to use this relation. This condition was also satisfied in all of the experiments.

The serialist and the holist problem solvers use quite different data. A serialist is embarrassed by redundant data, unless it is clearly marked as redundant. Failing that, he only succeeds if he discovers and sticks to one irredundant description of the goal relation, and, since a canonical description is to be elicited, this one must be the one the experimenter has in mind.

By way of contrast a holist deals concurrently with many descriptions, and even if asked to give the (experimenter's) canonical account of the goal



relation he generally constructs it by cross-reference to a set of different and redundant descriptions.

The same comments apply to the way in which both serialist and holist problem solvers use the goal relation in solving problems.

An individual's strategic type may be determined in various ways. If the subject is allowed to free-learn a taxonomy of animals (for example by gaining access to indexed cards on which are inscribed facts about the classification and the creatures classified), he exhibits a characteristic pattern of exploration (see Fig. 103).

Content analysis of the questions asked about the study material during free-learning (based on the cylindrance measures of section 3.1) also provides an independent discrimination; the serialist either strings through the card packs according to a linear plan or poses hypotheses which test *one*, generally *unary*, predicate value at once, and then questions the experimenter about the truth or falsity of the hypotheses. In contrast, the holist poses relational hypotheses; his questions are concerned with the validity of predicate functions of many variables.

These discriminations are more readily elicited during *teachback* where the hypotheses are fully externalised for scrutiny.

Further, there are characteristic differences in uncertainty and correct belief (as determined by confidence estimates; Chapter 1, section 7). As a slight oversimplification, the holist entertains a measurable uncertainty and correct belief for topics 'ahead of' the topics he is dealing with, and this uncertainty is reduced (with an indrease in correct belief) as he studies a cluster of topics from which he garners data and relates it to the topic 'ahead'. A serialist, on the other hand, is unable to give a confidence estimate in respect of a topic 'ahead' of his current focus (at any rate, a topic 'far ahead') and he typically does not explore further until the one (or occasionally the few) topic relation currently occupying his attention is

Figure 103 Patterns of exploration for the learning of a zoological taxonomy: (*a*), (*b*) serialists; (*c*) redundant holist; (*d*) irredundant holist. *Key*: Class *A*, pictures of typical members of subspecies; Class *B*, contextual information (behaviour, habitat, and so forth); Class *C*, statements about structure of taxonomy; Class *D*, physical characteristics used for making distinctions between subspecies; Class *E*, meaning of subspecies names and codes.

● Card number of a particular column of the array.
——— General search.
——— Testing many predicate hypothesis.
- - - - - Testing single predicate hypothesis.
→● Failure to find information sought after.
→● Transition between questions.

fully established (he is completely *certain*; ultimately he is certain of the *correct* topic relation).

All this begs the question of what a topic 'ahead of' or 'far ahead' of may be; this question is answered in due course and the reader is asked to accept a dictum that the goal relation is ahead of all others, and that topic relations elsewhere in the study material can be sampled (by posing questions and eliciting confidence estimates).

Finally, there are some paper and pencil tests which (less reliably) discriminate the holist and serialist.

A holist strategy and a serialist strategy are cognitive processes. They are also incompatible in the *same* context (though a student might change between contexts). In the context of a given chunk of study material, for example, the description of a species to be classified according to a taxonomy, the action of cognitive fixity should ingrain one type of strategy or the other.

This effect occurs. It is very pronounced, to the extent that, during a particular piece of learning, an individual (once started) can be typed dichotomously as 'holist' or 'serialist' (as tacitly assumed in the discussion up to this point). That is, in *one* conversation the strategic types are 'all or none' distinct. It *is* possible to shift type by external instructions. But the amount of disconfirming data needed to effect this transformation increases with time and, as a result of the transformation, the previously erected cognitive equipment has to be discarded; the student who is impelled to change 'starts afresh' and relearns.

In contrast, it is possible to characterise an individual as having a statistical *disposition* to use one kind of strategy or the other by examining his free-learning performance over different tasks, preferably several of them. If his successful performance is scrutinised, the disposition is *one* index of competence to execute the same type of learning strategy (there are several other methods of determining competence, to be introduced in the next volume).

Since the holist/serialist distinction rests upon a process distinction it is possible to design teaching strategies that are *matched* to a holist and *mismatched* to a serialist as well as strategies *matched* to a serialist and *mismatched* to a holist.

The design process is systematic (again, it is detailed in the next volume) and it entails finding paths through the topics, i.e. the part relations of a goal relation, that lead to the goal relations and guide the student's attention in a way that maximises (holist) or minimises (serialist) the degree of concurrent processing required.

With these preliminaries, it is possible to make sense of the intermediary-scale learning experiments (Pask and Scott, 1972). The design is fairly

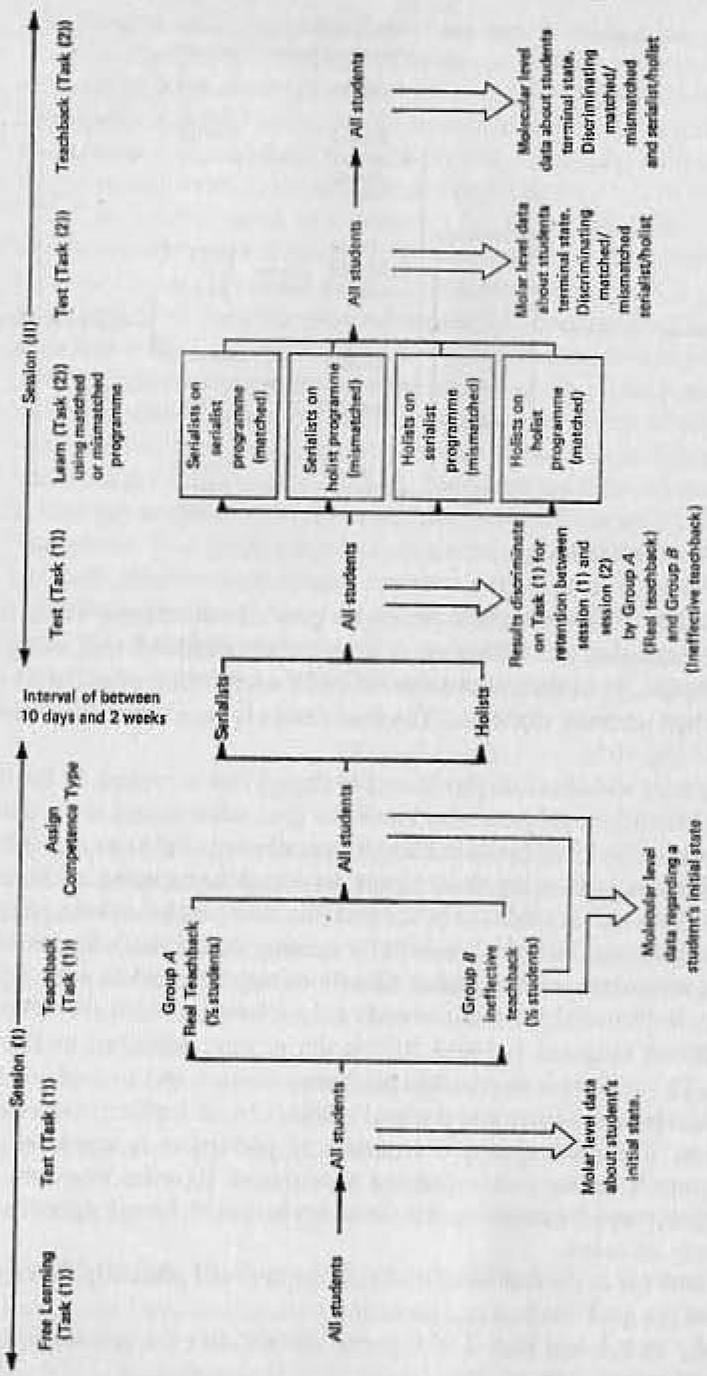


Figure 104 Final design.

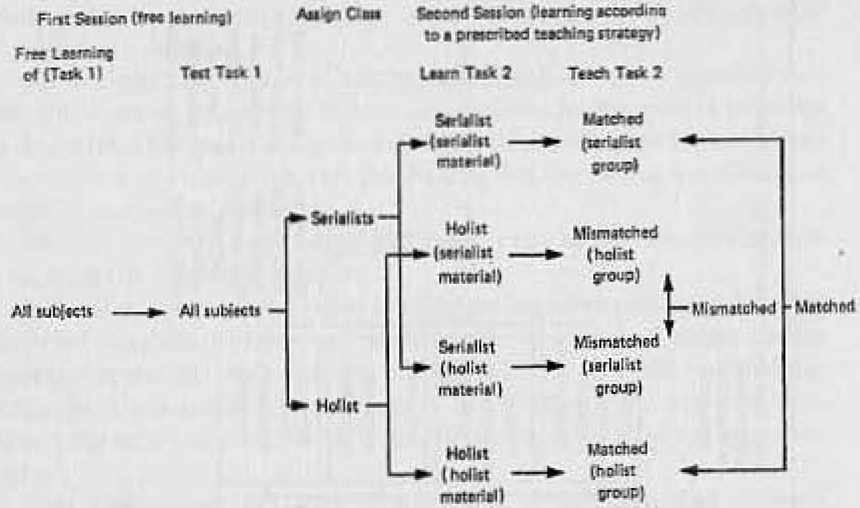


Figure 105 Outline design.

complex (Fig. 104), since the experiments provide information about free-learning strategies, their effect upon retention of teachback and, using the same subjects, yield data about the influence on learning of matched and mismatched teaching strategies. The final design is based upon the skeleton design of Fig. 105.

The species classification (taxonomy learning) task is typical. In the free-learning condition, subjects who know the goal relation and are provided with charts of the kind shown in Plate 10, are allowed to explore an environment of cards containing data about an imaginary species of Martian animals. The cards are indexed by content (for example: behaviour, physical characteristics and historical reasons for naming the animals). Some data is written; some data is pictographic. Questions may be asked as part of free-learning. Both verbal transactions and card pack explorations are recorded.

Teachback (Figures 104 and 105) is the process described in the last section. The student is required to achieve perfection and to explain all of the topic relations he uses and the goal relation. In the ineffective teachback condition, the same objective criterion of perfection is employed but explanation is neither demanded nor encouraged. In order to ensure that all students reach the criterion, it is necessary to furnish knowledge of results to remedy mistakes.

The tests (as in the last section of this chapter) call primarily for explanations of the goal relation and *parts* of it.

Finally task 1 and task 2 of Figures 104 and 105 are two distinct but structurally similar taxonomies.

3.2 Results 1. As expected, explanation chains do occur in teachback.

2. The explanation chains of serialists and holists are (as predicted) quite different in kind, especially with respect to the type and referent of the hypothesis tested during the production. (Subjects have been independently characterised as serialist or holist in task 1; simple statistical tests are inappropriate but pattern matches are significant.)

3. The establishment of a memory for a concept X may be inferred unequivocally from the existence in teachback of a chain reproducing the concept. Groups of twenty show differences significant at 0·1 per cent. This memory persists for a few weeks at least and for months in all subjects tested after that interval. There is reason to believe that the memory is indelible.

4. After learning to solve problems, one group of subjects were tested until they could achieve full proficiency. Another group of subjects, at the same level of proficiency, were submitted to teachback. After a couple of weeks, all the subjects were retested. The difference between the two groups is clear-cut to the extent that no statistical method is needed to discriminate their scores. Teachback subjects have a memory, test subjects did not do so, i.e. under these circumstances a memory is dependent on teachback.

Students previously assigned to the competence-type serialist (S) and holist (H) were instructed in task 2 via serialist training routines (SR) and holist training routines (HR). Of the four combinations; namely,

$$(A) = S \rightarrow SR$$
$$(B) = S \rightarrow HR$$
$$(C) = H \rightarrow HR$$
$$(D) = H \rightarrow SR$$

both (A) and (C) are matched, whereas both (B) and (D) are mismatched. Subsequently, all subjects were tested for retention and regeneration of the learned material.

Serialist subjects fare just as well, on the average, as holist subjects (the problems were chosen to obtain this result). However, the matched subjects have a much higher test score than the mismatched subjects. Groups of twenty give differences significant at the 0·1 per cent level ($0{\cdot}001 > p$).

It looks as though effective learning depends upon securing a matched condition. If the size of the effect is as large as we suspect, classroom education must be grossly inefficient and an improvement of several orders of magnitude could be achieved by the (entirely practicable) expedient of matching the tutorial strategy to the competence of an individual.

4 Outstanding Problems of Representation and Understanding

Two main requirements are outstanding; first, a reasonable means of representing what may be known (the goal relation and its various components or topic relations) and secondly, a reasonable method of detecting the occurrence of an *understanding*. It is clear enough at this point, that a conversation open to observation, measurement, and so on is (whatever else it may be) a sequence of *understandings*. It will later be possible to establish that the condition of understanding observed in teachback is fundamental.

The first requirement (a representation of knowledge) is obtrusive both on practical and theoretical grounds. From a pragmatic point of view it is necessary to deal with educationally realistic stretches of learning and topic or subject matters as large as a course (in statistics, biology, or sociology). Learning the background data for a taxonomic scheme with some dozen distinct tests in it (the Martian animals, for example) gives rise to explanations or learning strategies of the kind shown in Fig. 103. It is not difficult to appreciate that a study of chemistry learning, for example, would be altogether intractable if *this* subject matter was represented in the same way. Moreover, even if this representation were practicable and manageable, it would (for reasons to be given) prove inadequate; there is no way, in general, to express a teaching strategy in terms of such a figure though it happens to be possible in the specific cases under discussion. As an equally significant objection, there is no systematic method for combining chunks of knowledge to form curricula or semantic depositories of one kind or another. We seek, and have obtained, a representation scheme that does meet these and various other criteria quite satisfactorily.

From a theoretical point of view the representation in Fig. 103 (with categories of data cards corresponding to dimensions) is a description scheme or indexing scheme for learnable and knowable relations; it is not a representation of these relations, *per se*, together with the relations that unify them into a coherent body (the all-adumbrating 'goal relation'). In fact this goal relation (the species, membership of which is tested by the taxonomy) can be expressed as a subset of the values of these dimensions. Further, since the description is deliberately redundant, the goal relation can be expressed as a subset of the products of several distinct clusters of dimensions and as a subset of the products of *many* different *value* sets. However, the description is a description and it is not unique; many different descriptors and indexing schemes can be chosen to describe the same goal relation. Further, it is essential to admit this possibility because some subjects learn very successfully (the redundant holist of Fig. 103c is one example) in terms of additional properties and descriptors which they bring to bear on the task.

Of course, it is possible to represent the topic relations that are learned and to represent the goal relation as a relation between them. To do so, in general, yields a relational network similar to, but not identical with, the kind of network quite widely employed in artificial intelligence (for example, Winston (1970), or Winograd (1972); the latter stressing the descriptive and explanatory aspects of search in machine-interpretable terms). Such

structures are very large, even for modestly sized chunks of educational material; one is shown in Fig. 106 to give the reader an idea of the size

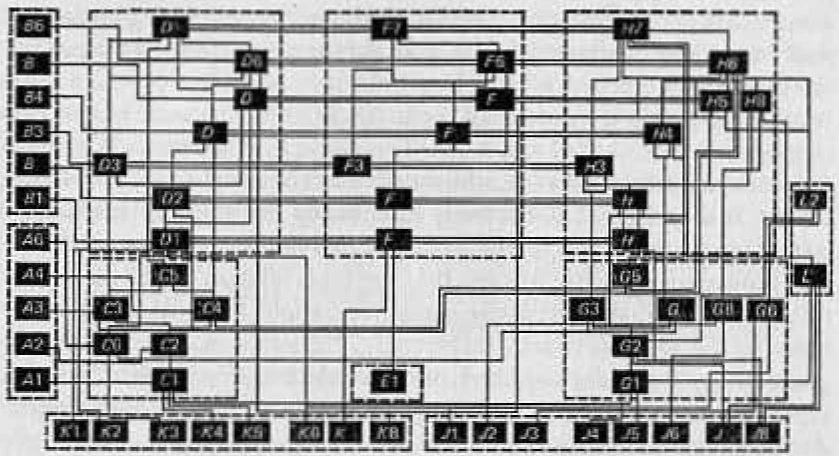


Figure 106 The 'Probability Theory' entailment structure: *A*1 to *A*6, peripherals for *C* (e.g. 'example from games of chance'); *B*1 to *B*6, peripherals for *D* (e.g. 'example from behavioural science'); *C*, deterministic experiments; *D*, random experiments; *E*, theory of simple results; *F*, probability theory; *G*, structural model; *H*, probability number model; *J*, *K*, primitives (e.g. 'order'); *L*, peripherals stating relations between parts of *G* and *H*. *Key for nodes of entailment structure*: *C*1, simple results; *C*2, composite results; *C*3, exclusive results; *C*4, inclusive results; *C*5, deterministic experiments; *C*6, random results; *D*1, frequency of simple results; *D*2, frequency of composite results; *D*3, frequency of exclusive results; *D*4, frequency of inclusive results; *D*5, conditional frequency of simple results; *D*6, conditional frequency of composite results; *D*7, random experiments; *E*1, theory of simple results; *F*1, probability of simple results; *F*2, probability of composite results; *F*3, probability of exclusive results; *F*4, probability of inclusive results; *F*5, conditional probability of simple results; *F*6, conditional probability of composite results; *F*7, probability theory; *G*1, event set; *G*2, composite events; *G*3, exclusive events; *G*4, inclusive events; *G*5, structural model; *G*6, probability numbers; *G*8, complement of composite event; *H*1, probability numbers (p. no.) of simple events; *H*2, p. no. of composite events; *H*3, p. nos. of exclusive events; *H*4, p. nos. of inclusive events; *H*5, conditional p. nos. of simple events; *H*6, conditional p. nos. of composite events; *H*7, probability theory; *H*8, p. no. of complement of composite event; *J*1, model/real world distinction; *J*2, inclusive/exclusive distinction; *J*3, arithmetic operations; *J*4, universe; *J*5, set and subset; *J*6, complementation; *J*7, intersection; *J*8, union; *K*1, long-run stability; *K*2, counting; *K*3, order; *K*4, qualities; *K*5, 'one at once' definition; *K*6, fractional numbers; *K*7, definition of probability relation; *K*8, definition of given knowledge; *L*1, relation between logic of structure and arithmetic of p. nos.; *L*2, relation between p. nos. and conditional p. nos.

involved and not (until the next volume) to reveal its structure. Figure 106 represents one thesis (of one subject matter expert) about elementary probability theory. Under execution, if a search or exploration is carried out, these networks are verb networks. Moreover, by performing operations that image the relations said to hold between the topic relations (and represented by directed arcs in Fig. 106), it is possible to construct the relation in question from other topic relations in the same mesh. This requirement is demanded and is called *cyclicity*; in other words Fig. 106 is not just a representation of an arbitrary thesis about elementary probability theory. It is certainly a thesis freely invented by the subject matter expert *apart from constraints upon what properties the relational network shall have*. One property is cyclicity; namely, that any topic relation can be derived by relational operations, that *image* mental operations from other topic relations in the same network. Incidentally, relational operations may be grouped together under a common rubric as relations of *entailment* (between topic relations) and the corresponding collation of mental operations is *discovery*.

The other constraint upon the person or group (or student, as a matter of fact) who specifies a thesis and thereby a network, is that each topic relation in the mesh shall be *Lev 0* explicable; either he can explain it or, given an appropriate physical device such as a programmable computing machine, a laboratory, or a game situation (for example, a role-playing game in management or economics), he can model it. It should be stressed that *Lev 0* explicability has no obvious connection with veridicial truth or even with formal rectitude. The expert (or the student) can explain scientific nonsense, if he wants to do so. This is quite intentional as the thesis is *his*. The explicability requirement is only that the model *works*; it is a criterion of *satisfaction* not a criterion of truth. Finally, a requirement of *consistency* is involved; namely, that if two or more topic relations are differently named and distinct in the mesh, they then do not have the same class of *Lev 0* explanations.

Between them, the canons of cyclicity and *Lev 0* explicability, both of which are imposed upon an acceptable mesh, guarantee that the mesh is learnable and memorable. They lend it the quality of a gestalt closely related to the interpretation of a gestalt given by McCulloch and Pitts (1943) though not quite identical with it.

To see this in outline, one should notice that cyclicity means that any subject equipped with the posited mental operations that carry one topic relation into another, is able to furnish an explanation of how he arrived at a *Lev 0* explanation for each topic relation in the mesh. This is a condition of *Lev 1* explicability. But the *understanding* of a topic relation is evidenced by the execution of a *Lev 0* explanation, together with its justification by a

Lev 1 explanation; hence, each topic relation in such a mesh *may be* understood. With reference to Chapter 10, the mesh is a permission giving structure for *learning*, i.e. as asserted before, it specifies what may be *known*. The class of models corresponding to the class of *Lev 0* explanations is no more nor less than a process representation, a permission giving structure for *doing*, which specifies what may be done. *Lev 0* explicability requires that one process representation is attached to each node in the relational network.

Such networks (generally called entailment meshes) have a couple of other apparently innocent but actually far-reaching properties.

(a) They are not really specified in extension but in terms of processes (goal directed or problem solving procedures); that is, intentionally.
(b) They are isomorphic to an indefinitely large number of other meshes.

Property (a) means that a topic relation could be explained in any context; the point was mooted in section 3.1 (the brief comment on verbally stated relations). A concept is a procedure that brings about a relation. If the procedure is used to build a model which, on execution, does bring about the desired relation then quite clearly this relation is defined in extension (by a listing). But the particular universe in which the model is built and executed may be chosen by the student (apart from limits of convenience) and in this sense the student is free to, and required to, engage in acts of predication (choosing the properties which *he* deems *relevant*), for his concept could be executed in many domains of interpretation.

Property (b) is of consequence when it comes to describing the relational mesh; in other words, imposing coordinates (descriptors) that act as the grid lines (in an often very curious topology) of a cognitive map. Because of property (b) it is possible to choose an indefinite number of descriptive schemes, all of which are compatible and satisfy the last requirements to be outlined.

(c) The description scheme uniquely indexes each node (representing a topic relation) in the entailment mesh, so that it can be ostended.

(d) By using the description scheme, it is possible to point at some 'blank' positions and place holders for nodes that do not currently exist, but might be brought into existence. By this means, it is possible to knit the ends of meshes together and it is possible for meshes to evolve, in a systematic manner, as a result of learning.

The hierarchical structure shown in Fig. 94 (page 252), an and/or tree is *not* a verb net; it is a *noun* tree. Its deficiencies, which were mildly criticised, can be summed up by the fact that it is not and, in general, is not derivable from, an entailment mesh though *this* one *can* be so derived. It does not represent knowledge in the ordinary sense; rather, it represents a peculiarly stilted description of knowledge. To show this point, consider any description of an entailment mesh. Amongst other *descriptors*, it is possible to choose (and to obtain the promised sense of a topic relation which is 'ahead of' another, it is necessary to choose) at least one descriptor with the broad connotation of subordinate/superordinate. In terms of *this* description the topic relations are partially ordered; that is, one aspect of their description is partially ordered and quite possibly this *ordering* looks like Fig. 94. Hence, within this ordering, some topic relations are 'ahead of' others and, as a rule, the goal relation is placed at the top as a 'head' in this description. The essential point is that there are many descriptions with different 'heads' for the same entailment mesh. In fact, Fig. 94 *is* such an ordering and it has one intriguing characteristic; namely, *any* topic could be chosen as head in some description of the underlying mesh. That is, it would be reasonable to learn about truth tables in terms of variables and connectives (as in Fig. 94) or, using a different ordering, to learn about connectives in terms of truth tables, or variables in terms of connectives and truth values.

The second requirement (a means of detecting the understandings that go on in a conversation) is best tackled in the context of an entailment mesh that forms the conversational domain; the topic or subject matter that is under discussion. In this context we need a mechanisable test for understanding. The 'mechanical' requirement upon the test or detection device is chiefly imposed to ensure that it is well defined. Of course, it is useful to have a mechanically executed test for other reasons (for example, in order to instrument a full-blooded system of computer-assisted instruction or computer-guided learning, or merely to relieve the experimenter, in a seriously large learning study, of an impracticable burden). However, there is nothing magical about this gadgetry; the test could be performed (as it is, in teachback) by a human experimenter. The crucial issue in either case is whether the test is definitive for it is to rank as an alternative to Turing's test for intelligence in conversation (Turing, 1963, 1969). We shall say that any conversation (whether between two men or a man and a machine) is characterised by a contiguous series of occasions upon which understanding is reached by a *process*, without too much commitment about *where* the process is executed (for example, in the teacher's brain, in the student's brain, or in some inanimate processor). This process might be deemed intelligent (i.e. an instance of intellect and not just 'able to pass intelligence tests', which is a trivial criterion, judged either by Turing's test or our own). As a matter of orientation, I happen to call the putatively intelligent process *learning* and to notice that it *is*, quite literally, symbolic *evolution*.

5 Taming Wild Understanding

The mechanical system which tests for and may also manipulate understanding in a CET externalised conversation is shown in Plates 11 and 12 (these pictures are chiefly intended to give the reader an overview of the equipment used in studies of learning; not, in this volume, to inform in any depth). The system is founded upon a theory of what understanding is (Pask, Scott, and Kallikourdis, 1973) and the following comments in the matter constitutes no more than an orienting outline; sufficient to indicate the flavour of this field of research.[4]

5.1 It is maintained that a conversation in a domain, R (satisfying the conditions of the last section), is a self-reproducing system. That is, using *Rep* to stand for 'reproduction' (in the symbolic and automaton theoretical sense as in Chapter 2) and choosing any topic relation (R_i) in R on which the conversation is momentarily focused at a given occasion, there is a stable entity

$$Rep\,(Rep\,(Rep\,(R_i)))$$

where $Rep\,(R_i)$ is a *concept* i of R_i

and $Rep\,(Concept\ i) = Rep\,(Rep\,(R_i))$ is a *memory* of R_i

If it were externalised in a conversation, by a CET technique, this entity would be symptomatised by a cycle of explanation (namely teachback) consisting of an explanation by one participant, agreed by the other, and a justifying explanation also agreed (the comment follows from our earlier specification of what an explanation or modelling operation *is*). Thus, if the conversational participants are A and B, the first part of the equation becomes (under these conditions) '*Lev 0* explanation by A of R_i to B and *Lev 0* explanation by B of R_i to A' (with agreement) and the next part becomes '*Lev 1* explanation by A of R_i to B and *Lev 1* explanation by B of R_i to A'; the last *Rep* term closing the system and stabilising its execution.

This 'stability' accompanies (is *due* to or gives *rise* to, as you prefer) the effect of cognitive fixity; i.e. it is a property of such processes that when executed in respect of a given domain they 'recognise' and reject classes of dissimilar processes executed in the same domain.

5.2 Such an entity is recursively specified with respect to R_i, or in general, R. But it may be executed in various processors, for example, in a teacher, a student, or a mechanical device. In order to *observe* understanding on the part of A and B, it is necessary (using a CET) to externalise the

4. Experiments using the system are described in Pask and Scott (1972b, 1973).

reproductive cycles that *might* be private to a student (say) and to represent them in terms of dialogue. For the facility shown in Plates 11 and 12, the dialogue takes place in a mechanised command and question language with various transactions, mediated through the interface, that point to topic relations and clusters of topic relations in the *described* entailment mesh (the large display). As a result of these transactions a conversation takes place between a regulatory heuristic (one participant, *B*) and one or more students (the other participant, *A*). In the course of it, at any occasion, agreement is reached regarding a topic or topics the student is currently aiming for (generally, topics he can describe by locating them in the cognitive map, but is not yet in a position to learn about) and topics that he takes as his goal and is learning about, as well as a backlog of topics that are already *understood* (the *understanding* condition has been satisfied). In order that goal topics shall be understood it is necessary for the student to explain them; in this facility we use non-verbal explanations that are models made on a computer-monitored, laboratory-like, modelling facility, which, unlike the rest of the equipment, is task specific and is labelled 'Statlab' in Plate 11.

All of the conditions of *any* node in the large display are marked by coded signal lamps; hence, the progress of learning, i.e. a learning *strategy*, is a wave of activity headed by *aim* nodes followed up by goal nodes, and leaving a train of *understood* nodes in its wake. These conditions are continually displayed to the student.

5.3 It can be shown that a system

$$Rep\,(Rep\,(Rep\,(R_i)))$$

can be dissected into two subsystems (Loefgren's (1972) reproductive automaton and its support in the domain *R*). The desired condition is to detect and externally observe an understanding so that the chains of explanation (alias reproduction) all penetrate the interface between a student and the heuristic governing the operation of the machine. If the student's mental organisation is *A* and if the machine heuristic is *B*, then the basic form of *B* is a procedure for securing this condition with respect to all topics in *R*.

5.4 The *Lev 0* component is the explanation or modelling operation. The *Lev 1* component need not be made explicit, since a description of the learning strategy as it is displayed *is* such a thing. In contrast to teachback, the *Lev 1* explanation or 'how a topic is to be learned' comes *before* the *Lev 0* explanation of the topic is furnished. If that is furnished, the topic *is* learned and understood.

5.5 There are several tricks in the system; amongst them is the trick that once an understood marker is assigned it cannot, under ordinary circumstances, be deleted. The justification for this particular trick relies upon the fact that the conversational domain is so constructed that a concept, once established, may be reproduced as a memory. Provided the conversation remains within this domain, the concept will not be forgotten.

5.6 It is possible to constrain the *B* heuristic in various ways. For example, it is possible to sample the student's uncertainty both regarding a *Lev 0* explanation and regarding any of the exploratory *Lev 1* explanations in the learning strategy (and so, for example, to give substance to the uncertainty distributions noted in section 3.1). The subjective uncertainties are computed by using BOSS as an input device (Chapter 1, section 7).

5.7 Further boundary conditions can be imposed to regulate uncertainty and to ensure rapid learning of concepts that are retained (at least whilst attention is concentrated on *R*). Other tutorial procedures (fostering learning to learn, for example) are possible and specific data gathering facilities can be instrumented.

5.8 The theoretical superstructure outlined here, to be developed in the next volume[5], allows us to predict rather than observe the isomorphism mooted in section 2.5. Both perceptual-motor, cognitive and, for that matter, social processes are identifiable and explicable as the components of an understanding (as it is interpreted within such a theory). The isomorphism is not merely an interesting correlation or a loose similitude.

5. Or, in different terms and using a more technical approach, the superstructure is discussed in Pask, Scott and Kallikourdis (1973).

12 Points of Departure and Development

In order to resolve the problems which were chiefly obtrusive in the last chapter and which were mooted and superficially aired in Chapter 4, it is necessary to introduce very radical and not always welcome innovations. The book has laid the necessary groundwork, in showing that a revision is required, in pointing out the reasons for disquiet and in indicating an embryonic form of the new theory. The theory itself is tersely stated in a monograph on cognition (Pask, 1973) and in a series of papers (Pask, 1972; Pask and Scott, 1972a, 1973; Pask, Scott and Kallikourdis, 1973) and it has been quite widely discussed, debated and modified as a result of criticism. The notions are presented, from a systemic or cybernetic point of view in the next volume.

1 Outline

In essence, the theory follows the recommendation of Chaper 4 and takes a *conversation* as the fundamentally observable unit. As a result, it is possible to cope with *reflective* systems in stages.

1.1 The participants (whether there are just two of them or many, as in a social organisation) are no longer distinct as processors, though under special conditions they may be *rendered* distinct. The cogency of this point of view (which on the face of it, is hard to stomach) is buttressed by some empirical data derived from work with small groups of individuals learning collectively about their surroundings and one another (the theoretical underpinnings are in Pask and Von Foerster (1960, 1961) and the empirical data is in Pask (1962), Lewis and Pask (1964), and Pask and Lewis (1969).

1.2 In order to sample the interaction between the human beings in such a group, it is essential to employ multi-level indices of communication. These are based upon the IPM tests used by Laing, Phillipson and Lee (1966) (much in the spirit of Batesons work on the 'double bind' situation) and these techniques are embedded in a special extension of game theory, the theory of *metagames*, presented by Howard (1966a,b) and recently

published as a monograph (Howard, 1973). Some of the concepts involved were introduced in a paper on stability and style in communication (Pask, 1971b). Viewed thus, it turns out that the stable entities in a learning group are coalition-like structures (in the sense of stable game playing coalitions) and not single human beings. These coalitions are the most resilient and tractable units to study.

1.3 Comparable phenomena are manifest in connection with the type of computation called concurrent as well as cooperative; especially if the computation images a reproducing population of automata. In order to substantiate this approach the preview of these systems, given in Chapter 3, has been extended and related more firmly to the standard theory of self-reproducing automata (in essence, by considering reproducing configurations on a tessellation plane that is, itself, a reproducing configuration). Moreover, it is necessary to show that this construction does not inevitably lead to an indefinite regress of systems. This development provides the dynamic or mechanistic backbone that makes it possible to embed the procedures executed during a conversation into a class of processors.

1.4 At various points, languages with a full semiotic (a pragmatics, semantics and a syntax) have been contrasted with formal syntactic languages. In order to theorise about dialogue and to make sense of reflective systems it is necessary (at the beginning of the next volume) to examine command and question languages in some detail, and to elaborate upon the ideas of 'information' appearing in Chapter 1 as well as the notion of 'tense'.

1.5 The following summary remarks are chiefly culled from the prefatory note to the series of papers in which some aspects of the work are set out (Pask, 1972).

The theory has eight compartments.

1. The theory of conversations between two or more individuals. A conversation takes place within a contractual framework and this, rather than a stimulus/response system, is argued to be the least meaningful experimental situation. In a conversation (and conversations are many faceted, enough to adumbrate most types of experiment) it is possible to observe psychological events; whereas in other situations, it is not. Observation consists in recording and interpreting dialogue between the participant individuals. Experimental regulation, parameter change, and so forth, is achieved by instructing one participant to execute a prescribed heuristic. It is crucial, however, that a participant does so and not the observer, and that the participant retains the necessary partiality as well as the liberty of his role; hence, the heuristic is not to be equated with an algorithm. Often

it is convenient to study man/machine conversations and, in this case, the heuristic becomes a special type of non-deterministic programme or a fuzzy algorithm executed, without numericisation, in the participant machine. For a limited, but outstandingly important class of conversations (so called *strict* conversations), it is possible for an observer to maintain the position of an external observer and thus to report precisely and often in quantifiable terms about the dialogue.

Clearly a conversation involves individuals who participate in an ongoing process of understanding, retaining and learning, and it also involves a set of topics, the conversational domain. As a result, further compartments (sub-theories) are generated.

2. The (sub)theory of individuals is concerned with characterising potentially conscious entities (human, mechanical, or both) which have certain invariant and unitary qualities (by virtue of which they count as 'individuals' rather than widely disseminated organisations).

It appears that many characterisations are possible and all of them rely upon a general notion of stability which can, with rectitude and some advantage, be conceived as a property of self-replication, which is usually contingent upon the choice of conversational domain.

One characterisation is familiar; 'an individual is a human being (or his brain) regarded as a biologically self-replicating system'. We call this an M-Individual or mechanically characterised individual. The other characterisation to receive serious attention rests upon a nested series of definitions; concept $\triangleq$ procedure; memory $\triangleq$, a procedure for replicating a concept; individual $\triangleq$ procedure for replicating a class of memories. The resulting entity is called a P-Individual (psychologically characterised individual) and has many of the properties ascribed by anthropologists to a role, in society or industry, for example. A P-Individual is also a procedure and, as such, is run or executed in some M-Individual, *qua* processor. However, it is quite exceptional to discover the (usually assumed) one to one correspondence between M-Individuals and P-Individuals; in general, the latter are more convenient units to work with though conversations involve both of them.

3. The (sub)theory of conversational domains in which P-Individuals may converse (and remain stable) is concerned with the representation of knowable and learnable relations. The representation schemes found adequate closely resemble 'verb networks' but have only a peripheral (and ambigouus) connection with 'noun trees', such as a species/genus classification. Moreover an acceptable representation of what may be known or learned (here called an entailment structure) is invariably cyclic (which gives it a Gestalt property) and invariably associated with a task structure that indicates what may be done to bring about the relations for which concepts

have been learned. The entailment structure may be given by some numinous authority (for example, a curriculum specialist), or, within certain limits, it may evolve in so far as the participants in a conversation on this domain are able to act upon and modify the structure. In either case, generation rules must be given and justified.

4. The (sub)theory of process is applicable to all manner of decision making, interrogation, and so on, but is chiefly developed for a theory of learning and teaching; i.e. the conversation is a tutorial contract, the entailment/task structures represent 'subject matter', one of the participants has the role of student and one (either human or machine) the role of teacher. Within the structure afforded by the other (sub)theories it is possible to predict well-defined styles and strategies of learning, to confirm their existence, and to check their efficacy. This aspect of the theory is the most directly applicable in designing maximally informative experiments, interviews, and so forth, or in specifying effective modes of tutorial conversations (some of them dramatically superior to others). However, the process theory is based upon the individual and the knowledge theory; moreover, these can be applied independently (the latter to prepare potentially informative data for advertising, course assembly, mass media presentations and management, for example).

5. The (sub)theory of vehicles comes about as follows. The dual characterisations (M-Individual, P-Individual) which proves to be a prerequisite for the developments mentioned already, give rise to the notion that P-Individuals (cultural entities, minds) inhibit M-Individuals (processors able to interpret these procedures, and *a fortiori*, brains). It is legitimate, though at first sight bizarre, to remark that developmental psychology is a study of how a P-Individual comes to be correlated with a vehicle which is a developing M-Individual. Odd though it sounds, this concept turns out to be useful, though it has not yet been properly exploited.

Equally, we may regard a P-Individual in equilibrium with (inhabiting, replicating in) a different sort of M-Individual. Some example are a collection of human beings in an institution or society (i.e. a cult), some plexus of human beings and mechanical data retrieval systems in a library or a government; and, in the purest and most thoroughly investigated case, a vehicle such as a motor car or an aeroplane designed for moving around in space. Quite literally, a driver regarded as a P-Individual comes to terms with the perceptual-motor/effector-receptor facilities of his motor car which is an M-Individual very different from the body with limbs he uses to walk down a street.

The argument can be inverted to show that the (sub)theory of process makes use of various facilities (the most advanced is the Course Assembly System and Tutorial Environment, CASTE). These too, are vehicles with which people drive through knowledge, rather than over space. It is a fascinating fact that this theoretical gambit permits the externalisation of rather subtle modes of discovery; even more (given that the entailment structure is modified by the participants), to a reification of the idea that learning is symbolic evolution.

6. The (sub)theory of *uncertainties* and *values* characterises certain macroscopic or molar-level measures upon the types of transaction (for example, explanations of explanations) symptomatic, in a conversation, of the (reproductive) processes that make up and give integrity to the 'P-Individuals' noted in (2). If a conversation is *strict*, in the sense of (1), then these are uncertainty/information indices (Chapter 1) since the goal relation is known to an external observer; otherwise, if the goal is determined by an innate preference, they are also indices of value. In either case, the measures are multidimensional (with minimal components expressing doubt about the focus of attention, about *method* or *means* for achieving a goal and about the 'outcomes' which may be immediately reached).

7. The sub-theory of conscious processes starts off as follows. The operating region (Chapter 7) is identified with a region in this 'space' of uncertainty/information index 'dimensions'. It is defined if, and only if, the conversation is genuine, i.e. if it is associated with consciousness. That, of course, in no way *explains* conscious phenomena. Nor, for that matter, does another aspect of the study which identifies information transfer with the synchronisation of otherwise asynchronous procedures (those of (1)), since 'information' was used with two quite different meanings in the last two sentences. But an attempt is made to unify these concepts by relating consciousness to a special interaction between a P-Individual as in (2) and one or more M-Individuals (5) which, from a commonsensical point of view, is where we might expect to find it.

8. The (sub)theory of compatibility bears upon such issues as conflict resolution, cognitive fixity and cognitive dissonance (or, at a more particulate level upon interference effects). It is a kind of immunology concerned with the recognition, as similar or dissimilar, of certain P-Individuals or components of P-Individuals executed in the context of one conversational domain (as in (3)) and it closely resembles the immune-response dynamics of biological organisations (hence, the name). In this case, of course, the competing entities are procedures and it is tempting to call this facet of the enquiry a (very primitive) immunology of ideas.

2 Some Applications of the Theory

Apart from the specific cases cited in section 1, several more global applications are of interest and potential significance though, at this point, empirical

support is either spaı se or altogther lacking. For instance, the theory predicts forms and levels of responsibility (notably in government, management, and so on) as a generalisation of the distinctions noted in section 1.5(4) and in Chapter 11. As part of the same theme, it predicts regulatory strategies likely to prove effective in the management of ecosystems. It is also possible, within the general framework, to comment usefully upon problems of architecture and urban development.

On a very different tack, the distinction between P-Individuals and M-Individuals casts doubt upon the usually accepted approximation that samples (as in a questionnaire survey) really are quasi-independent so that head counting is justifiable. On the one hand, it is maintained that such sampling operations, though useful because of an ingrained disposition to act as though a one to one M/P correspondence existed, have little validity when that condition does not (quite often) apply. Conversely, it is possible to explain the *actual* reliability of appropriate small-sample measurements (whether in the laboratory, for experiments with few, carefully handled, subjects or in the real life context of surveys; for example, the small sample data in Chapter 1). This orientation has repercussions, also, in the area of large system modelling (the 'world dynamics' model), or the on-line computation system of Beer (1973), 'cyberstride', which, although not yet fully exploited, is more in the cybernetic idiom.

Appendix A

(1) Let $i = 1, \ldots, n$ be an index variable (that is, a quantity assuming one and only one value at once) running over a set of elements, $s_i \in S$, and pointing them out uniquely (there is a one to one correspondence between the values of i and n elements of S).

Let $p_1, \ldots, p_n$ be probabilities so that $1 \geqslant p_i \geqslant 0$ and so that $\sum_i p_i = 1$. The uncertainty/information measure over those elements of S indexed by i is

$$H(s) = -\sum_{i=1}^{i=n} p_i \log p_i; \quad \text{for} \quad s_i \in S \tag{1}$$

which gives the *information*; the negative of this quantity is an *uncertainty*.

If it happens that a set S is a set of alternatives (for example, of initially exclusive and exhaustive events) then i runs over all (not just n) elements.

(2) Let x and y be index variables and let $p(x)$, $p(y)$ stand for general designators of p values associated with their distinct values x_i, y_j; $i = 1, \ldots, n$; $j = 1, \ldots, m$. Using a shorthand due to Garner (1962).

$$H(x) = -\sum_x p(x) \log p(x) \equiv H(x) = -\sum_{i=1}^{i=n} p(x = x_i) \log p(x = x_i)$$

or

$$H(y) = -\sum_y p(y) \log p(y) \equiv H(y) = -\sum_{j=1}^{j=m} p(y = y_j) \log p(y = y_j)$$

These quantities are maximised if the p values are all equal and minimised if one is unity, and the others (by definition) zero. Thus in the case of x (for y, similarly) the maximum value, for n unchanging equally likely alternatives, is

$$H^*(x) = -\sum \frac{1}{n} \log \frac{1}{n} = \log n$$

Further $\quad H^*(x) \geqslant H(x) \geqslant 0$

It is certainly true that the values of x and y circumscribe a possible universe; it may be the universe U_α, U_β; in which case these quantities are identical with $H(\alpha)$ and $H(\beta)$. But this need not always be so; for example, if x and y are state variables of a system, then X (the set of values of x) and Y (the set of values of y)

are regarded as indices over the same universe U_α. Either interpretation is legitimate; in the latter case, however, the observer is explicitly talking about relation $R \subset X \times Y \subseteq U_\alpha$ which may be due to undetermined or other than deterministic connections. Thus, either by juxtaposing two distinct universes U_α, U_β or by considering X and Y as alternative sets over the same universe, it is possible to assign probability numbers to joint events designated by pairs $\langle x,y \rangle \in X \times Y$; namely, the joint event probabilities $p(x,y)$. The joint uncertainty/information index, relevant to the selection of $\langle x,y \rangle$ in U_α is

$$H_\alpha(x,y) = -\sum_x \sum_y p(x,y) \log p(x,y)$$

and if x, y index separate universes U_α, U_β then this is identical with $H(\alpha,\beta)$. Any correlation between occurrences indexed by x and occurrences indexed by y reduces the value of this quantity. Its maximum value occurs in the absence of correlation and is

$$H^*_\alpha(x,y) = H_\alpha(x) + H_\alpha(y)$$

$$\text{or} \quad H^*(\alpha,\beta) = H(\alpha) + H(\beta) = H^*(U_\alpha \cup U_\beta)$$

The degree of coupling is obtained as a difference.

$$T_\alpha(x,y) = H^*_\alpha(x,y) - H_\alpha(x,y) = H_\alpha(x) + H_\alpha(y) - H_\alpha(x,y)$$

or

$$T(\alpha,\beta) = H(\alpha) + H(\beta) - H(\alpha,\beta) \tag{2}$$

(3) The former interpretation, $T_\alpha(x, y)$, is best regarded as a *coupling* and the latter, $T(\alpha, \beta)$, as a *transmission*. Though the forms are identical, 'coupling' exists between features of a *system*; regarded on other grounds, as coherent or unitary. 'Transmission' takes place between systems regarded, on other grounds, as distinct. In either case, the term in question is positive or zero.

$$H^*_\alpha(x,y) \geqslant T_\alpha(x,y) \geqslant 0$$

$$H^*(\alpha,\beta) \geqslant T(\alpha,\beta) \geqslant 0$$

If the value of $T_\alpha(x,y)$ is greater than zero then x and y are *not* independent (as supposed, in speaking of an *existent* relation between them); if $T(\alpha,\beta) > 0$ then U_α and U_β are *not* independent. Conversely, if $T_\alpha(x,y) = 0$ or $T(\alpha,\beta) = 0$ then, up to this moment in an observation, no dependency has been demonstrated.

The *redundancy* is simply the degree of 'internal transmission' between possible universes regarded as 'the same' (for example, state variables or states of the same system) and is a measure of *constraint*.

$$\text{Redundancy} \triangleq T(x,y) = H^*(x,y) - H(x,y)$$

or if x, y are unique indices on 'the same' universe then

$$H^*_\alpha = H_\alpha(x) + H_\alpha(y)$$

$$H_\alpha = H_\alpha(x,y)$$

$$\text{Redundancy} \triangleq H^*_\alpha - H_\alpha = T(\alpha, \alpha) \text{ (not } T(x,x) \text{ or } T(y,y)).$$

which is usually normalised to yield the specific measure

$$Z = \frac{H^*_\alpha - H_\alpha}{H_\alpha} = 1 - \frac{H_\alpha^*}{H_\alpha} \quad (3)$$

(Until further notice, the subscripts α, β, are omitted.)

(4) Conditional uncertainty/information/indices are derived from conditional probabilities of the form $p(y|x = x_i)$ meaning 'the probability of each value of y (stated separately for all m values) if one specific value of $x = x_i$ is given'. The selective uncertainty/information, given $x_i = x$, is

$$H(y|x = x_i) = -\sum_y p(y|x = x_i) \log p(y|x = x_i)$$

and the general conditional uncertainty/information is obtained as a weighted mean.

$$H(y|x) = -\sum_x p(x)\, H(y|x = x_i)$$
$$= -\sum_x p(x) \sum_y p(y|x) \log p(y|x)$$

The converse term $H(x|y)$ is obtained in the same manner and the following identities are demonstrable

$$H(x,y) = H(x) + H(y|x) = H(y) + H(x|y)$$
$$T(x,y) = T(y,x) = H(y) - H(y|x) = H(x) - H(x|y)$$

For more than two variables (or universes) the total constraint is more complex. It is designated $T^*(v,x,y)$ for variable v, x, y and is specified for any finite number of variables.

$$T^*(v,x,y) = H^*(v,x,y) - H(v,x,y)$$
$$= H(v) + H(x) + H(y) - H(v,x,y)$$

The specific couplings (transmissions) are terms such as

$$T(v, \langle x,y \rangle)$$

representing the coupling between values of v and pairs of values $\langle x,y \rangle$ of x and y. On decomposing the total constraint pairwise (recalling that $T(x,y) = T(y,x)$) the following equalities are obtained:

$$T^*(v,x,y) = T(v, \langle x,y \rangle) + T(x,y)$$
$$= T(x, \langle v,y \rangle) + T(v,y)$$
$$= T(y, \langle v,x \rangle) + T(v,x)$$

so that, in summary

$$T^*(v,x,y) = T(v,x) + T(v,y) + T(x,y) + Q(v,x,y)$$

where the T terms are couplings between pairs of variables and $Q(v,x,y)$ is an interaction (which may be positive, negative or zero) representing the trivariate

coupling, that cannot be expressed pair-wise (i.e. without considering the 'paired' coupling between 'one variable' and 'a pair of other variables').

$$Q(v,x,y) = T(v,x) + T(v,y) + T(x,y) - T^*(v,x,y) \quad (4)$$

For an m variable system, the expression has m-tuple, $(m - 1)$-tuple, . . . terms. If the variables are indices on universes U_α, U_β, U_γ that are held to be potentially distinct, then $Q(v,x,y) = Q(\alpha,\beta,\gamma)$.

(5) Von Foerster notes that redundancy (Z of equation 3) is one measure of organisation, and that a system is rightly regarded as self organising only if the rate of change of redundancy is positive (that is $dZ/dt > 0$). Considering the statistical quantities H^* and H as variables in a statistical metalinguistic model of the system, differentiate the expression for Z with respect to time, thus obtaining

$$\frac{dZ}{dt} = -\frac{H^*\left(\frac{dH}{dt}\right) - H\left(\frac{dH^*}{dt}\right)}{H^{*2}}$$

For $H^* > 0$ (that is, fully ordered systems are excluded) the condition that $dZ/dt > 0$ is satisfied if and only if

$$H\frac{dH^*}{dt} > H^*\frac{dH}{dt}$$

of which there are two special cases (for one or other variable unchanging) namely

(a) If $\frac{dH^*}{dt} = 0$, then $0 > \frac{dH}{dt}$

(b) If $\frac{dH}{dt} = 0$, then $\frac{dH^*}{dt} > 0$

Case (a) corresponds to adaptation in a fixed universe. For example, in the universe of input/output behaviour (x indexing input, y indexing output) in a fixed field of contingencies or relations U_α, where

$$H^*_\alpha = H^*_\alpha(x) + H^*_\alpha(y)$$
$$H_\alpha = H_\alpha(x,y)$$

The rate of change is approximated by considering as distinct the universes $U_\alpha(t)$, $U_\alpha(t + \Delta t)$. The self-organising condition is secured if, for any Δt, chosen the behavioural indices satisfy

$$H_{\alpha,t}(x,y) > H_{\alpha,t+\Delta t}(x,y)$$

Case (b) is of greater interest, since it entails the idea that U_α does change, $U_\alpha(t) \rightarrow U_\beta(t + \Delta t)$ as, for example, by changing the field of attention. The required condition is secured if, for any value of Δt, the maximum uncertainty/information/index values (signifying possible behavioural relations) satisfy

$$H^*_{\beta,t+\Delta t} > H^*_{\alpha,t}$$

For example, this transformation of H^* may be performed by changing the *learnable* relations between x and y. In some cases but not all (see Chapter 5 onwards), it is permissible to express this operation as changing the structure of *one* universe $U = U_\alpha \equiv U_\beta$ and thus transforming only $H^*_{t+\Delta t} > H^*_t$. This is so, for instance, if $U_\alpha \subset U_\beta$. Conversely, however, the transformation always *can* be expressed as an operation upon the universe: $U_\alpha \rightarrow U_\beta$. If H is to remain constant as required by condition (b) the transformation must usually be instantaneous (i.e. very rapid).

The general case is of greater interest and has two paradigms:

(a) For all $\alpha, \beta, t, t + \Delta t$: $U_\alpha \subset U_\beta$ so that $U = U_\beta$ and H^* is altered by reducing the constraint between x and y in U, as a result of which more relations can be learned. It is, as above, *possible* to express the *same* operation as a transformation $\alpha \rightarrow \beta$. Since there is usually no way of metricising the effect of relaxing constraint (though the *sense* of the *change* in H^* is determined) this operation is carried out in a steady-state system (see Chapter 7) *until* $0 > dH/dt$ and *if* dH/dt approaches the value 0.

(b) For some $\alpha, \beta, t, t + \Delta t$, U_α and U_β are not directly comparable, in the sense that an external observer's images of U_α and U_β do not necessarily agree with the *system's* image. In this case (see Chapter 9 onwards) the transformation $\alpha \rightarrow \beta$ that regulates H^* must be under the system's control though it may still be instigated by the same conditions on the rate of change of H.

Appendix B

Consider a test item with two alternative answers $\{a,b\}$. The student wishes to maximise his expected test score and the tester wishes to extract all available information about the student's present state of knowledge. Instead of requiring a simple 'yes—no' response to each alternative answer, the tester can seek more information about the student by asking him to assign probability numbers to the alternatives so as to represent his current state of doubt or degree of belief that a given alternative is the correct answer. In the case of two alternatives, the student is required to assign values $P(a)$ and $P(b)$ such that $P(a) = 1 - P(b)$. Clearly such responses are potentially more informative than simple 'yes—no' responses, since in the 'yes—no' case the student is restricted to only two possible response forms *either* $P(a) = 1$ and $P(b) = 0$ *or* $P(a) = 1$ and $P(b) = 0$.

Given a set of probability numbers as responses, the tester has to compute a score. Assuming the probability numbers *do* represent a student's degree of belief, he may, for example, sum the values the student has assigned to the correct alternatives, perhaps also weighting items for their relative importance. This procedure is analogous to that of counting the number of correct responses in the 'yes—no' case, but suffers from a serious deficiency; the tester cannot validly assume the assigned probability numbers do represent the student's degree of belief, since a student who is a 'good statistician' will realise that when such a scoring scheme is employed, he can maximise his expected score by assigning probability numbers that do *not* represent his degree of belief. This is illustrated by the following example.

If the student's actual degree of belief that alternative '*a*' is the correct answer is 0·75 and he follows instructions, setting $P(a) = 0{\cdot}75$ and $P(b) = 0{\cdot}25$, he can expect a score of 0·75 with probability 0·75 and a score of 0·25 with probability 0·25. This gives an *expected* score of $(0{\cdot}75 \times 0{\cdot}75) + (0{\cdot}25 \times 0{\cdot}25) = 0{\cdot}625$. However, the student can make his expected score larger by assigning a larger probability number to alternative '*a*'. In fact, he maximises his expected score by assigning $P(a) = 1{\cdot}00$. This gives an expected score of $(0{\cdot}75 \times 1{\cdot}00) + (0{\cdot}25 \times 0{\cdot}00) = 0{\cdot}75$. In general, the 'good statistician' can maximise his expected score by assigning a value of 1·00 if his degree of belief that an alternative is correct is greater than 0·50, and by assigning a value of 0·00 if his degree of belief that an alternative is incorrect is less than 0·50. If his degree of belief is 0·50, it makes no difference what value he assigns since all values yield equal expected scores.

What is required is a scoring system where the student's optimal strategy is to assign probability numbers that do reflect his degree of belief. Shuford, Massengill and Albert (1966) derive the necessary and sufficient conditions for a scoring system that satisfies this criterion for the case of two alternatives and show how to create a virtually inexhaustible number of such systems. In addition, they give partial results for the case of more than two alternatives.

An example scoring system for the case of two alternatives is one which specifies:

(1) that the student receives a score of $1 - (1 - P(a))^2$ points if he assigns $P(a)$ as his degree of belief that alternative 'a' is correct when it is in fact correct, and

(2) that he receives $1 - (P(a))^2$ if he assigns $P(a)$ to alternative 'a' when it is, in fact, incorrect.

Thus, if as before, the student has a degree of belief of 0·75 that alternative 'a' is correct and assigns $P(a) = 0{\cdot}75$, his expected score is $0{\cdot}75 \times (1 - (1 - 0{\cdot}75)^2) + 0{\cdot}25(1 - (1 - 0{\cdot}25)^2) = 0{\cdot}811$. Assigning $P(a)$ less than or greater than 0·75 leads to a lower expected score.

The following scheme (mentioned in the text) can be employed for the case of more than two alternatives and has the advantage over other such schemes that the student's score depends only on the value he assigns to the correct alternative.

Given the correct answer from a set of exclusive alternatives is the ith alternative and the student's assigned value to the ith alternative is P_i, this scheme specifies that:

(1) if P_i is greater than or equal to 0·1, he receives $1 + \log P_i$ points, and
(2) if P_i is less than 0·1 he receives -1 points.

Condition (2) provides a cut off point that gives a point scale running from -1 to $+1$ (rather than $-\infty$ to $+1$). Thus, for extreme values of P_i, some information is lost. In any practical applications, this loss in accuracy is insignificant.

Appendix C

1 Fuzzy Operations (Zadeh)

For a thorough description the reader is referred to Goguen (1968 and Zadeh (1973). The following notes and definitions are chiefly intended to indicate the kind of operation we have in mind and to convince the diffident reader that the concept of a fuzzy operation is not, itself, fuzzy (or vague).

A fuzzy subset X of a universe U (here a set of objects, see footnote on the first page of Chapter 1) is defined by a function $C_X(u)$ or $C_X : U \rightarrow$ {Interval 1, 0) that assigns a 'grade of membership' to each element of U. Those elements of U that are associated with positive values (C) of $C_X(u)$ are called the *support* of X. A fuzzy unit set $\{x\}$ has one support $u \in U$ and a positive grade of membership $C_{\{x\}}$ (U). Using a slash notation to designate grades of membership, i.e. C/u. A non-fuzzy unit set is $1/u$. X itself is the union (written $\underline{UN}$, Zadeh uses the integral to comprehend indefinitely large unions) of certain fuzzy unit sets. That is

$$X \triangleq \underset{u \in \text{support}}{\underline{UN}} (C_x(u)/u)$$

The complement of X is

$$Not\ X \triangleq \underset{u \in U}{\underline{UN}}/(1 - C(u)/u)$$

The union of fuzzy sets X, Y, is

$$Or\ (X,Y) \triangleq \underset{u \in U}{(\underline{UN}/)}[C_X(u)\ \underline{\vee}\ C_Y(u)]/u$$

where $\underline{\vee}$ is the maximal operator;

$$r\ \underline{\vee}\ s = Max\ (r,s) \triangleq \begin{Bmatrix} r \text{ if } r \geqslant s \\ s \text{ if } s > r \end{Bmatrix}$$

The intersection of fuzzy sets X, Y is

$$and\ (X,Y) \triangleq \underset{u \in U}{\underline{UN}}[C_X(u)\ \underline{\wedge}\ C_Y(u)]/u$$

where $\underline{\wedge}$ is the minimal operator

$$r\ \underline{\wedge}\ s = Min\ (r,s) \triangleq \begin{Bmatrix} r \text{ if } s > r \\ s \text{ if } r \geqslant s \end{Bmatrix}.$$

A fuzzy relation, R, is a fuzzy subset of a cartesian product; if R relates X to Y then

$$R \triangleq \underset{x \times y}{\underline{UN}}\, C_R(x,y)/\langle x,y\rangle : x \in X, y \in Y.$$

If R and S are fuzzy relations between X, Y and Y, Z their composition is

$$R \bigcirc S \triangleq \underset{x \times z}{\underline{UN}}\, Supremum\, (C_R(x,y) \,\underline{\wedge}\, C_S(y,z),/\langle x,z\rangle$$

where 'supremum' is used in the usual sense of 'collection of maximal elements' of a relation and $C_R(x,y)$ is a grade of membership function of two or more variables.

One special case of a fuzzy relation is the fuzzy cartesian product of universes U, V with elements u, v namely,

$$X \otimes Y \triangleq \underset{U \times V}{\underline{UN}}\, c_X(u) \,\underline{\wedge}\, C_Y(v)/\langle u,v\rangle$$

Any non fuzzy (standard) conditional: 'If α, then β else η' is equivalent (since 'if . . . then' is an implication $\rightarrow$), to the Boolean expression $(\alpha \rightarrow \beta) \wedge (\alpha^{\sim} \rightarrow \eta)$ where the '$\sim$' stands for Boolean negation. By extension, the *fuzzy conditional* is defined (for the same or different universes, U, V and *fuzzy* sets W, X, Y), as

$$\text{'If } X \text{ then } Y \text{ else } W\text{'} \triangleq X \otimes Y \text{ or } (not\,(x) \otimes W)$$

which expresses, in executable or intensional form, a fuzzy relation, R, from U to V and
which is generalised to image the (standard) extended conditional form

'If X_1 then Y_1 else if X_2 then Y_2 else . . . if X_n then Y_n'

A *fuzzy algorithm* is a non-fuzzily indexed (i.e. well ordered) sequence of instructions of this form (or the degenerate forms' unconditional imperative') and 'assignments' using, in execution, the major inference rule (for a fuzzy relation R from domain X to codomain Y) which is

$$Y = X \bigcirc R$$

The execution is *numericised* if each supremum is resolved by ranking (using any consistent ranking rule) to select a unique outcome.

2 Finite-Function Machines (Von Foerster)

These comments establish the salient distinction between finite-state machines and finite-function machines; the serious reader is (as usual) referred to the original papers.

The characterising equations of a finite-state machine are

$$u_{\tau+1} = f(u_\tau, z_\tau)$$
$$v_\tau = g(u_\tau, z_\tau)$$

and may be expanded to eliminate all but a finite history of n inputs and an initial internal state or output state. By this construction, it is possible to avoid direct

reference to any internal state function, f. Thus, supposing g has an inverse $z = \phi(u,v)$, it follows that

$$v_{\tau+1} = g(u_{\tau+1}, f(u_\tau, \phi(v_\tau, u_\tau))$$

or that

$$v_\tau = g^1(u_\tau, u_{\tau-1}, v_{\tau-1})$$

but (from the last equation)

$$v_{\tau-1} = g^1(u_{\tau-1}, u_{\tau-2}, v_{\tau-2});$$

which, if substituted in the last equation, gives

$$v_\tau = g^2(u_\tau, u_{\tau-1}, u_{\tau-2}, v_{\tau-2})$$

or, for n stages

$$v_\tau = g^n(u_\tau, u_{\tau-1}, \ldots, u_{\tau-n}, v_{\tau-n})$$

which is a *recursive function* (the behavioural history) containing only one reference to an operation or computation: namely $\langle u_{\tau-n}, v_{\tau-n}\rangle$ (or, equivalently, the internal state at $\tau - n$: namely $z_{\tau-n} = \phi(u_{\tau-n}, v_{\tau-n})$).

Turning to a finite-function machine, its characterising equations are

$$f_{\tau+1} = F(u_\tau, f_\tau)$$
$$g_\tau = G(u_\tau, f_\tau)$$

and are thus isomorphic with the finite-state machine equations under the correspondences given below.

Next internal state function $\longleftrightarrow$ Internal state

Output state function $\longleftrightarrow$ Output state

If we apply the steps which lead to the expression for a finite-state machine, stripped of direct reference to internal states (in terms of a recursive function g^n), we obtain

$$g = G^n((u_\tau, u_{\tau-1}, \ldots, u_{\tau-n}), (g_{\tau-1}, g_{\tau-2}, \ldots, g_{\tau-n}))$$

which is a *recursive functional*.

The important difference is that the recursive functional contains explicit reference to a history of operations (the several $g_{\tau-\tau}$) whereas the recursive function does not (just one $v_{\tau-n}$). This is the salient distinction between a finite-state machine and a finite-function machine. Only the latter contains the computations needed to reconstruct an image of its history.

Appendix D

Specification 1 **Model I Simulation**

Nodes in the environment. Sum = 256. Nodes are indexed by a number k or by coordinates α, β, equivalent to k (as in Fig. 22). Each node designates a food store.

Food store. The amount of food at node k at trial n is $u_k(n)$, maximum value 128.

Food increment. For $u_k(n)$ less than 128, the amount of food added at trial n is $\Delta u_k(n)$, which is a function of $O_k(n)$, the average occupancy of node k over the previous 8 trials and a parameter c (with values 1, 2, or 4 determined by the experimenter). This function is: $\Delta u_k(n) = c$, if $O_k(n) = 0$ or 1; $\Delta u_k(n) = c(1 + O_k(n))$ if $O_k(n) > 1$.

Occupancy. The number of automata at node k at trial n is Occupancy$_k(n)$. The average value of Occupancy$_k(n)$ over the previous 8 trials is $O_k(n)$.

Motions of an automaton indexed i. Consider automaton i at node $k = \alpha, \beta$, at trial n. As in Fig. 22, it can remain where it is or move into any of the positions $\alpha - 1, \beta + 1; \alpha, \beta + 1; \alpha + 1, \beta + 1; \alpha - 1, \beta; \alpha + 1, \beta; \alpha - 1, \beta - 1; \alpha, \beta - 1; \alpha + 1, \beta - 1$; provided that it is not at the edge of the field. These motions are indexed with a number, ρ, between 0 and 8 (0 designating the act of remaining where it is).

Sensory capabilities. Automaton i senses the value of u at any node to which it can move. Hence it has 9 sensory inputs indexed by ρ.

Stomach content. The content of the stomach of automaton i at the start of trial n is $\theta_i(n)$, maximum value 128. If i is at node k, then $\theta_i(n)$ is incremented by $\Delta\theta_i(n) = \eta \cdot u_k(n) - \theta_i(n)$ if $u_k(n) > \theta_i(n)$. If $\theta_i(n) > u_k(n)$, the value of $\theta_i(n)$ is unchanged. $1 > \eta > 0$.

Age of i. The age of an automaton is the number of trials since it entered the simulation.

Aging and metabolism. For each trial the value of $\theta_i(n)$ is decremented by an amount F(age of i). This function has the following form (though several functions were used in the experiments):

F(age of i)	1	1	1	1	1	1	1	1	2	2	4	4	8	8	16	16	32	64
Age of i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	>16	>24

Decay. Automaton i is deleted from the simulation at trial n if $0 > \theta_i(n)$.

Experience weights. Let $\theta_i|p$ denote an increment of stomach food contingent upon one of the 9 motions indexed by p. Automaton i builds up a vector $G_i(n)$ of 9 weights $g_{ip}(n) = \sum \Delta\theta_i|p / \sum$ all $\Delta\theta_i$, the summation being taken over the entire history of the automaton.

Selecting motions. At trial n, positioned at node k, the automaton i has a sensory input of 9 u values, which are denoted $u_{ip}(n)$ relative to node k. It examines the weighted sensory input, having components $g_{ip}(n) \cdot u_{ip}(n)$, and selects the maximum if it exists. If no maximum exists, it selects an arbitary member of the maximal subset. The automaton moves in whichever direction is associated with the selection it has made. Thus if $g_{i1}(n) \cdot u_{i1}(n)$ is the selected maximum, then the automaton moves in direction 1 (i.e. from node $k \equiv \alpha, \beta$ to node $\alpha - 1, \beta + 1$).

Parameter P_i. The value of $P_i(n)$ is the average number of other automata that have occupied the same node as automaton i over the previous 8 trials ($P_i(n)$ is an external control variable).

Reproduction. If automaton i and automaton j occupy the same node k at trial n and if $\theta_i(n) > \mu$ and $\theta_j(n) > \mu$ then an automaton l is created at node k. An amount of food γ is subtracted from the stomach of i and the stomach of j and l is credited with an initial quantity of food, 2γ. Normally $\gamma = \frac{1}{4}\mu$ and $\mu = 64$.

Inheritance. The initial value of $G_l(n)$ is $\frac{1}{2}(G_i(n) + G_j(n))$. The initial value of $P_l(n)$ is $\frac{1}{2}(P_i(n) + P_j(n))$.

Processing. The automata are processed at trial n according to the magnitude of the $\theta_i(n)$. If several $\theta_i(n)$ are of the same value, the automata concerned are processed in an arbitrarily chosen order.

Density control mechanism. The reproductive parameter $\mu = P_i(n)/50$ for automaton i at trial n.

Parallel simulation. Since $u_k(n)$ is modified while an automaton at node k is processed, it is necessary to simulate parallel processing when determining the motion of an automaton. For this purpose a COPY ARRAY is made of the values of $u(n)$ for all 256 nodes at the beginning of trial n. From the COPY ARRAY the program (when processing node k) derives a SENSORY FIELD (k, n), which consists in the 8 values of u that belong to the 8 nodes accessible to an automaton located at node k. These values (rather than the prevailing values of $u(n)$) are used in determining the motion of an automaton.

Storage allocation. A node is constructed as a set of k-indexed words specifying $u_k(n)$, Occupancy$_k(n)$, and the list of k occupants at trial n. It has locations for the temporary storage of SENSORY FIELD and certain statistical data. Automaton i is a set of i-indexed words that specify $\theta_i(n)$, $G_i(n)$, $P_i(n)$, age of i, $\rho_i(n)$, and the name of the location automatons at trial n.

Specification 2 **Model II Simulation**

Nodes in the environment. Number and connectivity as in Specification 1.

Food store and incrementation.

$$u_k(n) = C\cdot \text{Occupancy}_k(n-4) + C\cdot \text{Occupancy}_k(n-3) + C\cdot \text{Occupancy}_k(n-2) + C\cdot \text{Occupancy}_k(n-1).$$

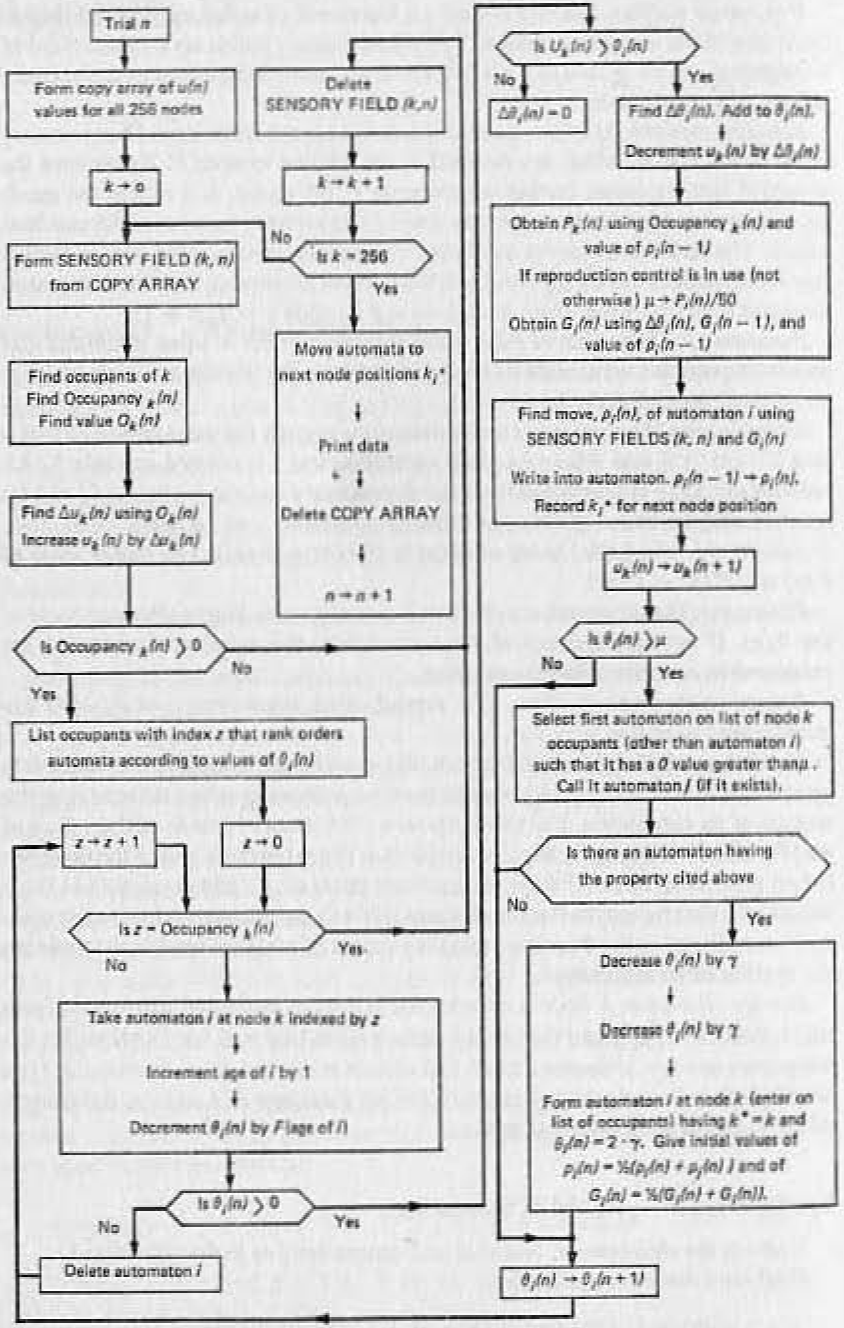


Figure 107 Flow-chart for Model I Simulation.

The automata do not 'eat' separately. The available food is equally divided between the occupants of a node and an automaton receives the YIELD of a node.

Occupancy. As in Specification 1; the average occupancy, however, is computed over only 4 trials. (For details, consult the flow chart of Figs. 107 and 108.)

Types of automata. There are 4 types of automata, denoted $-$, $|$, $+$, $\times$, and their possible motions are shown in Fig. 22.

Sensory capabilities. An automaton can sense the value of u at any node to which it can move. Since different 'types' of automata have different permitted motions, they also have different sensory fields.

Stomach content. Similar to Specification 1. As in Specification 1, however, the value of $\Delta\theta_i(n)$ is the same for all automata residing at node k at trial n. This value is the amount of food at node k divided by the number of automata residing at this node.

Age of i. As in Specification 1.

Aging and metabolism. As in Specification 1, except that the function F(age of i) can be differently defined for each type of automaton.

Decay. As in Specification 1.

Experience weights. These do not exist in Model II simulations.

Selecting motions. As in Specification 1, except that automaton i has no experience weights. Hence, at trial n, it selects a direction of motion associated with the maximum among its sensory inputs (if no maximum exists, it selects a motion randomly from among the maximal subsets).

Parameter P_i. This parameter does not exist in Model II simulations.

Reproductions. The conditions for reproduction are those described in Specification 1, but the results of reproduction depend on the types of automata that encounter one another and satisfy these conditions at the trial concerned. A pair of alternative reproduction rules, R_1 and R_2, are defined, and one of these is used in a particular experiment (unless stated otherwise, R_1 is employed). Using o to denote a combination of automata, p to denote the probability of a particular outcome, and $\Rightarrow$ to denote the progeny produced as an outcome, R_1 and R_2 are defined as follows.

$$
\begin{aligned}
R_1 = {} & -o- \Rightarrow - \\
& |o| \Rightarrow | \\
& -o| = |o- \Rightarrow p = \tfrac{1}{2}(+), p = \tfrac{1}{2}(\times) \\
& +o+ \Rightarrow + \\
& \times o \times \Rightarrow \times \\
& +o\times = \times o+ \Rightarrow p = \tfrac{1}{2}(+), p = \tfrac{1}{2}(\times) \\
& +o- = -o+ \Rightarrow \text{Infertile} \\
& \times o- = -o\times \Rightarrow \text{Infertile} \\
& +o| = |o+ \Rightarrow \text{Infertile} \\
& \times o| = |o\times \Rightarrow \text{Infertile}
\end{aligned}
$$

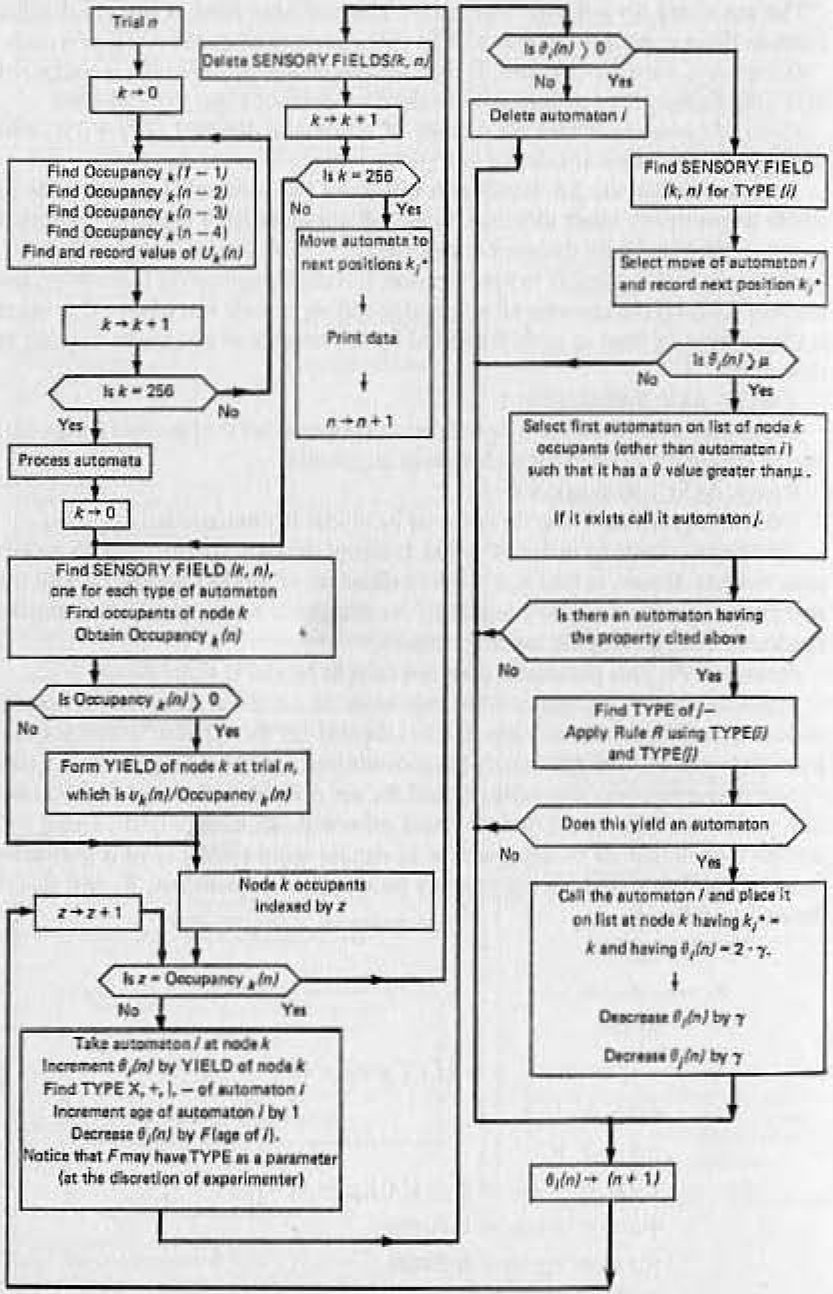


Figure 108 Flow-chart for Model II Simulation.

$$
\begin{aligned}
R_2 = -o- &\Rightarrow - \\
|o| &\Rightarrow | \\
-o| = |o- &\Rightarrow p(+) = \tfrac{1}{2}, p(\times) = \tfrac{1}{2} \\
+o+ &\Rightarrow p(+) = \tfrac{1}{2}, p(\text{Infertile}) = \tfrac{1}{2} \\
\times o \times &\Rightarrow p(\times) = \tfrac{1}{2}, p(\text{Infertile}) = \tfrac{1}{2} \\
+o\times = \times o+ &\Rightarrow p = \tfrac{1}{4}(+), \\
&\quad p(\times) = \tfrac{1}{4}, p(\text{Infertile}) = \tfrac{1}{2} \\
+o- = -o+ &\Rightarrow \text{Infertile} \\
\times o- = -o\times &\Rightarrow \text{Infertile} \\
+o| = |o+ &\Rightarrow \text{Infertile} \\
\times o| = |o\times &\Rightarrow \text{Infertile.}
\end{aligned}
$$

Inheritance. Apart from the type determination entailed by R_1 and R_2, there is no carryover from a given generation to the next generation. Indeed, there are no properties G or P to inherit.

Processing. The automata are processed in an arbitrarily selected order.

Parallel simulation. Since $u_k(n)$ is not modified in the course of processing node k, the COPY ARRAY is unnecessary and the current values of $u(n)$ are computed by the program at the beginning of a trial. SENSORY FIELD is derived from this collection of values, only there are 4 different SENSORY FIELDS, one for each type of automaton, namely ×, +, |, −.

Storage allocation. A node is a set of k-indexed words specifying $u_k(n)$ and the 4 previous occupancies used in computing the value of $u_k(n)$. An automaton is a set of i-indexed words that specify $\theta_i(n)$, age of i, and the location, at trial n, of the automaton.

Glossary

The items are arranged in an order conducive to rational development. The list is short enough to be scanned readily and the looked-for-item discovered without difficulty.

Set S A collection of unordered unitary elements. Membership in the collection is written '$\in$' so that $s \in S$ means 'element s is a member of set S'.

Subset X a 'portion of a set' (written, $X \subset S$) 'either a portion of or all of, a set' (written, $X \sqsubseteq S$). If the elements of subsets are specified explicitly the elements in question are enclosed in brackets (curly brackets, to distinguish them from *ordered* sets as below). For example, if $S = \{a,b,c,\}$ then some subsets are $X = \{a,b\}$ and $Y = \{a,c\}$.

Power Set The set of all subsets of a set (usually, including the *null* or *empty* set). For example, if S has elements $\{a,b,c\}$ then the Power set of S is $\{a\}$, $\{b\}$, $\{c\}$, $\{a,b\}$, $\{a,c\}$, $\{b,c\}$, $\{a,b,c\}$, *Null*.

Intersection, Union If X and Y are subsets of S then $X \cap Y$ is the *intersection* of X and Y (the subset of elements of S that belong to X *and* to Y) whilst $X \cup Y$ is the *union* of X with Y (the subset of elements of S belonging *either* to X *or* to Y *or both*).

Complement with respect to S The complement of X is $S - X$; the elements of S that do not belong to X.

Disjoint Subsets of S are subsets, X, Y ($X \subset S$, $Y \subset S$) such that $X \cap Y$ is void. Hence, $X \sqsubseteq S = Y$ and $Y \sqsubseteq S = X$.

Property That which is common to the members of a set (for example, if S is the set of things, X of red things, then X is the property 'redness').

Cartesian Product for two or more sets, S and T, their Cartesian product, $S \times T$, is the collection of all ordered pairs $\langle s,t \rangle$ with one element taken from S, and one from T (for example, all pairs of men and women). Ordered *pairs* are often called 2-tuples; this generalises, for an *ordered set of n elements*, to n-tuples $\langle s_1 \ldots s_n \rangle \in S_1 \times \ldots \times S_n$.

Relation R A relation, given as a listing, is a subset $R \subset S \times T$ (for example, if R is 'married to', then R contains all those men and women who are husband and wife). R may also be represented by a mapping or series of linkages between the elements of S and T. In a monogamous society R is a one to one mapping. Generally, R may consist in any one, one; one, many; many, one; or many, many mapping and it may involve all or only some of the elements (those involved are called R's *domain* and *codomain* or *range*). In general, also $R \subset S_1 \times \ldots \times S_n$, and n is called the adicity of the relation (thus $R \subset S \times T$ is a 2-adic or binary relation).

Function F A kind of relation; either a one to one or many to one mapping with specified domain. If the domain is less than an (understood) set the function is a 'partial' function. Since F is specified as a relation $F \subset S_1 \times \ldots \times S_n$, n is the adicity of the function (or, equivalently, F is a function of n variables).

Variable A named indexing device that assumes but one value at once. For example, a variable u that ranges over the elements s of S; a variable v that ranges over the elements t of T. The values of u denote elements of S, the values of v denote elements of T; in general, a vector $\langle u_1 \ldots u_n \rangle$ of n variables index the elements of $S_1 \times \ldots \times S_n$ and its values stand for n-tuples $\langle s_1 \ldots s_n \rangle \in S_1 \times \ldots \times S_n$.

Classes of Functions Functions are often classified by their domain/range for example, 'Boolean Functions' of a variable assuming values $\{0,1\}$ (or 'True/False') and having possible values $\{0,1\}$; an integer-arithmetic function (of a variable with integer values). Functions are also classified according to the number of variables needed to index their domain. Thus $F(u)$ is a 1 place function.

Assertoric Logic (In contrast to command logics and question logics considered in the text). A language together with rules for usage (syntax) and inference (see below). Statements in the language are given semantic interpretations; commonly as denoting relations and conditions that hold in one or more sets.

Proposition p (in a language) A statement like 's is red' or 's is married to t' and interpreted, if true, as '$s \in X$' or '$\langle s,t \rangle \in R \subset S \times T$'.

Predicate P An adjective. Equisignificantly, the name for a (usually partial) function from a set S to a set of truth values. Commonly, the truth value set is taken to be $\{1,0\}$ (indicating 'true' and 'false') but this is not always the case (see, for example, the Fuzzy sets in Appendix C). If u is a variable indicating members, s, of a set, S, then $P(u) = True$, is a *one place* predicate statement, meaning that the pointed at element has the *property* named by P. If P is used to name $X \subset S$ then X is (as before) a property and $P(u)$ is true for all elements S (pointed at by u) that are members of X. A 2-place predicate $Q(u,v)$ stands for a 2-adic relation. $Q(u,v)$ is true when u and v point at elements, $\langle s_0,t_0\rangle$, of S and T such that $\langle s_0,t_0\rangle$ is a married couple). In general, there are n-place predicates $Q(u_1 \ldots u_n)$.

Conjunction For propositions p,q written ,'$p \wedge q$' meaning that '*p and q*' is true if p and q are both true.

Disjunction (unqualified and inclusive) written '$p \vee q$' meaning that '$p \vee q$' is true if either p or q or both are true.

Negation The proposition Not-p. If p asserts that $s \in X$ then Not-p asserts that $s \in S - X$, the complement of X in S.

Exclusive Disjunction For propositions, p, q, r, the exclusive disjunction written '$p \veebar q \veebar r$' means, if true, that one and only one of p, q, r is true. Consequently, exclusive disjunctions are often known as alternative sets, and are used for example, to represent alternative answers to a question.

Implication $p \rightarrow q$, means that *if* p is true, *then* q is true.

Equivalence $p \rightarrow q$ and $q \rightarrow p$; written $p \equiv q$. For example, $p \rightarrow q \equiv$ Not $p \vee q$.

Inference Typical inference rules are the primary rule (Modus Ponens of classical logic); if $p \rightarrow q$ and 'p' is true then it is legitimate to infer that q is true; and Modus Tollens (If '$p \rightarrow q$' is true and q is false then it is permissible to infer the falsity of p). Inference Rules figure as metastatements about the language employed to accommodate the logical system and are necessarily augmented by other (somewhat obvious) metatheoretical manipulative permissions; for example, substitution and replacement and logical truism or tautology (if q is derived by substitution and replacement from p then the statement $p \rightarrow q$ is a tautology).

If then else conditional imperative 'If A then B, else C' (where A, B, C are expressions in an action or programming language) means, when the label attached to this statement is encountered:
'If condition A is satisfied, bring about (do, secure or evaluate) B; if not, bring about C' (and, having done so, proceed to the next label).
The execution of a conditional imperative satisfies or brings about a function or a relation (it invariably *does* so, though other effects may be more obtrusive). So does the execution of an algorithm or program.

Assignment Statement (written $u \Rightarrow v$ or $u \Rightarrow 5$) meaning 'give u the value of v' or 'give u the value of 5'.

Program An (ordered) set of assignment statements, and conditional imperative statements, which is open to execution, as a process with a well specified beginning and ending.

Recursive Definition A non viciously circular definition, such as the following definition of 'oddments'.

Base X is an *oddment*; Y is an *oddment*.

Recursive Part An *oddment* placed *on* an *oddment*, is an *oddment*. For example, X on Y, or X *on* (X *on* Y) or Y *on* X *on* . . . (X *on* Y).

Exclusion or uniqueness Requirement Nothing other than these is an oddment.

It is assumed that whoever, or whatever, employs this definition is able to recognise X, Y and to perform the 'operation' *on* and to recognise the result of applying *on*.

Recursively or constructively specified functions

Base Recursive Functions are simple functions, namely, *Constant valued* functions (like $f(x) =$ *Constant*, for all values of x) *or simple order* functions (like the successor function, $f(x) = x + 1$ and functions that search an index or move an object in a constant manner; for example $f(x) =$ (*next above* x) or *Identity Functions* mapping a (possibly complex) element into a property or itself; thus $f(x) = x$, or $f(\langle a,b,c\rangle) = b$.

Recursive Part If composition, written 'o', means doing one thing after another then any composition of simple functions is a recursive function; for example, $f \, o \, g$, or $f \, o \, f \, o \, g$.

Any class of functions of which the member functions are orderable (by a *simple order* function) and are generated from some ancestor function by a common operation, is a recursive function.

Any functions of m variables generated by substitution, with constant value assignment, from a recursive function of $m + 1$ variables is a recursive function.

Exclusion Rule No other functions are recursive functions.

Compiler A program for translating programs, expressed conveniently in a 'high level' and humanly understandable (but, all the same 'formal') language such as ALGOL, or FORTRAN or PLANNER, into a machine code that is interpreted at a mechanical level (and is fussy enough to defeat human patience). No detailed knowledge of high level or machine languages is required; the interested reader may refer to B. Higman's *A Comparative Study of Programming Languages* (MacDonald, 1967).

Miscellaneous Mathematical terms A graph is a structure consisting of *nodes* (drawn as points) and connecting *arcs* that join some or all of the nodes. In a *directed graph* the arcs are directional and often called *edges*. Both nodes and edges are labelled (for example, nodes may represent members of a group; edges, paths of communication between the members). Formally a directed graph is a binary relation on a set of nodes. Though graphs are often invoked, familiarity with them is not essential. The interested reader should consult C. Berge's *The Theory of Graphs* (Methuen, 1962) and Busaker and Saaty's *Finite Graphs and Networks* (McGraw-Hill, 1965). Occasionally we refer to *Games* with the meaning given in *game theory* or *Decision Theory*. Again, little background knowledge is necessary, but readers intrigued by Game Theory will find Luce and Raiffa's *Games and Decisions* (Wiley, 1957) still the most readable and informative reference.

Miscellaneous Statistical Terms Used in discussing experimental results The following comments are superficial but easily augmented by reference to any elementary statistics text book. The *mean* of a set of observations is their average (arithmetic mean) value. The *variance* is an index of scatter about the mean (usually the *standard deviation*, abbreviated as SD is cited; this quantity is the square root of the variance). A significance level (cited as a probability or a percentage) is attached to any serious comment or finding; the usual values are 1% and 0·1% (or $p = 0{\cdot}01$ and $p = 0{\cdot}001$). Thus an assertion that any numerical quantity is significant at the 0·1% level (or $0{\cdot}001 \geqslant p$) means that the quantity might have assumed this value haphazardly or randomly with 1 chance in 1000 Statistical tests are employed in stating hypotheses about absolute values, *differences* between mean values, and *trends* (for example, 'the difference is significant at the 0·1% level'). *Correlation* coefficients measure the undirected (other than random) association between quantities; once again, correlations are quoted together with a properly calculated level of significance.

References

ACKOFF, R. (1973). Science in the systems age. *Operations Research*, May/June, 661–71.

ALEKSANDER, I. (1973). Random logic nets: stability and adaption. *Int. J. for Man-Machine Studies*, **5**(1): 115–32.

AMAREL, S. (1969). Representation of problems and goal directed procedures in computers. Bulletin ASC 1 No. 2, 8–35.

ANGEL, E. S. and G. A. BEKEY (1968). Adaptive finite models of manual control systems. *I.E.E.E. Trans. on Man-Machine Systems*, March 1968: 15–21.

APTER, M. (1966). *Cybernetics and Development*. Pergamon Press.

ARBIB, M. A. (1964). *Brains, Machines and Mathematics*. McGraw-Hill.

ASHBY, W. R. (1962). The self reproducing system. In: *Aspects of the Theory of Artificial Intelligence*. C. E. Muses (ed). Plenum Press.

ASHBY, W. R. (1964a). Constraint analysis. In: *General Systems Yearbook*, Vol. 9.

ASHBY, W. R. (1964b). The set theory of mechanism and homeostasis. In: *General Systems Yearbook*, Vol. 9. Reprinted in: *Automaton Theory and Learning Systems*, D. J. Stewert (ed), 1967: 23–51. Academic Press.

ASHBY, W. R. (1964c). *An Introduction to Cybernetics*. 2nd edn. Chapman and Hall.

ASHBY, W. R. (1968). Some consequences of Bremmerman's limit for information processing systems, pp. 69–76. In: *Cybernetic Problems in Bionics*. H. C. Oestricher and D. R. Moore (eds). Gordon and Breach.

ASHBY, W. R. (1969). Tables of identities governing information flows within large systems. *Commun. ASC*. **1**: 2.

ASHBY, W. R. (1970). Information flows within co-ordinated systems, pp. 57–64. In: *Progress of Cybernetics*. J. Rose (ed). Gordon and Breach.

ATKINSON, R. C. (1969). Models for memory. In: *Programmed Learning Research*. F. de Brisson and M. de Montmollin (eds). Dunod.

ATKINSON, R. C. and R. M. SHIFFRIN (1967). *Human Memory: A Proposed System and Its Control Processes*. Technical Report, 110. Stanford University.

BACH, E. (1964). *An Introduction to Transormational Grammars*. Holt, Rinehart and Winston.

BAKER, J. (1969). The uncertain student and the understanding computer. In: *Programmed Learning Research*. Proceedings of the NATO Symposium on Major Trends in Programmed Learning Research, Nice, 1968. Dunod.

BANERJI, R. B. (1970). *Theory of Problem Solving*. Elsevier, New York.

BANERJI, R. B. (1971). Some linguistic and statistical problems in pattern recognition, pp. 409–19. In: *Pattern Recognition*. Vol. 3. Pergamon Press.

BANNISTER, D. and J. MAIR (1968). *The Evaluation of Personal Constructs*. Academic Press.

BAR-HILLEL, Y. (1964). *Language and Information*. Addison Wesley.

BARICELLI, M. (1962). Numerical testing of evolution theories. *Acta Biotheoretica*, **16**: 69–126.

BARRALT-TORRIJOS, J. (1973). *A Programmatic Interpretation of Combinatory Logics*, (thesis), School of Information and Computer Science, Georgia Institute of Technology.

BARRALT-TORRIJOS, J. and L. CHIARAVIGLIO (1971). *On the Combinatory Definability of Hardware*. Report NSR-GN-655 (Feb.), School of Information Science, Georgia Institute of Technology.

BARRON, R. L. (1968). Self organising and learning control system. In: *Cybernetic Problems in Bionics*. H. L. Oestricher and D. R. Moore (eds). Gordon and Breach.

BATESON, C. (1972). *Our Own Metaphor*. Knopf.

BATESON, G. (1972). *Steps to an Ecology of Mind*. Intertext Books.

BATESON, G., D. JACKSON, J. HALEY and J. WEAKLAND (1956). Towards a theory of schizophrenia. *Behav. Sci.*, **1**: 251–69.

BEER, S. (1966). *Decision and Control*. Wiley.

BEER, S. (1973). *Fanfare for Effective Freedom*. 3rd Richard Goodman Memorial Lecture, Brighton Polytechnic, Feb 14.

BELBIN, R. M. (1969). *The Discovery Method in Training*. Training Information Paper H5, Dept. of Employment, HMSO.

BELL, G. I. (1973). Predator-prey equations simulating an immune response. *Math. Biosci.*, **16**(3–4): 291–315.

BELLMAN, R. (1957). *Dynamic Programming*. Princeton University Press.

BEURLE, R. L. (1954). *Activity in a Block of Cells Capable of Regenerating Impulses*. R.R.E. Memoranda, 1042 and 1043.

BEURLE, R. L. (1959). Storage and manipulation of information in the brain. *J. Inst. Elect. Engrs.*, **5** (new series).

BONNER, J. T. (1958). *The Evolution of Development*. Cambridge University Press.

BREMMERMAN, H. J. (1962). Optimization through evolution and recombination, pp. 93–106. In: *Self-Organising Systems*. M. Yovits, G. Jacobi and S. Goldstein (eds). Spartan Press.

BRICKER, P. D. (1955). The identification of redundant stimulus patterns. *J. Exp. Psych.*, **49**: 73–81.

BRILLOUIN, L. (1956). *Science and Information Theory*. Academic Press.

BRILLOUIN, L. (1964). *Scientific Uncertainty and Information*. Academic Press.

BROADBENT, D. E. (1971). *Decision and Stress*, Academic Press.

BRUNER, J., J. GOODNOW and G. AUSTIN (1965). *A Study of Thinking*. Wiley.

BURKS, A. W. (ed) (1970). *Essays on Cellular Automata*. University of Illinois Press.

CARNAP, R. (1950). *Logical Foundations of Probability*. University of Chicago Press.

CARNAP, R. and Y. BAR-HILLEL (1953). *An Outline of a Theory of Semantic Information*. MIT Research Laboratory, Electronics Rech. Rep. 247.

CARTLEDGE, J. W. and G. C. REJAC (1970). Simulation of the Tsembega ritual cycle. Term thesis. I.C. 5625, School of Information Science, Georgia Institute of Technology.

CHIARAVIGLIO, L. (1071). *Contribution to Summis Symposium*. Georgia Institute of Technology.

CHIARAVIGLIO, L., H. R. HAGEN and R. C. ROEHKASSE (1972). Abstract digital computers. In: *Annual Progress Report*. School of Information Science, Georgia Institute of Technology.

CHIARAVIGLIO, L., J. POORE and J. BARRALT-TORRIJOS (1971). *On the Combinatory Definability of Software*. Report NSR-GN-655 (March), School of Information Science, Georgia Institute of Technology.

CODD, E. F. (1968). *Cellular Automata*. Academic Press.

CONANT, R. and W. R. ASHBY (1970). Every good regulator of a system must be a model of that system. *Int. J. System Sci.*, **1**(2): 81–97.

GRAIK, K. (1966). In: *The Nature of Psychology*. S. Sherwood (ed). Cambridge University Press.

CROSSMAN, E. R. F. W. (1953). Entropy and choice time: the effect of frequency unbalance on choice response. *Quart. J. Exp. Psych.*, **5**: 41–51.

CROWDER, N. A. (1960). Automatic tutoring by intrinsic programming. In: *Teaching Machines and Programmed Learning*. I. A. A. Lumsdaine and R. Glaser (eds). Nat. Educ. Asson U.S.

DAVIS, I. K. and J. HARTLEY (1972). *Contributions to an Educational Technology*. Butterworth.

DAVIS, M. (1958). *Computability and Unsolvability*. McGraw-Hill.

DEXTER, M. (1972). Unpublished Ph.D. thesis. School of Information Sciences, Georgia Institute of Technology.

DIENES, D. (1972). *Petri Nets*. I.E.T. Memorandum, Open University.

ELMAGHRABY, S. E. (1970). Some network models in management science. In: *The Series Lecture Notes in Operations Research and Mathematical Systems*, 29. Springer-Verlag.

ELSHOUT, T. and M. ELSHOUT (1969). The programmed instruction of problem solving. In: *Programmed Learning Research*. F. de Brisson and M. Montmoullin (eds). Dunod.

ERNST, G. W. and A. NEWELL (1969). *G.P.S.: A Case Study in Generality and Problem Solving*. Academic Press.

EVANS, C. R., T. WILKINSON and P. LAMBERT (1967). *Auditory 'Stabilized Images'; Fragmentation and Distortion of Words with Repeated Presentation*. Report Auto 30, Autonomics Division, National Physical Laboratory, Teddington.

FEIGENBAUM, E. A. and G. FELDMAN (1963). *Computers and Thought*. McGraw-Hill.

FESTINGER, L. (1957). *A Theory of Cognitive Dissonance*. Stanford University Press.
FINETTI, BRUNO DE (1962). Does it make sense to speak of good probability appraisers? pp. 357–64. In: *The Scientist Speculates*. I. J. Good (ed). Heinemann.
FITTS, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *J. Exp. Psych.*, **47**: 381–91.
FOGEL, L. J., A. J. OWENS and M. J. WALSH (1966). *Artificial Intelligence Through Simulated Evolution*. Wiley.
FU, K. S. (1968). *Sequential Methods in Pattern Recognition and Machine Learning*. Academic Press.
GABBAY, D. M. (1969). *A Model Theory for Tense Logic*. Report nos. 1 and 2. Applied Logic Branch. The Hebrew University of Jerusalem.
GAGNE, R. M. (1962). The acquisition of knowledge. *Psych. Rev.*, **69**(4): 355–65.
GAGNE, R. M. (1969). Learning categories and instructional strategies. In: *Programmed Learning Research*. Proceedings of the NATO Symposium on Major Trends in Programmed Learning Research, Nice, 1968. Dunod.
GAINES, B. R. (1968). *Automated Feedback Trainers for Perceptual Motor Skills*. Final Report, Ministry of Defence Contract on Servomechanisms in Operator Training Tasks.
GAINES, B. R. (1969a). Stochastic computers, pp. 66–76. In: *The Encyclopedia of Linguistics Information and Control*. A. R. Meatham and R. A. Hudson (eds). Pergamon Press.
GAINES, B. R. (1969b). Adaptively controlled instruction for a tracking skill. In: *Programmed Learning Research*. Proceedings of the NATO Symposium on Major Trends in Programmed Learning Research, Nice, 1968. Dunod.
GAINES, B. R. (1972). Axioms for adaptive behaviour. *Int. J. Man-Machine Studies*, **4**(2): 169–200.
GALLENTER, E. (ed.) (1959). *Automatic Teaching: The State of the Art*. Wiley.
GILBERT, T. F. (1962). Mathetics, Technology of Education. *J. of Mathetics*, **1**(1): 1–25.
GILSTRAP, L. O. (1971). *Keys to Developing Machines with High Level Artificial Intelligence*. ASMI, New York, April.
GLASER, R. (1962). Some research problems in automated instruction: instructional programming and subject matter structure. In: *Programmed Learning and Computer Based Instruction*. J. E. Coulson (ed). Wiley.
GLUSHKOV, V. M. (1966). *Introduction to Cybernetics*. Academic Press.
GOGUEN, J. A. The Logic of Inexact Concepts. *Synthese* **19**, 325–73.
GREY WALTER, W. (1953). *The Living Brain*. Penguin.
GREY WALTER, W. (1969). Electric signs of expectancy and decision in the human brain. In: *Cybernetic Problems in Bionics*. H. L. Oestreicher and D. R. Moore (eds). Gordon and Breach.
GUILFORD, J. P. (1956). The structure of intellect. *Psych. Bull.*, **53**: 267–93.
HARLOW, H. F. (1959). Learning set and error factor theory. In: *Psychology: a Study of a Science*. Vol. 2. Koch (ed). McGraw-Hill.
HARRAH, D. (1966). *Communication: A Logical Model*. MIT Press.

HARRAH, D. (1973). The logic of questions and its relevance to instructional science. *Instructional Sci.*, **1**(4): 447–68.
HELSON, H. (1964). *Adaptation Level Theory*, Harper and Row.
HESSE, M. (1963). *Models and Analogies in Science*. Sheed and Ward.
HEWITT, C. (1969). PLANNER: a language for proving theorems in robots, pp. 295–301. In: *Proceedings of the International Joint Conference on Artificial Intelligence*. Mitre Corp.
HEWITT, C. (1971) Unpublished Ph.D. thesis, MIT.
HICK, W. E. (1952). On the rate of gain of information. *Quart. J. Exp. Psychol.*, **4**: 11–26.
HOBBS, L. C., D. J. THEIS, J. TRIMBLE, H. TITUS and I. HIGHBERG (eds) (1970). *Parallel Processor Systems, Technologies and Applications*. Spartan Books.
HODGESON, A. and E. BENNET (1967). Structural communication. *Systematics*, **5**(3).
HODGESON, A. (1968). *A Communication Technique for the Future*. Ideas Curriculum Lab. Goldsmiths College, Univ. of London, No. 7.
HOLLAND, J. G. and B. F. SKINNER (1961). *The Analysis of Behaviour*. McGraw-Hill.
HOLT, A. W. (1968). *Final Report for the Information System Theory Project*. Rome Air Development Centre, Contract No. AF30 (602) 4211.
HOLT, A. W. and F. COMMONER (1972). *Events and Conditions*. Information System Theory Project, Applied Data Research Inc., New York, Contract No. DAH–CO4–68–C–0043.
HOMME, L. E. (1966). Contiguity theory: psychological management. *Psych. Rec.*, **16**.
HOWARD, N. (1966a). The theory of metagames. *General Systems Yearbook*, **11**: 167–86.
HOWARD, N. (1966b). The mathematics of metagames. *General Systems Yearbook*, **11**: 187–200.
HOWARD, N. (1973). *Paradoxes of Rationality*. MIT Press.
HUDSON, E. M. (1962). *An Adaptive Tracking Simulator*. I.R.E. Congress on Human Factors in Electronics. Long Beach, California.
HUNT, E. B., J. MARIN and P. J. STONE (1966). *Experiments in Induction*, Academic Press.
HYDEN, H. (1965). Activation of nuclear RNA of neurons and glia in learning, pp. 178–239. In: *The Anatomy of Memory*: 1. D. P. Kimble (ed). Science and Behaviour Books Inc.
IVANHENKO, A. G. and U. LAPPA (1967). *Cybernetic Forecasting Techniques*. Elsevier.
JONCKHEERE, A. R. (1954). A test of significance for the relation between *m* rankings and *k* ranked categories. *Brit. J. Math. Stat. Psych.*, **7**: 93–100.
KAGAN, J. (1966). Developmental studies in reflection and analysis, pp. 487–522. In: *Perceptual Development in Children*. A. H. Kidd and J. L. Ricoire (eds). University of London Press.
KAUFFMAN, S. A. (1971). Cellular homeostasis, epigeneis and replication in

randomly aggregated macromolecular systems. *J. Cyb.* (*Trans. Amer. Soc. for Cyb.*), 1 (Jan–March): 71–96.

KELLEY, C. R. and D. J. PROSIN (1968). *Adaptive Performance Measurement*. Final Report on Contract Nonr. 4986(00) Dunlop and Assoc. Inc. Santa Monica, California.

KELLEY, C. R. and E. J. KELLEY (1970). *A Manual for Adaptive Techniques*. Final Report on Contract Nonr. 4986(00) Dunlop and Assocs. Inc. Santa Monica, California.

KELLY, G. A. (1955). *The Psychology of Personal Constructs*. Vols. 1 and 2. Norton.

KERNER, E. H. (1957). Statistical mechanics of interacting biological species. *Bull. Math. Biophys.*, 19: 121.

KERNER, E. H. (1959). Further considerations of statistical mechanics of biological associations. *Bull. Math. Biophys.*, 21: 217.

KILMER, W. L., W. S. MCCULLOCH and J. BLUM (1969). A model of the vertebrate central command system. *Int. J. Man-Machine Studies*, 1(3): 279–310.

KONORSKI, J. (1962). The role of central factors in differentiation. In: *Information Processing in the Central Nervous System*. I.U.S.P. Leiden.

LAING, R. D., S. PHILLIPSON and A. R. LEE (1966). *Interpersonal Perception*. Tavistock.

LANDA, L. V. (1971). The algorithmicization of instruction. Trans. by V. Bennet and edited by F. Kopstein. Educational Technology Publishers (Ms. copyright).

LEWIS, B. N. (1970). *Decision Logic Tables for Algorithms and Logical Trees*. HMSO. CAS Occasional Paper, 12.

LEWIS, B. N., I. S. HORABIN and C. P. CANE (1967). *Flow Charts, Logical Trees and Algorithms for Rules and Regulations*. HMSO. CAS Occasional Paper, 2.

LEWIS, B. N. and G. PASK (1964). The development of communication skills under adaptively controlled conditions. *Programmed Learning*, 1: 69–88.

LEWIS, B. N. and G. PASK (1965). The theory and practice of adaptive systems, pp. 213–66. In: *Teaching Machines and Programmed Learning*. Vol. 2. *Data and Directions*. R. Glaser (ed). Nat. Educ. Assn. U.S.

LILLY, J. C. (1972). *Programming and Metaprogramming in the Human Biocomputer*. Julian Press.

LILLY, J. C. and D. SHIRLEY (1968). *Perception of Repeated Speech*. I.E.E. NPL Conference on Pattern Recognition. NPL.

LOEFGREN, L. (1968). An axiomatic explanation of complete self-reproduction. *Bull. Math. Biophys.*, 30: 415–24.

LOEFGREN, L. (1972). Relative explanations of systems. In: *Trends in General Systems Theory*. G. Klir (ed). Wiley.

LORENZ, K. (1952). *King Solomon's Ring*. Methuen.

LOUGUET-HIGGINS, C. and D. MICHIE (1971). A holographic model for memory. *Nature*.

LUCE, R. D., R. T. BUSH and E. R. GALANTER (1963). *Handbook of Mathematical Psychology*. Vols. 1 and 2. Wiley.

LUMSDAINE, A. A. (1963). Assessing the effectiveness of instructional programmes. In: *Teaching Machines and Programmed Learning II*. R. Glaser (ed). Nat. Educ. Assn U.S.

MCCULLOCH, W. S. (1965). *Embodiments of Mind*. MIT Press.

MCCULLOCH, W. S. and W. PITTS (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.*, 5: 115–33. Reprinted in *Embodiments of Mind*, W. S. McCulloch, MIT Press, 1965.

MCGILL, W. J. (1963). Multivariate information transmissions, pp. 84–102. In: *Readings in Mathematical Psychology*. R. D. Luce, R. T. Bush and E. R. Galanter. Wiley. Reprinted from *Psychometrika*, 19: 97–116 (1954).

MANNA, Z. (1970). Correctness of non-deterministic programmes. *Artificial Intell.* 1: 1.

MARKLE, S. M. (1960). Programmed instruction in English. In: *Teaching Machines and Programmed Learning I*. Nat. Educ. Assn U.S.

MARKOV, A. A. (1961). *Theory of Algorithms*. Works of the Math. Inst. I.M.V.A. Steklov. Vol. 42. Trans. Israel Program for Scientific Translations. Oldbourn Press.

MATHESON, J. E. (1964). *Optimum Teaching Procedures Derived from Mathematical Learning Models*. Report CS/2, Computer Co-ordinated Systems, Institute of Engineering Economic Systems, Stanford University.

MESAROVIC, M. D. (1962). Self-organisational systems. In: *Self-Organising Systems*. M. Yovitz, G. Jacobi and E. Goldstein (eds). Spartan Press.

MESAROVIC, M. D. (1963). *General Systems*. Proceedings of the 2nd IFAC Conference, Basle.

MILLER, G., E. GALANTER and K. H. PRIBRAM (1960). *Plans and the Structure of Behaviour*. Holt.

MINSKY, M. (1967). *Computation: Finite and Infinite Machines*. Prentice-Hall.

MINSKY, M. (ed.) (1968). *Semantic Information Processing*. MIT Press.

MINSKY, M. and S. PAPPERT (1969). *Perceptrons*. MIT Press.

MYHILL, J. (1970). The abstract theory of self-reproduction, pp. 206–18. In: *Essays on Cellular Automata*. A. W. Burks (ed). University of Illinois Press.

NAGEL, E. and J. R. NEUMANN (1959). *Godel's Proof*. Lund, Humphries and Co.

NEWELL, A., J. C. SHAW and H. A. SIMON (1962). The process of creative thinking. In: *Contemporary Approaches to Creative Thinking*. H. E. Gruber, G. Terrell and M. Wertheimer (eds). Atherton Press.

PASK, G. (1957). A teaching machine for radar training. *Automation Progress*, April: 214–17.

PASK, G. (1958). Electronic keyboard teaching machines. *J. Nat. Assn Educ. Commerce*. Reprinted (pp. 336–49). In: *Teaching Machines and Programmed Learning Vol.* 1. R. Glaser and A. Lumsdaine (eds). Nat. Educ. Assn U.S., 1960.

PASK, G. (1959). Physical analogues to the growth of a concept, pp. 877–922. In: *Mechanisation of Thought Processes*. A. Utley (ed). HMSO.

PASK, G. (1960). The growth process in the cybernetic machine, pp. 765–94. In: *Proceedings 2nd Congress International Association of Cybernetics*, Namur, 1958. Gauthier-Villers.

PASK, G. (1961a). *An Approach to Cybernetics*. Hutchinson.

PASK, G. (1961b). A proposed evolutionary model, pp. 229–54. In: *Principles of Self-Organisation.* H. Von Foerster and G. Zopf (eds). Pergamon Press.

PASK, G. (1967). The Control of Learning in Small Subsystems of an educational system. *IEEE Transactions on Human Factors in Electronics,* **1.** HFE. 8 pp. 78–93.

PASK, G. (1963). The use of analogy and parable in cybernetics with emphasis upon analogies for learning and creativity. *Dialectica,* **17:** 167–202.

PASK, G. (1964a). A proposed experimental method for the behavioural sciences, pp. 171–80. In: *Progress in Bio-Cybernetics.* Vol. 1. N. Weiner and J. Schade (eds). Elsevier.

PASK, G. (1964b). Self-organising systems involved in human learning and performance, pp. 247–335. In: *Proceedings 3rd Bionics Symposium, Dayton, Ohio, 1963.* USAF ASD-TDR-63-946, ASTIA.

PASK, G. (1965). Learning machines. (Survey paper at IFAC Conference, Basle, 1963), pp. 393–411. In: *Automatic and Remote Control.* Butterworths.

PASK, G. (1968). A cybernetic model for some kinds of learning and mentation, pp. 531–85. In: *Cybernetic Problems in Bionics.* M. C. Oestreicher and D. R. Moore (eds). Gordon and Breach.

PASK, G. (1969a). The computer simulated development of populations of automata. *Math. Biosci.,* **4:** 101–27.

PASK, G. (1969b). Strategy competence and conversation as determinants of learning. *Programmed Learning,* Oct., 250–67.

PASK, G. (1970a). The meaning of cybernetics in the behavioural sciences, pp. 15–45. In: *Progress of Cybernetics.* Vol. 1. J. Rose (ed). Gordon and Breach. Reprinted in *Cybernetica* (1970), No. 3: 140–59 and No. 4: 240–50 and in *Artoga Communications* (1971): 146–8.

PASK, G. (1970b). Cognitive systems, pp. 349–405. In: *Cognition: A Multiple View.* L. Garvin (ed). Spartan Press.

PASK, G. (1971a). A cybernetic experimental method and its underlying philosophy. *Int. J. Man-Machine Studies,* **3:** 279–337.

PASK, G. (1971b). Interaction between individuals: its stability and style. *Math. Biosci.,* **11:** 59–84.

PASK, G. (1972). A fresh look at cognition and the individual. *Int. J. Man-Machine Studies,* **4:** 211–16.

PASK, G. (1973a). Models for social systems and for their languages. *Instructional Sci.,* **1**(4): 395–445.

PASK, G. (1973b). *Conversation, Cognition and Learning: a Cybernetic Theory and Methodology.* Open University Monograph (abridged form). In press (Elsevier Press, 1974).

PASK, G. and B. N. LEWIS (1964). *Research on the Design of Adaptive Teaching Systems with a Capability for Selecting and Altering Criteria for Adaptation.* Annual Summary Report, No. 4, USAF Contract No. A F 61(052)–402. System Research Ltd.

PASK, G. and B. N. LEWIS (1967). The adaptively controlled instruction of a transformation skill. *Programmed Learning,* April: 74–86.

PASK, G. and B. N. LEWIS (1968). The use of a null-point method to study the acquisition of simple and complex transformation skills. *Brit. J. Math. Stat. Psych.,* **21**(1): 61–83.

PASK, G. and B. N. LEWIS (1969). The self-organisation of a three person task orientated group, pp. 291–311. In: *The Simulation of Human Behaviour.* Proceedings of the NATO Symposium, Paris, 1967. Dunod.

PASK, G., B. N. LEWIS and R. J. FELDMAN (1965). *A Cybernetic Investigation of Learning and Perception.* Final Report, USAF Contract No. AF61 (052)–640, ASTIA. System Research Ltd.

PASK, G. and G. MALLEN (1969). Some recent advances in adaptively controlled teaching systems, pp. 233–39. In: *Proceedings 5th International Congress on Cybernetics.* Namur: Gauthier-Villers.

PASK, G., G. MALLEN, M. ROBINSON, M. DOWSON and B. C. E. SCOTT (1967). *Uni-dimensional Tracking Skill.* Final Scientific Report, USAF Contract No. AF61 (052) 964. System Research Ltd.

PASK, G., G. MALLEN and B. C. E. SCOTT (1968). *Cybernetic Models for the Learning Process.* Final Scientific Report, USAF Contract No. F61 (052)–67C–0010. System Research Ltd.

PASK, G. and B. C. E. SCOTT (1970). *Learning Strategies and Individual Competence.* Final Report, SSRC Project No. HR983/1. System Research Ltd.

PASK, G. and B. C. E. SCOTT (1971). Learning and teaching strategies in a transformation skill. *Brit. J. Math. Stat. Psych.,* **24:** 205–29.

PASK, G. and B. C. E. SCOTT (1972a). Learning strategies and individual competence. *Int. J. Man-Machine Studies,* **4:** 217–53.

PASK, G. and B. C. E. SCOTT (1972b). *Uncertainty Regulation in Learning Applied to Procedures for Teaching Concepts of Probability.* Final Report, SSRC Project HR 1203/1. System Research Ltd.

PASK, G. and B. C. E. SCOTT (1973). CASTE: a system for exhibiting learning strategies and regulating uncertainties. *Int. J. Man-Machine Studies,* **5:** 17–52.

PASK, G., B. C. E. SCOTT and D. KALLIKOURDIS (1973). A theory of conversations and individuals. *Int. J. Man-Machine Studies,* **5**(4): 321–76.

PASK, G., B. C. E. SCOTT, G. MALLEN and M. DAWSON (1969a). *Comparative Study of Adaptively Controlled and Other Methods of Instruction.* Vols. 1 and 2. Final Report, MOD Contract No. A/70/GEN/9725. Systems Research Ltd.

PASK, G., B. C. E. SCOTT, W. LEWIS and M. ELSTOB (1969b). *Research into the Adaptively Controlled Instruction of Compensatory Tracking Skills.* Final Scientific Report, USAF Contract No. F61 052 68 C 0071. System Research Ltd.

PASK, G. and H. VON FOERSTER (1960). A predictive model for self-organising systems. Part 1. *Cybernetica,* **3**(4): 258–300.

PASK, G. and H. VON FOERSTER (1961). A predictive model for self-organising systems. Part 2. *Cybernetica,* **4**(1): 20–55.

PENROSE, L. S. (1962). On living matter and self-replication, pp. 258–72. In: *The Scientist Speculates.* I. J. Good (ed). Heinemann.

PEREIRA, L. (1973). Unpublished Ph.D. thesis. Institute of Cybernetics, Brunel University.

PETRI, C. A. (1965). *Communication with Automata*. Trans. by Clifford F. Greene Jr. A Supplement to Technical Documentary Report, 1, Prepared for Rome Air Development Center Contract AF30(602) = 3324.

PEW, R. C. (1966). Acquisition of hierarchical control over the temporal organisation of a skill. *J. Exp. Psych.*, **71**(5): 764–71.

PHILLIPS, L. (1970). Course notes for Bayesian statistics. Psychology Dept., Brunel University.

PIERCE, J. R. (1962). *Signal, Symbols and Noise*. Hutchinson.

POLYA, G. (1954). *Mathematics and Plausible Reasoning:* Vol. I, *Induction and Analogy in Mathematics*; Vol. II, *Patterns of Plausible Inference*. Princeton University Press.

PRESSEY, S. L. (1960). A machine for automatic teaching of drill material. In: *Teaching Machines and Programmed Learning*. A. A. Lumsdaine and R. Glaser (eds). Nat. Educ. Assn U.S., 1960.

QUILLIAN, M. R. (1969). The teachable language comprehender: a simulation programme and a theory of language. *Communs ACM*, **12**(8): 459.

RAPAPORT, A. (1967). Escape from paradox. *Sci. Amer.* July: 50–6.

RAPPAPORT, R. (1969). *Pigs for the Ancestors*. Yale University Press.

REITMAN, W. R. (1965). *Cognition and Thought*. Wiley.

RESCHER, N. (1966). *The Logic of Commands*. Routledge and Kegan Paul.

RESCHER, N. and A. URQUHART (1971). *Temporal Logic*. Springer-Verlag.

RESINES, S. (1966). *Mathematical Logic*. UNAM Press.

ROBINSON, M. (1973). Unpublished Ph.D. thesis. Institute of Cybernetics, Brunel University.

ROSENBLATT, F. (1961). *Principles of Neurodynamics, Perceptrons and the Theory of Brain Mechanisms*. Cornell Aeronautical Lab. Report No. VG 1196-G-8.

SCANDURA, J. M. (1970). The role of rules in behaviour; toward an operational definition of what (rule) is learned. *Psych. Rev.*, **77**: 516–33.

SCOTT, B. C. E. (1963). Unpublished Ph.D. thesis. Institute of Cybernetics, Brunel University.

SCOTT, B. C. E. and G. PASK (1973). The adaptively controlled instruction of keyboard skills. A series of three papers submitted to *Int. J. Man-Machine Studies*.

SEIGMAN, P. and W. STAPLETON (1971). *A Methodology Utilising Semantic Information Measures (SIM) for Conversational or Dialogue Experiments*. School of Information Science, Georgia Institute of Technology.

SHANNON, C. E. and W. E. WEAVER (1949). *Mathematical Theory of Communication*. University of Illinois Press.

SHIMURA, M. (1973). Fuzzy sets in rank-ordering objects. *J. Math. Anal. and Appl.* (in press).

SHROEDINGER, E. (1944). *What is Life?* Cambridge University Press.

SHUFORD, E. M. (1965). *Cybernetic Testing*. ESD–TR–65–467, Decision Sciences Lab., L. G. Hanscom Field, Beford, Mass.

SHUFORD, E. H., H. MASSENGILL and G. ALBERT (1966). Admissable probability measurement procedures. *Psychometrika*, **31**: 125.

SIMON, H. A. (1966). Thinking by computers and Scientific discovery and the psychology of problem solving. In: *Mind and Cosmos*. R. G. Colodny (ed). University of Pittsburgh Press.

SKINNER, B. F. (1968). *The Technology of Teaching*. Appleton-Century-Crofts.

SKINNER, B. F. (1969). *Contingencies of Reinforcement. A Theoretical Analysis*. Appleton-Century-Crofts.

SMALLWOOD, R. D. (1962). *A Decision Structure for Teaching Machines*. MIT Press.

SOMERHOFF, G. (1969). The Abstract Characterisation of Living Systems. In: *Systems Thinking*. L. Emery (ed). Penguin. (This paper summarises and updates work from 1950 onwards and is usefully compared with articles by other contributors to the book.)

SPENCER-BROWN, G. (1969). *The Laws of Form*. George Allen and Unwin.

STANISLAND, A. C. (1966). *Patterns of Redundancy*. Cambridge University Press.

STAVRINIDES, A. (1973). Unpublished Ph.D. thesis. Institute of Cybernetics, Brunel University.

STEINBUCH, K. VON (1961). Learning matrices. *Kybernetika*, **3**: 117–24.

STEWART, D. J. (ed) (1967). *Automaton Theory and Learning Systems*. Academic Press.

STOHLUROW, L. M. (1961). *Teaching by Machine*. OE 34010, Cooperative Research Monograph 6. U.S. Department of Health, Education and Welfare.

TANNER, W. P. and J. A. SWETS (1954). A decision making theory of visual detection. *Psych. Rev.*, **61**: 401–09.

TARJAN, R. (1963). Problems of stability in control systems. In: *Optimising and Adaptive Control*. Bollinger, Truxall and Minar (eds). Instrument Soc. Amer.

THORPE, W. H. (1956). *Learning and Instinct in Animals*. Methuen.

TINBERGEN, N. (1953). *Social Behaviour in Animals*. Methuen.

TODA, M. (1962). Design for a fungus eater. *Behav. Sci.*, **7**: 75–87.

TURING, A. M. (1963). Computing machinery and intelligence. In: *Computers and Thought*. E. Feigenbaum and J. Feldmann (eds). McGraw-Hill.

TURING, A. M. (1969). Intelligent machinery. In: *Machine Intelligence*. Vol. 5. B. Meltzer and D. Michie (eds). Edinburgh University Press.

VAN DER VELDT, J. (1928). Cited in Woodworth, R. S. (1950).

VAN GICH, J. P. (1971). Changes in the mental content of work exemplified by lumber sorting operation. *Int. J. Man-Machine Studies*, **3**(1): 12–29.

VITANYI, P. (1973). Sexually reproducing automata. *Math. Biosci.* (in press).

VON FOERSTER, H. (1960). On self-organising systems and their environments. In: *Self-Organising Systems*. M. C. Yovitts and S. Cameron (eds). Pergamon Press.

VON FOERSTER, H. (1971). Molecular ethology. In: *Molecular Mechanisms of Memory and Learning*. C. Ungar (ed). Pergamon.

VON NEUMANN, J. (1969). *Theory of Self-Reproducing Automata*. A. Burks (ed). University of Illinois Press.

WADDINGTON, C. H. (1957). *The Strategy of The Genes*. George Allen and Unwin.

WALD, A. (1950). *Statistical Decision Functions*. Wiley.

WARREN, R. M. (1961). Illusory changes of distinct speech upon repetition—the verbal transformation effect. *Brit. J. Psych.*, **52**(3): 249–58.
WELFORD, A. T. (1968). *Fundamentals of Skill*, Methuen.
WIENER, N. (1948). *Cybernetics* (2nd edn., 1962). Wiley.
WILKINS, L. (1963). *Social Deviance*. Tavistock.
WINOGRAD, T. (1972). *Understanding Natural Language*. Edinburgh University Press.
WINSTON, P. (1970). *Learning Structural Descriptions by Example*. Project Mac TR–76, MIT Press.
WITTGENSTEIN, L. (1953). *Philosophical Investigations*. Trans. G. E. M. Anscombe. Blackwell.
WOODWORTH, R. S. (1950). *Experimental Psychology*. Methuen.
WYNNE EDWARDS, W. C. (1963). *Animal Dispersion*. Oliver and Boyd.
ZADEH, L. A. (1968). Fuzzy algorithms. *Inform. Contr.*, **12**: 94–102.
ZADEH, L. A. (1971). Toward a theory of fuzzy systems. In: *Aspects of Network and System Theory*. R. E. Kalman and N. DeClaris (eds). Holt, Rinehart and Winston.
ZADEH, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. In: *IEE Transactions on Systems, Man and Cybernetics*. Vol. SMC–3, 1, Jan 1973.

Subject Index

The index is selective and (deliberately) incomplete. Certain words or phrases appear often and are omitted; for example, 'Learning', 'Adaptation' (as psychological omissions); 'Feedback', 'System', 'Goal Directed', 'Alternative Set', 'Decision' (as Cybernetic or System Theoretic omissions); and 'Inference', 'Deduction/Induction' (in the field of logic and philosophy). The remaining, less common, words are usually indexed to points of discussion, rather than places where they are mentioned in the text.

Index of Names