Appendix A
Concrete Art Manifesto

In 1930 the Dutch painter Theo van Doesbourg (a pseudonim for Christian Emil Marie Küpper) published the *Manifesto for Concrete Art*, advocating the total freedom of art from the need to describe or represent natural objects or sentiments. The *Manifesto* is reported in Fig. A.1.

The translation of the *Manifesto* is the following one:

BASIS OF CONCRETE PAINTING

We say:

1. Art is universal.
2. A work of art must be entirely conceived and shaped by the mind before its execution. It shall not receive anything of nature's or sensuality's or sentimentality's formal data. We want to exclude lyricism, drama, symbolism, and so on.
3. The painting must be entirely built up with purely plastic elements, namely surfaces and colors. A pictorial element does not have any meaning beyond "itself"; as a consequence, a painting does not have any meaning other than "itself".
4. The construction of a painting, as well as that of its elements, must be simple and visually controllable.
5. The painting technique must be mechanic, i.e., exact, anti-impressionistic.
6. An effort toward absolute clarity is mandatory.

Carlsund, Doesbourg, Hélion, Tutundjian and Wantz.

L. Saitta and J.-D. Zucker, *Abstraction in Artificial Intelligence and Complex Systems*,
DOI: 10.1007/978-1-4614-7052-6,

ART CONCRET

GROUPE ET REVUE FONDÉS EN 1930 A PARIS

PREMIÈRE ANNÉE · NUMÉRO D'INTRODUCTION · AVRIL MIL NEUF CENT TRENTE

BASE DE LA PEINTURE CONCRÈTE

Nous disons :

1° L'art est universel.

2° L'œuvre d'art doit être entièrement conçue et formée par l'esprit avant son exécution. Elle ne doit rien recevoir des données formelles de la nature, ni de la sensualité, ni de la sentimentalité.
Nous voulons exclure le lyrisme, le dramatisme, le symbolisme, etc.

3° Le tableau doit être entièrement construit avec des éléments purement plastiques, c'est-à-dire plans et couleurs. Un élément pictural n'a pas d'autre signification que « lui-même » en conséquence le tableau n'a pas d'autre signification que « lui-même ».

4° La construction du tableau, aussi bien que ses éléments, doit être simple et contrôlable visuellement.

5° La technique doit être mécanique c'est-à-dire exacte, anti-impressionniste.

6° Effort pour la clarté absolue.

Carlsund, Doesbourg, Hélion, Tutundjian, Wantz.

Fig. A.1 *Concrete Art Manifesto*, by Theo van Doesburg (1930)

Appendix B
Cartographic Results for Roads

This appendix shows in Fig. B.1 some more roads and their different representations with and without abstraction. The representations result from:

- a direct *symbolization* (initial),
- the cartographic result produced by the hand-crafted expert system GALBE, specifically developed to generalize road [389, 391],
- the result produced by the set of rules obtained by learning without abstraction,
- the result produced by the set of rules obtained by combining learning and abstraction.

L. Saitta and J.-D. Zucker, *Abstraction in Artificial Intelligence and Complex Systems*,
DOI: 10.1007/978-1-4614-7052-6, © Springer Science+Business Media New York 2013

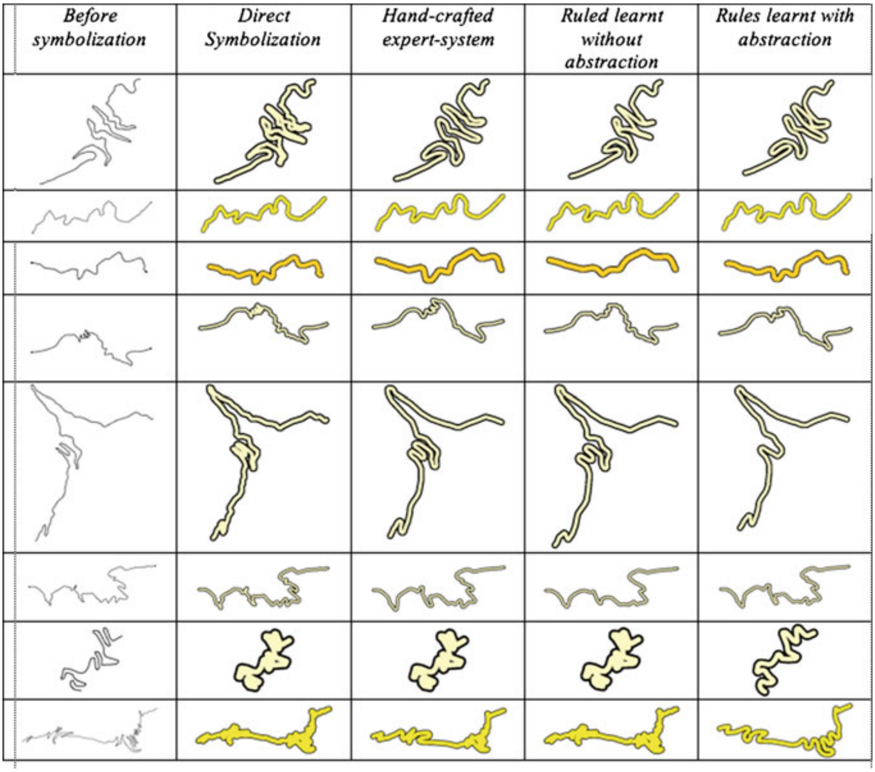

Fig. B.1 Different road generalization results, for different roads. The improvements brought by abstraction are clearly visible

Appendix C
Relational Algebra

In this appendix we recall the basic notions of Relational Algebra for manipulating relational databases. Relational Algebra has been proposed by Ullman [540] as a formal tool for modeling relational database semantics.

Relational databases provide operators for handling *relations* in their extensional form. Given a set $\mathcal{X}$ of variables, a n-ary relation $R(x_1, x_2, \ldots, x_n)$ involving the variables in $\mathcal{X}$ is represented as a table with n columns and k rows, where each row describes an n-ple of individuals of $\mathcal{X}$ satisfying R.

The type T of a relation $R(x_1, x_2, \ldots, x_n)$ is defined as:

$$T : X_1 \times X_2 \times \cdots \times X_n \tag{C.1}$$

where $X_1, X_2, \ldots, X_n$ are the domains from which the individuals bound to x_1, $x_2, \ldots, x_n$ can be taken. The relation $R(x_1, x_2, \ldots, x_n)$ is a subset of its type.

A relational database provides a set of operators that allow one to compute new relations from the existing ones [539]. Operators are usually made available as primitive functions of a query language, which may depend on the specific database implementation. Relational Algebra provides a formal definition of the semantics of these operators, which is independent of the syntax of the query language.

Here, we briefly recall the basic notions of Relational Algebra, whereas a more extensive introduction can be found in [540]. In the following, the list of the basic operators is reported.

Union

Given two relations R_1 and R_2 of the same arity, the union $R = R_1 \cup R_2$ is a relation obtained by taking the union of the tuples occurring either in R_1 or in R_2.

L. Saitta and J.-D. Zucker, *Abstraction in Artificial Intelligence and Complex Systems*,
DOI: 10.1007/978-1-4614-7052-6,

PhD

ID	SURNAME	AGE
23	Smith	38
40	Adams	39
132	Ross	32

MANAGERS

ID	SURNAME	AGE
72	Adams	50
40	Adams	39
132	Ross	32

PhD ∪ MANAGERS

ID	SURNAME	AGE
23	Smith	38
40	Adams	39
132	Ross	32
72	Adams	50

PhD ∩ MANAGERS

ID	SURNAME	AGE
40	Adams	39
132	Ross	32

PhD – MANAGERS

ID	SURNAME	AGE
23	Smith	38

Fig. C.1 Given the tables corresponding to the relations $R_1 = PhD$ and $R_2 = MANAGERS$, we can construct the tables *PhD* ∪ *MANAGERS*, *PhD* ∩ *MANAGERS*, and *PhD-MANAGERS*

Intersection

Given two relations R_1 and R_2 of the same arity, the intersection $R = R_1 \cap R_2$ is a relation obtained by only keeping the tuples occurring in both relations R_1 and R_2.

Set difference

Given two relations R_1 and R_2 of the same arity, the difference $S = R_1 - R_2$ is obtained by eliminating from R_1 the tuples that occur in R_2.

In Fig. C.1 examples of the Union, Intersection, and Set Difference operators are reported.

Cartesian product

Let R_1 and R_2 be two relations of arity n and m, respectively. The Cartesian product $R = R_1 \times R_2$ is a relation of arity $n+m$, whose tuples have been obtained by chaining one tuple of R_1 with one tuple of R_2 in all possible ways.

Projection

Let R_1 and R_2 be two relations of arity n and m, respectively, with $n > m$; the relation R_2 will be called a projection of R_1 if it can be generated by taking the distinct tuples obtained by deleting a choice of $(n - m)$ columns in R_1. The projection is formally written as $R_2 = \pi_{i_1,i_2,\ldots,i_m}(R_1)$, where $i_1, i_2, \ldots, i_m$ denote the columns of R_1 that are to be kept in R_2.

Selection

Let R be a n-ary relation. A selection $S = \sigma_\theta(R)$ is obtained by selecting all tuples in R satisfying a condition θ stated as a logical formula, built up using the usual connectives $\wedge$, $\vee$, $\neg$, the arithmetic predicates $<$, $>$, $=$, $\leq$, $\geq$ and the values of the tuple's components.

PhD

ID	SURNAME	AGE
23	Smith	38
40	Adams	39
132	Ross	32

LOCATION

CITY	REGION
Rome	Lazio
Milan	Lombardia
Bergamo	Lombardia

PhD × MANAGERS

ID	SURNAME	AGE	CITY	REGION
23	Smith	38	Rome	Lazio
23	Smith	38	Milan	Lombardia
23	Smith	38	Bergamo	Lombardia
40	Adams	39	Rome	Lazio
40	Adams	39	Milan	Lombardia
40	Adams	39	Bergamo	Lombardia
132	Ross	32	Rome	Lazio
132	Ross	32	Milan	Lombardia
132	Ross	32	Bergamo	Lombardia

Proj-PhD

SURNAME	AGE
Smith	38
Adams	39
Ross	32

Sel-PhD

ID	SURNAME	AGE
23	Smith	38
132	Ross	32

Fig. C.2 Given the relations $R_1 = PhD$ and $R_2 = LOCATION$, the Cartesian product of R_1 and R_2 contains 9 tuples, obtained by concatenating each tuple in R_1 with each tuples in R_2. Relation *Proj-PhD* is the projection of relation *PhD* over the attributes *SURNAME* and *AGE*, i.e., *Proj-PhD* $= \pi_{SURNAME,AGE}(PhD)$. Finally, relation *Sel-PhD* is obtained by selection from *PhD*, and contains the tuples that satisfy the condition $AGE \leqslant 38$, i.e., *Sel-PhD* $= \sigma_{AGE \leqslant 38}(PhD)$

FATHERHOOD

FATHER	CHILD
John	Ann
Stuart	Jeanne
Robert	Albert

R-FATHERHOOD

PARENT	CHILD
John	Ann
Stuart	Jeanne
Robert	Albert

Fig. C.3 Given the relations $R = FATHERHOOD$, we can rename attribute *FATHER* as *PARENT*, obtaining the new relation *R-FATHERHOOD*, i.e., *R-FATHERHOOD* $= \rho_{PARENT \leftarrow FATHER}(R)$

In Fig. C.2 examples of the Cartesian product, Projection, and Selection operators are reported.

Renaming

If R is a relation, then $R(B \leftarrow A)$ is the same relation, where column A is now named B. The renaming operation is denoted by $R(B \leftarrow A) = \rho_{B \leftarrow A}(R)$. In Fig. C.3 an example of the Renaming operator is reported.

Natural-join

Let R and S be two relations of arity n and m, respectively, such that k columns $A_1, A_2, \ldots, A_k$ in S have the same name as in R. The natural join $Q = R \bowtie S$ is the $(n + m - k)$ arity relation defined as:

$$\pi_{A_1,A_2,\ldots,A_{(n+m-k)}} \sigma_{R.A_1=S.A_1 \wedge R.A_1=S.A_1 \wedge \cdots \wedge R.A_k=S.A_k}(R \times S).$$

AFFILIATION

RESEARCHER	UNIVERSITY
Johnson	Stanford
Ross	MIT
Archer	MIT

RESEARCH

UNIVERSITY	FIELD
Stanford	Law
MIT	Physics
MIT	Informatics
CMU	Informatics

AFFILIATION $\bowtie$ RESEARCH

RESEARCHER	UNIVERSITY	FIELD
Johnson	Stanford	Law
Ross	MIT	Physics
Ross	MIT	Informatics
Archer	MIT	Physics
Archer	MIT	Informatics

Fig. C.4 Given the two relations *AFFILIATION* and *RESEARCH*, their natural join is obtained by considering all tuples that have the *UNIVERSITY* value in common

In other words, each tuple of Q is obtained by merging two tuples of R and S such that the corresponding values of the shared columns are the same.

In Fig. C.4 an examples of the Natural-Join operator is reported.

Appendix D
Basic Notion of First Order Logics

In this appendix we recall the basic notions of First Order Logic (FOL), in particular those that have been used in this book. Readers interested in a deeper understanding of the topic can find excellent introductions in many textbooks (see, for instance, [496, 545]).

First Order Logic (also known as First Order Predicate Calculus) is a language used in Mathematics, Computer Science, and many other fields, for describing formal reasoning. It is an extension of Propositional Logic to the manipulation of variables. The definition of a logical language has two parts, namely the syntax of the language, and the semantic.

D.1 Syntax

A FOL language $\mathcal{L}$ is a 5-tuple $\langle \mathbb{C}, \mathbb{X}, \mathbb{O}, \mathbb{P}, \mathbb{F} \rangle$, where $\mathbb{C}$ is a set of constants, $\mathbb{X}$ is a set of variables, $\mathbb{O}$ is the set of logical operators, $\mathbb{F}$ is a set of function names and $\mathbb{P}$ is a set of predicate names. All symbols occurring in the definition of $\mathcal{L}$ are partitioned into two sets:

Logical symbols—Logical symbols include:

- *Logical connectives*: $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), $\rightarrow$ (implication).
- *Quantifiers*: $\forall$ (universal quantifier), $\exists$ (existential quantifier).
- Parentheses and punctuation symbols.
- An infinite set of variable names. Each variable X takes value in a given domain Ω_X.

L. Saitta and J.-D. Zucker, *Abstraction in Artificial Intelligence and Complex Systems*,
DOI: 10.1007/978-1-4614-7052-6,

Non-logical symbols—Non-logical symbols include:

- *Predicate* symbols. A predicate $p(x_1, \ldots, x_n)$, of arity n, describes an elementary property of, or an elementary relation among sets of objects represented by a set of variables.
- *Function* symbols. A function $f(x_1, \ldots, x_n)$, of arity n, associates to a tuple of objects, represented by the set of variables, a value or another object.
- *Constants*. These are the identifiers of objects, and can be seen as function symbols of 0-arity.

As an example, let us consider the following Language $\mathcal{L} = \langle \mathbb{C}, \mathbb{X}, \mathbb{O}, \mathbb{P}, \mathbb{F} \rangle$, where:

$\mathbb{C} = \{$`John, Mary, Ann, Rob, Tom, Billy, Lawrence, Mia`$\}$ is a set of constants.

$\mathbb{X} = \{x, y, z, \ldots, x_1, x_2, x_3, \ldots\}$ is a set of variables.

$\mathbb{O} = \{\wedge, \vee, \neg, \rightarrow, \forall, \exists\}$ is the set of standard logical operators.

$\mathbb{F} = \{mother(x), father(x)\}$ is the set of functions. Function $mother(x)$ ($father(x)$) assign to x his/her mother (father).

$\mathbb{P} = \{\mathbf{married}(x, y), \mathbf{grandmother}(x, y), \mathbf{siblings}(x, y)\}$ is the set of predicates.

The expression power of the language resides in the possibility of combining the elementary symbols to form complex *terms* and *formulas*.

D.1.1 Formulas

Logical formulas are expressions built up over the dictionary defined by logical and non logical symbols. *Well-formed-formulas* (wffs) are the ones with the syntax recursively defined in the following. We must defined *terms*, and *formulas*.

Terms

- A constant is a term.
- A variable is a term.
- If f is a function symbol of arity n and $t_1, \ldots, t_n$ are terms, $f(t_1, t_2, \ldots, t_n)$ is a term.

Formulas

- If p is a predicate symbol of arity n, and $t_1, t_2, \ldots, t_n$ are terms, then $p(t_1, t_2, \ldots, t_n)$ is an atomic formula.
- If φ_1 and φ_2 are formulas, $(\varphi_1 \wedge \varphi_2)$, $(\varphi_1 \vee \varphi_2)$, $(\varphi_1 \rightarrow \varphi_2)$, are formulas.
- If φ is a formula, then $\neg\varphi$ is a formula.
- If φ is a formula and x is a variable occurring in φ, then $\forall x(\varphi)$ and $\exists x(\varphi)$ are formulas.

Only expressions that can be obtained by finitely many applications of the above rules are formulas.

Frequently, in the literature an atomic formulas is called a *literal*. A literal consisting of a non-negated predicate $p(t_1, t_2, \ldots, t_n)$ is said a *positive literal*, whereas a negated predicate of the type $\neg p(t_1, t_2, \ldots, t_n)$ is said a *negative literal*.

In the language introduced above as an example, terms are, for instance, `Mary`, y, $mother(x)$, and $father(mother(x))$. Moreover, $\mathbf{sibling}(x, y)$, $\mathbf{married}(x, y) \wedge \mathbf{grandmather}(x, z)$, $\neg\mathbf{married}(y, z)$, $\mathbf{married}(mother(x), father(x))$, $\exists x.\mathbf{sibling}(x, y)$, $\forall x\ \exists y.\mathbf{grandmother}(y, x)$ are all well-formed formulas.

D.2 Semantics

FOL formulas make assertions on generic objects denoted by variables. In order to let a formula assume a precise meaning in the description of the world, it is necessary to define an *interpretation*, in which generic objects, represented by variables, can be mapped to specific individuals.

An interpretation is a *universe* U of individuals, together with a set of rules assigning a meaning to formulas with respect to U. More precisely, for atomic formulas we have:

- Constants identify (are associated to) individuals in U.
- Function symbols are associated to operations in U, which build new objects (or values) starting from the primitive ones. In other words, the semantic of a function $y = f(x_1, \ldots, x_n)$ is the set of tuples $(x_1, \ldots, x_n, y)$, where $x_j \in \Omega_j$ $(1 \leqslant j \leqslant n)$, and $y \in \Omega_y$, such that f associates y to the tuple $(x_1, \ldots, x_n)$.
- 0-arity predicates are mapped to *True* or *False*.
- n-ary predicates are mapped to n-ary relations, *i.e*, to set of n-ary tuples of objects existing in U and satisfying the predicate.

In other words, objects, operations, and relations are the *extension* of constants, functions, and predicates, respectively. Among formulas, we have to distinguish between *open* and *closed* formulas. Open formulas are those that contain at least one *free variable*, namely a variable that is not assigned to a specific value. Closed formulas are those that do not contain free variables. A free variable can be closed by either assigning to it a specific constant, or attaching to it a quantifier. For instance, the formula $\mathbf{married}(x, y)$ is open, whereas *siblings*(`John, Mary`) and $\exists x.\mathbf{sibling}(x, Ann)$ are closed ones. Open formulas (called "concepts" by Frege) have an extension associated to them, whereas closed formulas (called "sentences" by Frege) have associated a truth value.

Replacing a variable x by a constant A is called a *substitution* $\theta = x/A$. An atomic formula $q(x/A)$ is true in U if the constant A belongs to the unary relation R_q, corresponding to predicate q.[1] In analogous way, the atomic formula $p(y/B, z/C)$ is

---

[1] With x/A, x/B, y/C we mean that the variables x, y, and z are replaced by the constant values A, B, and C, respectively.

AND

φ_1	φ_2	$\varphi_1 \wedge \varphi_2$
F	F	F
F	T	F
T	F	F
T	T	T

OR

φ_1	φ_2	$\varphi_1 \vee \varphi_2$
F	F	F
F	T	T
T	F	T
T	T	T

NOT

φ	$\neg\varphi$
F	T
T	F

Implication

φ_1	φ_2	$\varphi_1 \rightarrow \varphi_2$
F	F	T
F	T	T
T	F	F
T	T	T

Bi-Implication

φ_1	φ_2	$\varphi_1 \leftrightarrow \varphi_2$
F	F	T
F	T	F
T	F	F
T	T	T

Fig. D.1 Semantics of logical connectives AND ($\wedge$), OR($\vee$), NOT ($\neg$), Implication ($\rightarrow$), and BI-Implication ($\leftrightarrow$)

true iff the tuple (B, C) belongs to the table defining the binary relation R_p, associated to predicate p.

The truth of complex formulas can be evaluated in a universe U by combining the truth of the single atomic formulas according to the classical semantics of the logical connectives (see Fig. D.1). For instance, the formula $\varphi(x, y) = q(x/A) \wedge p(x/A, y/B)$ is true iff A belongs to relation R_q and (A, B) belongs to relation R_p.

By referring to the truth tables reported in Fig. D.1, it is easy to prove that, among the five connectives $\wedge$, $\vee$, $\neg$, $\rightarrow$, and $\leftrightarrow$, only three of them are essential because *implication* and *bi-implication* can be expressed as a combination of the others. For instance, formula $\varphi \rightarrow \psi$ (φ implies ψ), is semantically equivalent to $\neg\varphi \vee \psi$, while formula $\varphi \leftrightarrow \psi$ (φ implies ψ and ψ implies φ) is semantically equivalent to $(\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi)$.

D.3 Clausal Form

In wffs quantifiers can be nested arbitrarily. However, it can be proved that any wff can be syntactically transformed in such a way that all quantifiers occur only at the most external level, while preserving the formula's semantics. This syntactic form is called *prenexed* form. Moreover, the existential quantifier can be eliminated by introducing the so called *Skolem function*.

The prenexed form of a formula can be a universally quantified formula of the type $\forall_{x_1,x_2,\ldots,x_n}.\varphi(x_1, x_2, \ldots, x_n)$, where φ is a formula with only free variables, which is built up by means of the connectives $\wedge$, $\vee$, $\neg$, and, possibly, $\rightarrow$ and $\leftrightarrow$. Finally, any formula, built up through the connectives $\vee$, $\wedge$ and $\neg$, can be represented in *Conjunctive Normal Form* (CNF), i.e., as a conjunction of disjunctions of atoms (literals). In particular, any FOL sentence can always be written as in the following:

$$\begin{aligned}\forall_{x_1,x_2,\ldots,x_n}.[(L_{11} \vee L_{12} \vee \ldots \vee L_{1k_1}) \wedge (L_{21} \vee L_{22} \vee \ldots \vee L_{2k_2}) \wedge \ldots \\ \wedge(L_{m1} \vee L_{m2} \vee \ldots \vee L_{mk_m})]\end{aligned} \tag{D.1}$$

where L_{ij} denotes a positive or negative literal, with any subset of the variables $x_1, x_2, \ldots, x_n$ as arguments.

Form (D.1) is usually referred to as *clausal form* (the word *clause* denotes a disjunction of literals), and is the one most widely used for representing knowledge in Relational Machine Learning.

For the sake of simplicity, notation (D.1) is usually simplified as follows:

- Universal quantification is implicitly assumed, and the quantifier symbol is omitted.
- Symbol $\wedge$ denoting conjunction is replaced by "," or implicitly assumed.

Horn clauses. A Horn clause is a clause with at most one positive literal. Horn clauses are named after the logician Alfred Horn [262], who investigated the mathematical properties of similar sentences in the non-clausal form of FOL. The general form of Horn clauses is then:

$$\neg L_1 \vee \neg L_2 \vee \ldots \vee \neg L_{k-1} \vee L_k, \tag{D.2}$$

which can be equivalently rewritten as

$$\neg(L_1 \wedge L_2 \wedge \ldots \wedge L_{k-1}) \vee L_k \qquad \equiv \qquad L_1 \wedge L_2 \wedge \ldots \wedge L_{k-1} \rightarrow L_k \tag{D.3}$$

Horn clauses play a basic role in Logic Programming [299] and are important for Machine Learning [382]. A Horn clause with exactly one positive literal is said a *definite clause*. A definite clause with no negative literals is also called a *fact*.

DATALOG. *DATALOG* is a subset of a Horn clause language designed for querying databases. It imposes several further restrictions to the clausal form:

- It disallows complex terms as arguments of predicates. Only constants and variables can be terms of a predicate.
- Variables are range restricted, i.e., each variable in the conclusion of a clause must also appear in a non negated literal in the premise.

Appendix E
Abstraction Operators

All operators that we have defined so far are summarized in Table E.1. They are grouped according to the elements of the description frame they act upon, and their abstraction mechanism. Even though there is quite a large number of them, several operators can be "technically" applied in the same way, exploiting synergies. For instance, equating values of a variable can be implemented with the same code for attributes, argument values in functions and relations, and in a function's co-domain. Nevertheless, we have kept them separate, because they differ in meaning, and also in the impact they have on the Γ's.

As it was said at the beginning, the listed operators are defined at the level of description frames, because they correspond to abstracting the observations that are obtained from the sensors used to analyze the world. To each one of them a corresponding method is associated, which acts on specific $\mathcal{P}$-*Sets* according to *rules* that guide the actual process of abstraction.

E.1 Some More Operators

In this appendix some operators are described in addition to those introduced in Chap. 7. The complete set of available operators can be found in the book's companion site.

The introduced operators are by no means intendeded to exhaust the spectrum of abstractions that can be thought of. However they are sufficient to describe most of the abstractions proposed in the past in a unified way. Moreover, they provide a guide for defining new ones, better suited to particular fields.

L. Saitta and J.-D. Zucker, *Abstraction in Artificial Intelligence and Complex Systems*,
DOI: 10.1007/978-1-4614-7052-6, © Springer Science+Business Media New York 2013

Table E.1 Summary of the elementary abstraction and approximation operators, classified according to the elements of the description frame they act upon and their mechanism

Operators	Elements	Arguments	Values
Hiding	ω_{hobj}, ω_{htype}, ω_{hattr}, ω_{hrel}, ω_{hfun}	$\omega_{hfunarg}$, $\omega_{hrelarg}$	$\omega_{hattrval}$ $\omega_{hfunargval}$ $\omega_{hfuncodom}$ $\omega_{hrelargval}$
Equating	ω_{eqobj}, ω_{eqtype}, ω_{eqattr}, ω_{eqfun}, ω_{eqrel}	$\omega_{eqfunarg}$ $\omega_{eqrelarg}$	$\omega_{eqattrval}$ $\omega_{eqfunargval}$ $\omega_{eqfuncodom}$ $\omega_{eqrelargval}$
Building hierarchy	$\omega_{hierattr}$, $\omega_{hierfun}$, $\omega_{hierrel}$, $\omega_{hiertype}$		$\omega_{hierattrval}$ $\omega_{hierfuncodom}$
Combining	ω_{coll}, ω_{aggr}, ω_{group}	ω_{constr}	
Approximating	ρ_{repl} ρ_{idobj}, ρ_{idtype}, ρ_{idattr}, ρ_{idfun}, ρ_{idrel}	ρ_{repl} $\rho_{idfunarg}$ $\rho_{idrelarg}$	ρ_{repl} $\rho_{idattrval}$ $\rho_{idfunargval}$ $\rho_{idfuncodom}$ $\rho_{idrelargval}$

E.1.1 Operator that Hides a Type: ω_{htype}

If $X^{(g)} = \Gamma_{TYPE}^{(g)}$ and $y = \texttt{t}$, type $\texttt{t}$ cannot be anymore observed in a system, and objects that were previously of this type become of type $\texttt{obj}$. We define:

$$\omega_{htype}(\texttt{t}) \underset{def}{=} \omega_h(\Gamma_{TYPE}^{(g)}, \texttt{t})$$

and we obtain:

$$\Gamma_{TYPE}^{(a)} = \Gamma_{TYPE}^{(g)} - \{\texttt{t}\} \cup \{\texttt{obj}\}$$

The corresponding method $\mathsf{meth}_{htype}(\mathcal{P}_g, \texttt{t})$, applied to an observed $\mathcal{P}_g$, replaces with $\texttt{obj}$ the type of all objects of type $\texttt{t}$.

E.1.2 Operator that Hides a Value from a Function's Codomain: $\omega_{hfuncodom}$

If $X^{(g)} = \Gamma_{\mathcal{F}}^{(g)}$, $y = (f_h, CD(f_h))$, $\texttt{v} \in CD(f_h)$, then the operator

$$\omega_{hfuncodom}(f_h, CD(f_h), \texttt{v}) \underset{def}{=} \omega_h(\Gamma_{\mathcal{F}}^{(g)}, (f_h, CD(f_h)), \texttt{v})$$

removes some value $\texttt{v}$ from the codomain of f_h. Then an abstract function is created, whose codomain is given by:

$$CD(f_h^{(a)}) = CD(f_h) - \{\texttt{v}\},$$

and

$$\Gamma_{\mathcal{F}}^{(a)} = \Gamma_{\mathcal{F}}^{(g)} - \{f_h\} \cup \{f_h^{(a)}\}$$

For instance, let us consider the function *Price*, with codomain $CD(Price) = \{\texttt{cheap}, \texttt{moderate}, \texttt{fair}, \texttt{costly}, \texttt{very-costly}\}$; if we want to remove the value `very-costly`, we have to specify, in method

$$\mathsf{meth}_{hfuncodom}(\mathcal{P}_g, Price, CD(Price), \texttt{very-costly}),$$

what happens for those tuples in $FCOV(f_h^{(a)})$ that contain $\texttt{v}$. One possibility is that the value is turned into `UN`.

E.1.3 Operator that Builds Equivalence Classes of Relations: ω_{eqrel}

If $X^{(g)} = \Gamma_{\mathcal{R}}^{(g)}$ and $y^{(a)} = R^{(a)}$, the operator makes indistinguishable all relations R satisfying $\varphi_{eq}(\mathrm{R}_1, \ldots, \mathrm{R}_k)$. Let

$$\Gamma_{\mathcal{R},eq} = \{(\mathrm{R}_1, \ldots, \mathrm{R}_k) | \varphi_{eq}(\mathrm{R}_1, \ldots, \mathrm{R}_k)\}$$

be the set of indistinguishable relations. We define:

$$\omega_{eqrel}(\varphi_{eq}(\mathrm{R}_1, \ldots, \mathrm{R}_k), R^{(a)}) \underset{def}{=} \omega_{eqelem}(\Gamma_{\mathcal{R}}^{(g)}, \varphi_{eq}(\mathrm{R}_1, \ldots, \mathrm{R}_k), R^{(a)})$$

The operator $\omega_{eqrel}(\varphi_{eq}(\mathrm{R}_1, \ldots, \mathrm{R}_k), R^{(a)})$ generates first the set $\Gamma_{\mathcal{R},eq}$, obtaining:

$$\Gamma_{\mathcal{R}}^{(a)} = \Gamma_{\mathcal{R}}^{(g)} - \Gamma_{\mathcal{R},eq} \cup \{R^{(a)}\}$$

It is the method $\mathsf{meth}_{eqrel}(\mathcal{P}_g, \varphi_{eq}(\mathrm{R}_1, \ldots, \mathrm{R}_k), R^{(a)})$ that specifies how the cover of $R^{(a)}$ has to be computed.

As an example, let us suppose that the set of relations to be made indistinguishable be define extensionally, as in the case of functions. For instance, let

$$\Gamma_{\mathcal{R},eq} = \{R_{IsMotherof}, R_{IsStepMotherof}\},$$

where:

$$R_{IsMotherof} \subseteq \Gamma^{(g)}_{\texttt{women}} \times \Gamma^{(g)}_{\texttt{people}}$$
$$R_{IsStepMotherof} \subseteq \Gamma^{(g)}_{\texttt{women}} \times \Gamma^{(g)}_{\texttt{people}}$$

If we state the equivalence between the two relations, we may keep only $R^{(a)}$ in place of the two. Again, method $\mathsf{meth}_{eqrel}(\mathcal{P}_g, \varphi_{eq}(\mathrm{R}_1, \ldots, \mathrm{R}_k), R^{(a)})$ shall specify how the cover $RCOV(R_{IsStepMotherof})$ must be computed.

E.1.4 Operator that Builds Equivalence Classes of Values in a Function's Codomain: $\omega_{eqfuncodom}$

If $X^{(g)} = \Gamma^{(g)}_{\mathcal{F}}$, $Y = (f_h, CD(f_h))$, $V_{eq} \subseteq CD(f_h)$, then the operator equates values of the codomain of function f_h and set all equal to $\mathrm{v}^{(a)}$. We define:

$$\omega_{eqfuncodom}(f_h, CD(f_h), V_{eq}, \mathrm{v}^{(a)}) \underset{def}{=} \omega_{eqval}(\Gamma^{(g)}_{\mathcal{F}}, (f_h, CD(f_h)), V_{eq}, \mathrm{v}^{(a)})$$

An abstract function is defined:

$$f_h^{(a)} \subseteq \underbrace{\Gamma^{(g)}_{\mathcal{O}} \times \ldots \times \Gamma^{(g)}_{\mathcal{O}}}_{1\,\ldots\ldots\ldots\, t_h} \to CD(f_h) - V_{eq} \cup \{\mathrm{v}^{(a)}\}$$

Then:

$$\Gamma^{(a)}_{\mathcal{F}} = \Gamma^{(g)}_{\mathcal{F}} - \{f_h\} \cup \{f_h^{(a)}\}$$

Method $\mathsf{meth}_{eqfuncodom}(\mathcal{P}_g, (f_h, CD(f_h)), V_{eq}, \mathrm{v}^{(a)})$ handles the cover of $f_h^{(a)}$ by replacing in $FCOV(f_h^{(a)})$ all occurrences of members of V_{eq} with $\mathrm{v}^{(a)}$.

For the sake of exemplification, let us consider a gray-level picture, in which the attribute *Intensity* of a pixel x can take on a value in the integer interval [0, 255]. Let τ be a threshold, such that:

$$I^{(a)}(x) = \begin{cases} 255 & \text{if } I(x) > \tau, \\ I(x) & \text{otherwise.} \end{cases} \tag{E.1}$$

In Eq. (E.1) all values greater than the threshold are considered equivalent. An example is reported in Fig. E.1.

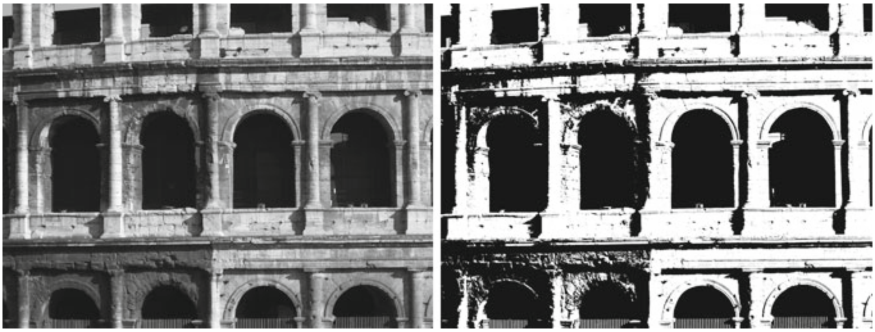

Fig. E.1 Example of method $\mathsf{meth}_{eqfuncodom}(\mathcal{P}_g, (f_h, CD(f_h)), V_{eq}, \mathrm{v}^{(a)})$. The picture on the left is a 256-level gray picture. By a thresholding operation, all pixels whose intensity is greater than τ are considered white

E.1.5 Operator that Builds a Hierarchy of Attributes: $\omega_{hierattr}$

If $X^{(g)} = \Gamma_{\mathcal{A}}^{(g)}$, $Y = \emptyset$, $Y_{child} = \Gamma_{\mathcal{A},child}^{(g)}$, and $y^{(a)} = (A^{(a)}, \Lambda^{(a)})$, then the operator works on an attribute hierarchy, where a set of nodes, those contained in $\Gamma_{\mathcal{A},child}^{(g)}$, are replaced by $(A^{(a)}, \Lambda^{(a)})$. We define:

$$\omega_{hierattr}\left(\Gamma_{\mathcal{A},child}^{(g)}, (A^{(a)}, \Lambda^{(a)})\right) \underset{def}{=} \omega_{hier}\left(\Gamma_{\mathcal{A}}^{(g)}, \Gamma_{\mathcal{A},child}^{(g)}, (A^{(a)}, \Lambda^{(a)})\right)$$

and we obtain:

$$\Gamma_{\mathcal{A}}^{(a)} = \Gamma_{\mathcal{A}}^{(g)} - \Gamma_{\mathcal{A},child}^{(g)} \cup \{(A^{(a)}, \Lambda^{(a)})\}.$$

The method $\mathsf{meth}_{hierattr}(\mathcal{P}_g, \Gamma_{\mathcal{A},child}^{(g)}, (A^{(a)}, \Lambda^{(a)}))$ states how the values in $\Lambda^{(a)}$ must be derived from those in the domains of the attributes in $\Gamma_{\mathcal{A},child}^{(g)}$.

As an example, let us consider the attributes *Length* and *Width*. We introduce the abstract attribute $LinearSize^{(\mathrm{a})}$, such that *Length* **is-a** $LinearSize^{(\mathrm{a})}$ and *Width* **is-a** $LinearSize^{(\mathrm{a})}$. We have then, $\Gamma_{\mathcal{A},child} = \{Length, Width\}$, and $A^{(a)} = LinearSize^{(\mathrm{a})}$. The values of the attribute $LinearSize^{(\mathrm{a})}$ are to be defined; for instance, we may assume that, for an object x,

$$LinearSize^{(\mathrm{a})}(\mathrm{x}) = Max[Length(\mathrm{x}), Width(\mathrm{x})].$$

The original attribute do not enter Γ_a.

E.1.6 Operator that Builds a Hierarchy of Relations: $\omega_{hierrel}$

If $X^{(g)} = \Gamma_{\mathcal{R}}^{(g)}$, $Y = \emptyset$, $Y_{child} = \Gamma_{\mathcal{R},child}^{(g)}$, and $y^{(a)} = R^{(a)}$, then the operator works on a relation hierarchy, where a set of nodes, those contained in $\Gamma_{\mathcal{R},child}^{(g)}$, are replaced by $R^{(a)}$. We define:

$$\omega_{hierrel}\left(\Gamma_{\mathcal{R},child}^{(g)}, R^{(a)}\right) \underset{def}{=} \omega_{hier}\left(\Gamma_{\mathcal{R}}^{(g)}, \Gamma_{\mathcal{R},child}^{(g)}, R^{(a)}\right)$$

and we obtain:

$$\Gamma_{\mathcal{R}}^{(a)} = \Gamma_{\mathcal{R}}^{(g)} - \Gamma_{\mathcal{R},child}^{(g)} \cup R^{(a)}$$

The method $\mathsf{meth}_{hierrel}\left(\mathcal{P}_g, \Gamma_{\mathcal{R},child}^{(g)}, R^{(a)}\right)$ states how the cover of $R^{(a)}$ must be computed starting from those of the relations in $\Gamma_{\mathcal{R},child}^{(g)}$.

As an example, let $R_{HorizAdjacent} \subseteq \Gamma_{\mathcal{O}}^{(g)} \times \Gamma_{\mathcal{O}}^{(g)}$ and $R_{VertAdjacent} \subseteq \Gamma_{\mathcal{O}}^{(g)} \times \Gamma_{\mathcal{O}}^{(g)}$ be two relations over pairs of objects. The former is verified when two objects touch each other horizontally, whereas the latter is verified when two objects touch each other vertically. We introduce the abstract relation $R_{Adjacent}^{(a)} \subseteq \Gamma_{\mathcal{O}} \times \Gamma_{\mathcal{O}}$, which does not distinguish the modality (horizontal or vertical) of the adjacency. In this case we have $\Gamma_{\mathcal{R},child} = \{R_{HorizAdjacent}, R_{VertAdjacent}\}$ and the new relation $R^{(a)} = R_{Adjacent}^{(a)}$. Operator $\omega_{hierrel}^{(\Psi)}(\mathcal{P}_g, \Gamma_{\mathcal{R},child}^{(g)}, R^{(a)})$ will establish that, for instance:

$$FCOV(R_{Adjacent}^{(a)}) = FOCV(R_{HorizAdjacent}) \cup FCOV(R_{VertAdjacent})$$

The original relations are hidden in the abstract space.

E.2 Approximation Operators

In this section we illustrate some additional approximation operators.

E.2.1 Operator that Identifies Types: ω_{idtype}

If $X^{(g)} = \Gamma_{TYPE}^{(g)}$ and $y^{(a)} = \mathtt{t}^{(a)}$, the operator makes all types satisfying $\varphi_{id}(\mathtt{t}_1, \ldots, \mathtt{t}_k)$ indistinguishable. Then type $\mathtt{t}^{(a)}$ is applied to all objects in the equivalence class. We define:

$$\omega_{idtypes}(\varphi_{id}(\mathtt{t}_1, \ldots, \mathtt{t}_k), \mathtt{t}^{(a)}) \underset{def}{=} \omega_{idelem}(\Gamma_{TYPE}^{(g)}, \varphi_{id}(\mathtt{t}_1, \ldots, \mathtt{t}_k), \mathtt{t}^{(a)})$$

The operator $\omega_{idtype}(\varphi_{id}(\texttt{t}_1, \ldots, \texttt{t}_k), \texttt{t}^{(a)})$ generates first the set of $\Gamma_{TYPE,id}$ of indistinguishable types, and then it applies $\texttt{t}^{(a)}$ to the obtained class. All types in $\Gamma_{TYPE,id}$ become $\texttt{t}^{(a)}$, obtaining:

$$\Gamma_{TYPE}^{(a)} = \Gamma_{TYPE}^{(g)} - \Gamma_{TYPE,id} \cup \{\texttt{t}^{(a)}\}$$

It is the method $\mathsf{meth}_{idtype}(\mathcal{P}_g, \varphi_{id}(\texttt{t}_1, \ldots, \texttt{t}_k), \texttt{t}^{(a)})$ that specifies what properties are to be assigned to $\texttt{t}^{(a)}$, considering the ones of the equated types. For instance, if the types in $\Gamma_{TYPE,id}$ have different sets of attributes, $\texttt{t}^{(a)}$ could have the intersection of these sets, or the union, by setting some values to NA, depending on the choice of the user.

As an example, we can consider the types `chair` and `armchair` and we can equate them to be both $\texttt{chair}^{(a)}$.

E.2.2 Operator that Approximates Attribute Values: $\omega_{idattrval}$

If $X^{(g)} = \Gamma_{\mathcal{A}}^{(g)}$, $Y = (A, \Lambda_A)$, and $V_{id} = \Lambda_{A,id} \subseteq \Lambda_A$, then the operator makes indistinguishable a subset $\Lambda_{A,id}$ of the domain Λ_A of A. We define:

$$\omega_{idattrval}((A, \Lambda_A), \Lambda_{A,id}, \text{v}^{(a)}) \underset{def}{=} \omega_{idval}(\Gamma_{\mathcal{A}}^{(g)}, (A, \Lambda_A), \Lambda_{A,id}, \text{v}^{(a)})$$

We obtain an approximate attribute $A^{(a)}$ such that $\Lambda_{A^{(a)}} = \Lambda_A - \Lambda_{A,id} \cup \{\text{v}^{(a)}\}$, and

$$\Gamma_{\mathcal{A}}^{(a)} = \Gamma_{\mathcal{A}}^{(g)} - \{(A, \Lambda_A)\} \cup \{(A^{(a)}, \Lambda_{A^{(a)}})\}$$

For the sake of exemplification, let us consider an attribute, say *Color*, which takes values in the set:
{`white, yellow, olive-green, sea-green, lawn-green, red, pink, light-green, dark-green, blue, light-blue, aquamarine, orange, magenta, cyan, black`}.

We might consider equivalent all the shades of green, and identify them with $\text{v}^{(a)}$ = `sea-green`. In this case, the true shade of green is no more known (see Fig. E.2.

As another important example, let us consider the discretization of real numbers. Let us consider the interval [0, 100), and let us divide it into 10 subintervals $\{[10k, 10(k+1)) \mid 0 \leqslant k \leqslant 9\}$. Numbers falling inside one of the intervals are considered all equal to the mean value $10k + 0.5$.

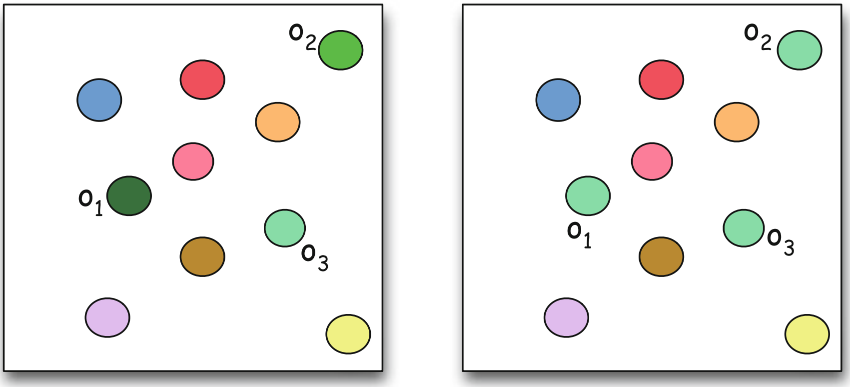

Fig. E.2 Application of method meth$(\mathcal{P}_g, \omega_{eqattrval}\big((Color, \Lambda_{Color}), V_{id}, \mathrm{v}^{(a)}\big)$ to the figure on the *left*. Let V_{id} = {`olive-green, sea-green, lawn-green, light-green, dark-green`}. Objects $\mathrm{o}_1, \mathrm{o}_2$, and o_3 have color `dark-green, lawn-green,` and `sea-green`, respectively. After equating all shades of green to `sea-green`, the color of all three objects becomes `sea-green`. *[A color version of the figure is reported in Fig.* H.16 *of Appendix* H*]*

E.3 Some More Methods

In Chap. 7 the methods associated to some operators have been described. In this section we give some additional examples, whereas the complete set of methods is provided in the book's companion site.

Let us consider the operators that hide an attribute, or a function, or a relation, i.e., $\omega_{hattr}\ ((A_m, \Lambda_m))$, $\omega_{hfun}\ (f_h)$, and $\omega_{hrel}\ (R_k)$. Hiding an attribute or a function or a relation are all instantiations of the same PDT introduced in Sect. 7.2.1 , and then we group them together in Table E.2, whereas their bodies are reported in Tables E.3, E.4, and E.5, respectively.

Also at the description level the operators $\omega_{hattr}((A_m, \Lambda_M))$, $\omega_{hfun}(f_h)$, and $\omega_{hrel}(R_k)$ are similar; in fact, they simply hide from the appropriate set the concerned element (attribute, function, or relation), as it was illustrated in Sect. 7.2.1. But when we must apply them to a specific $\mathcal{P}_g$, some complication may arise. Let us look first at Table E.2.

Operator $\omega_{hattr}((A_m, \Lambda_m))$ hides the attribute from the set of available ones, and, as a consequence, meth$\big(\mathcal{P}_g, \omega_{hattr}(A_m, \Lambda_m)\big)$ hides the value of that attribute in each object in $\mathcal{P}_g$. Hiding an attribute may cause the descriptions of some objects to become identical. However, as each object has a unique identity, they remain distinguishable.

As both functions and relations cannot have an attribute as an argument, removing A_m does not have any further effect. For the hidden information, it is not necessary to store all the tuples hidden in $\mathcal{A}_g$, but only the value of A_m for each object.

Table E.2 Summary of methods $\text{meth}(\mathcal{P}_g, \omega_{hattr}(A_m, \Lambda_M))$, $\text{meth}(\mathcal{P}_g, \omega_{hfun}(f_h))$, and $\text{meth}(\mathcal{P}_g, \omega_{hrel}(R_k))$

NAME	$\text{meth}(\mathcal{P}_g, \omega_{hattr})$	$\text{meth}(\mathcal{P}_g, \omega_{hfun})$	$\text{meth}(\mathcal{P}_g, \omega_{hrel})$
INPUT	$\mathcal{P}_g, (A_m, \Lambda_m)$	$\mathcal{P}_g, f_h$	$\mathcal{P}_g, R_k$
OUTPUT	$\mathcal{P}_a$	$\mathcal{P}_a$	$\mathcal{P}_a$
APPL-CONDITIONS	$A_m \in \mathcal{A}_g$	$f_h \in \mathcal{F}_g$	$R_k \in \mathcal{R}_g$
PARAMETERS	$\emptyset$	$\emptyset$	$\emptyset$
MEMORY	$\Delta^{(\mathcal{P})}$	$\Delta^{(\mathcal{P})}$	$\Delta^{(\mathcal{P})}$
BODY	See Table E.3	See Table E.4	See Table E.5

Table E.3 Pseudo-code for the method $\text{meth}\big(\mathcal{P}_g, \omega_{hattr}(A_m, \Lambda_m)\big)$

METHOD $\text{meth}\big(\mathcal{P}_g, \omega_{hattr}(A_m, \Lambda_m)\big)$
Let $|\mathcal{O}_g| = N$
$\Delta^{(\mathcal{P})} = \emptyset$
for n = 1, N **do**
  $\mathcal{A}_a = \mathcal{A}_g - \{(\circ_n, A_1(\circ_n), ..., A_m(\circ_n), ..., A_M(\circ_n))\}$
  $\Delta^{(\mathcal{P})} = \Delta^{(\mathcal{P})} \cup \{(\circ_n, A_m(\circ_n))\}$
  $\mathcal{A}_a = \mathcal{A}_a \cup \{(\circ_n, A_1(\circ_n), ..., A_{m-1}(\circ_n), A_{m+1}(\circ_n), ..., A_M(\circ_n))\}$
end

Table E.4 Pseudo-code for the method $\text{meth}\big(\mathcal{P}_g, \omega_{hfun}(f_h)\big)$

METHOD $\text{meth}\big(\mathcal{P}_g, \omega_{hfun}(f_h)\big)$
$\Delta_{\mathcal{F}}^{(\mathcal{P})} = \{f_h\}$
$\mathcal{O}_a = \mathcal{O}_g$
$\mathcal{A}_a = \mathcal{A}_g$
if $CD(f_h) = \Gamma_{\mathcal{O}}^{(g)}$
 then $\mathcal{F}_a = \mathcal{F}_g - \{FCOV(f_h)\}$
 forall $f_j(x_1, ..., x_{t_j}) \mid \exists x_i = f_h$ **do**
  Define $f_j^{(a)}(x_1, ..., x_{i-1}, x_{i+1,...}x_{t_j})$
  $\mathcal{F}_a = \mathcal{F}_a - FCOV(f_j) \cup FCOV(f_j^{(a)})$
  $\Delta_{\mathcal{F}}^{(\mathcal{P})} = \Delta_{\mathcal{F}}^{(\mathcal{P})} \cup \{(f_j, x_i)\}$
 end
 $\mathcal{R}_a = \mathcal{R}_g$
 $\Delta_{\mathcal{R}}^{(\mathcal{P})} = \emptyset$
 forall $R_k(x_1, ..., x_{t_k}) \mid \exists x_i = f_h$ **do**
  Define $R_k^{(a)}(x_1, ..., x_{i-1}, x_{i+1,...}x_{t_k})$
  $\mathcal{R}_a = \mathcal{R}_a - FCOV(R_k) \cup FCOV(R_k^{(a)})$
  $\Delta_{\mathcal{R}}^{(\mathcal{P})} = \Delta_{\mathcal{R}}^{(\mathcal{P})} \cup \{(R_k, x_i)\}$
  end
endif

Table E.5 Pseudo-code for the method $\mathsf{meth}\big(\mathcal{P}_g, \omega_{hrel}(R_k)\big)$

METHOD $\mathsf{meth}\big(\mathcal{P}_g, \omega_{hrel}(R_k)\big)$
$\mathcal{O}_a = \mathcal{O}_g$
$\mathcal{A}_a = \mathcal{A}_g$
$\mathcal{F}_a = \mathcal{F}_g$
$Ra = \mathcal{R}_g - \{RCOV(R_k)\}$
$\Delta_{\mathcal{R}}^{(\mathcal{P})} = RCOV(R_k)$

Hiding a function is a simple operation, *per se*, but it may have indirect effects on the set of functions and relations. In fact, if the co-domain of f_h is the set of objects, there may be in $\Gamma_{\mathcal{F}}^{(g)}$ or $\Gamma_{\mathcal{R}}^{(g)}$ some function or relation that has f_h as one of its arguments. Then, hiding f_h, these arguments disappear and new abstract functions or relations, with one less argument, are to be defined, increasing thus the degree of abstraction. Hinding a relation has no side-effects.

E.4 Complete List of Operators

As a conclusion, we report here the complete list of the domain-independent operators available so far.

Table E.6 List of currently available operators, organized according to their nature, elements acted upon, and abstraction mechanism

Operator	Type	Arguments	Effects	Comments
ω_{hobj}	Abstr	$\mathtt{o}$	Hides object $\mathtt{o}$	
ω_{htype}	Abstr	$\mathtt{t}$	Hides type $\mathtt{t}$	All objects of type $\mathtt{t}$ are hidden in every $\mathcal{P}_g$
ω_{hattr}	Abstr	(A_m, Λ_m)	Hides attribute A_m with domain Λ_m	Values of attribute A_m are hidden in all objects
ω_{hfun}	Abstr	f_h	Hides function f_h	
ω_{hrel}	Abstr	R_k	Hides relation R_k	
$\omega_{hattrval}$	Abstr	$(A_m, \Lambda_m), v_i$	Hides value $v_i \in \Lambda_m$	Value $\mathtt{v}_i$ is replaced by $\mathtt{UN}$ in all $\mathcal{P}_g$
$\omega_{hfunargval}$	Abstr	$f_h, x_j, \mathtt{o}$	Hides the value $\mathtt{o}$ from the domain of argument x_j of function f_h	Value $\mathtt{o}$ assumed by x_j is replaced by $\mathtt{UN}$ in the $FCOV(f_h)$ of all $\mathcal{P}_g$
$\omega_{hrelargval}$	Abstr	$R_k, x_j, \mathtt{o}$	Hides the value $\mathtt{o}$ from the domain of argument x_j of relation R_k	Value $\mathtt{o}$ assumed by x_j is replaced by $\mathtt{UN}$ in the $RCOV(R_k)$ of all $\mathcal{P}_g$
$\omega_{hfuncodom}$	Abstr	$f_h, CD(f_h), v$	Hides value $\mathtt{v}$ from the codomain of f_h	Value $\mathtt{v}$ assumed by f_h is replaced by $\mathtt{UN}$ in all $\mathcal{P}_g$
$\omega_{hfunarg}$	Abstr	f_h, x_j	Argument x_j of function f_h is hidden	Arity t_h of f_h is reduced by 1
$\omega_{hrelarg}$	Abstr	R_k, x_j	Argument x_j of function R_k is hidden	Arity t_k of R_k is reduced by 1
ω_{eqobj}	Abstr	$\varphi_{eq}, \mathtt{o}^{(a)}$	Builds up an equivalence class with the objects satisfying φ_{eq}	$\mathtt{o}^{(a)}$ is a generic name denoting the class
ω_{eqattr}	Abstr	φ_{eq}, $(A^{(a)}, \Lambda_{A^{(a)}})$	Builds up an equivalence class with the attributes satisfying φ_{eq}	$\mathtt{A}^{(a)}$ is a generic name denoting the class
ω_{eqfun}	Abstr	φ_{eq} $f^{(a)}$	Builds up an equivalence class with the functions satisfying φ_{eq}	$\mathtt{f}^{(a)}$ is a generic name denoting the class

(continued)

Table E.6 continued

Operator	Type	Arguments	Effects	Comments
ω_{eqrel}	Abstr	φ_{eq} $R^{(a)}$	Builds up an equivalence class with the relations satisfying φ_{eq}	$\mathtt{R}^{(a)}$ is a generic name denoting the class
ω_{eqtype}	Abstr	$\varphi_{eq}(\mathtt{t}_1, \ldots, \mathtt{t}_k)$ $\mathtt{t}^{(a)}$	Builds up an equivalence class with the types satisfying φ_{eq}	$\mathtt{t}^{(a)}$ is a generic name denoting the class
$\omega_{eqattrval}$	Abstr	$(A, \Lambda_A), \Lambda_{A,eq}, \mathtt{v}^{(a)}$	All values in $\Lambda_{A,eq}$ form an equivalence class $\mathtt{v}^{(a)}$	$\mathtt{v}^{(a)}$ is a generic name denoting the class
$\omega_{eqfunargval}$	Abstr	$f_h, x_j, \Gamma_{\mathcal{O},eq}, \mathtt{o}^{(a)}$	All values in $\Gamma_{\mathcal{O},eq}$ form an equivalence class $\mathtt{o}^{(a)}$	$\mathtt{o}^{(a)}$ is a generic name denoting the class
$\omega_{eqrelargval}$	Abstr	$R_k, x_j, \Gamma_{\mathcal{O},eq}, \mathtt{o}^{(a)}$	All values in $\Gamma_{\mathcal{O},eq}$ form an equivalence class $\mathtt{o}^{(a)}$	$\mathtt{o}^{(a)}$ is a generic name denoting the class
$\omega_{eqfuncodom}$	Abstr	$f_h, CD(f_h), V_{eq}, \mathtt{v}^{(a)}$	All values in V_{eq} form an equivalence class $\mathtt{v}^{(a)}$	$\mathtt{v}^{(a)}$ is a generic name denoting the class
$\omega_{eqfunarg}$	Abstr	$f_h, Z_{eq}, z^{(a)}$	All values in Z_{eq} form an equivalence class $z^{(a)}$	$z^{(a)}$ is a generic name denoting the class
$\omega_{eqrelarg}$	Abstr	$R_k, Z_{eq}, z^{(a)}$	All values in Z_{eq} form an equivalence class $z^{(a)}$	$z^{(a)}$ is a generic name denoting the class
$\omega_{hierattrval}$	Abstr	(A_m, Λ_m), $\Lambda_{m,child}, \mathtt{v}^{(a)}$	The values of attribute A_m belonging to $\Lambda_{m,child}$ are hidden, and a new node $\mathtt{v}^{(a)}$ is created, such that $\forall \mathtt{v}_i \in \Lambda_{m,child} : \mathtt{v}_i$ **is − a** $\mathtt{v}^{(a)}$	A node of higher level is created in a hierarchy of attribute's values
$\omega_{hiertype}$	Abstr	$\Gamma^{(g)}_{TYPE,child}, \mathtt{t}^{(a)}$	Types $\mathtt{t} \in \Gamma^{(g)}_{TYPE,child}$ are hidden, and a new type $\mathtt{t}^{(a)}$ is created, such that $\mathtt{t}$ **is − a** $\mathtt{t}^{(a)}$	In each $\mathcal{P}_g$, all objects of a type $\mathtt{t} \in \Gamma^{(g)}_{TYPE,child}$ are hidden and replaced by a corresponding object of type $\mathtt{t}^{(a)}$

(continued)

Table E.6 continued

Operator	Type	Arguments	Effects	Comments
$\omega_{hierfuncodom}$	Abstr	$(f_h, CD(f_h))$, $CD(f_h)_{child}, \mathtt{v}^{(a)}$	The values of $CD(f_h)$ belonging to $CD(f_h)_{child}$ are hidden, and a new node $\mathtt{v}^{(a)}$ is created, such that $\forall \mathtt{v}_i \in CD(f_h) \ : \ \mathtt{v}_i$ **is − a** $\mathtt{v}^{(a)}$	A node of higher level is created in a hierarchy of values in a function's codomain
$\omega_{hierattr}$	Abstr	$\Gamma^{(g)}_{\mathcal{A},child}, (A^{(a)}, \Lambda^{(a)})$	The attributes contained in $\Gamma^{(g)}_{\mathcal{A},child}$ are replaced by a new attribute, $(A^{(a)}, \Lambda^{(a)})$, such that $\forall A_i \in \Gamma^{(g)}_{\mathcal{A},child} \ : \ A_i$ **is − a** $A^{(a)}$	Values in Λ_i, for each $A_i \in \Gamma^{(g)}_{\mathcal{A},child}$ are linked to corresponding values in $\Lambda^{(a)}$
$\omega_{hierfun}$	Abstr	$\Gamma^{(g)}_{\mathcal{F},child}, f^{(a)}$	The functions contained in $\Gamma^{(g)}_{\mathcal{F},child}$ are replaced by a new function, $f^{(a)}$ such that $\forall f_i \in \Gamma^{(g)}_{\mathcal{F},child} \ : \ f_i$ **is − a** $f^{(a)}$	Values in $CD(f_i)$, for each $f_i \in \Gamma^{(g)}_{\mathcal{F},child}$ are linked to corresponding values in $CD(f^{(a)})$
$\omega_{hierrel}$	Abstr	$\Gamma^{(g)}_{\mathcal{R},child}, R^{(a)}$	The relations contained in $\Gamma^{(g)}_{\mathcal{R},child}$ are replaced by a new relation, $R^{(a)}$ such that $\forall R_i \in \Gamma^{(g)}_{\mathcal{R},child} \ : \ R_i$ **is − a** $R^{(a)}$	
ω_{coll}	Abstr	$\mathtt{t}, \mathtt{t}^{(a)}$	Makes a single, collective type $\mathtt{t}^{(a)}$, starting with a number of elements of type $\mathtt{t}$	
ω_{aggr}	Abstr	$(\mathtt{t}_1, \ldots, \mathtt{t}_s), \mathtt{t}^{(a)}$	Makes a composite type $\mathtt{t}^{(a)}$, starting with a set of objects of different types $\mathtt{t}_1, \ldots, \mathtt{t}_s$	
ω_{group}	Abstr	$\varphi_{group}, \mathtt{G}^{(a)}$	Forms a group with name $\mathtt{G}^{(a)}$ with the set of objects satisfying φ_{group}	The group can be defined extensionally. The operator acts on *objects*, not types

(continued)

Table E.6 continued

Operator	Type	Arguments	Effects	Comments
ω_{constr}	Abstr	Constr	Defines a new description element (attribute, function, relation) $Constr : \Gamma^{(g)}_{\mathcal{A},\mathcal{F},\mathcal{G}} \rightarrow \Gamma^{(a)}_{\mathcal{A},\mathcal{F},\mathcal{G}}$ $\Gamma^{(g)}_{\mathcal{A},\mathcal{F},\mathcal{R}} = \Gamma^{(g)}_{\mathcal{A}} \cup \Gamma^{(g)}_{\mathcal{F}} \cup \Gamma^{(g)}_{\mathcal{R}}$ $\Gamma^{(a)}_{\mathcal{A},\mathcal{F},\mathcal{R}} = \Gamma^{(a)}_{\mathcal{A}} \cup \Gamma^{(a)}_{\mathcal{F}} \cup \Gamma^{(a)}_{\mathcal{R}}$	Most often used to construct a new attribute
$\rho_{replattrval}$	Approx	$(A_m, \Lambda_m), \mathtt{v}_i, \mathtt{v}^{(p)}$	Value $\mathtt{v}_i$ of attribute A_m is replaced with value $\mathtt{v}^{(p)}$	$\mathtt{v}^{(p)}$ is a specific value, different from $\mathtt{v}_i$
$\rho_{replfun}$	Approx	$f_h, g_h^{(p)}$	Function f_h is replaced with function $g_h^{(p)}$	$g_h^{(p)}$ is a specific function, different from f_h
$\rho_{replrel}$	Approx	$R_k, R_k^{(p)}$	Relation R_k is replaced with relation $R_k^{(p)}$	$R_k^{(p)}$ is a specific relation, different from R_k ρ_{idobj}
ρ_{idobj}	Approx	$\varphi_{id}(\mathtt{o}_1, \ldots, \mathtt{o}_k), \mathtt{o}^{(p)}$	Set of objects satisfying φ_{id} form an equivalence class All objects are equated to $\mathtt{o}^{(p)}$	$\mathtt{o}_h^{(p)}$ is a one of the objects in the equivalence class
ρ_{idtype}	Approx	$\varphi_{id}(\mathtt{t}_1, \ldots, \mathtt{t}_k), \mathtt{t}^{(p)}$	Set of types satisfying φ_{id} form an equivalence class All objects of the equated types become of type $\mathtt{t}^{(p)}$	$\mathtt{t}_h^{(p)}$ is one of the types in the equivalence class
$\rho_{idattrval}$	Approx	$(A, \Lambda_A), \Lambda_{A,id}, \mathtt{v}^{(p)}$	Values in $\Lambda_{A,id}$ become equal to $\mathtt{v}^{(p)}$	$\mathtt{v}^{(p)}$ is a one of the values in $\Lambda_{A,id}$
$\rho_{idfuncodom}$	Approx	$f_h, CD(f_h), V_{id}, \mathtt{v}^{(p)}$	Values in V_{id} become equal to $\mathtt{v}^{(p)}$	$\mathtt{v}^{(p)}$ is a one of the values in V_{id}
$\rho_{idfunarg}$	Approx	$f_h, Z_{id}, z^{(a)}$	All arguments in Z_{id} are equated to $z^{(a)}$	$z^{(a)}$ is an element of Z_{id}
$\rho_{idrelarg}$	Approx	$R_k, Z_{id}, z^{(a)}$	All arguments in Z_{id} are equated to $z^{(a)}$	$z^{(a)}$ is an element of Z_{id}

Type "Abstr" stands for abstraction operator, whereas "Approx" stands for approximation operator

Appendix F
Abstraction Patterns

In this appendix two more abstraction patterns are described, for the sake of illustration. The complete set, corresponding to the full set of operators listed in Appendix E, can be found in the book's companion Web site.

In Table F.1 the pattern referring to *hiding an argument* of a function or relation is reported.

Table F.1 HIDING-ARGUMENT—Abstraction Pattern that hides an argument of a function or relation

NAME	HIDING-ARGUMENT
ALSO KNOWN	Described by Plaisted [419] as "propositionalization". As it requires a structured representation, it is less popular than hiding an element. In Machine Learning it may correspond to the task of "propositionalization".
GOAL	In Problem Solving and Automated Reasoning it is meant to speed up inference by providing a sketch of a proof without variables.
TYPICAL APPLICATIONS and KNOWN USE	Very much used in databases, where it corresponds to the *projection* operation in relational algebra.
IMPLEMENTATION ISSUES	Problems with this operator may arise when the unique argument of a univariate function has to be hidden. In this case the function becomes a constant. A relation, whose arguments are all hidden, becomes a Boolean variable with an empty cover.
KNOWN USES	Machine Learning, CSP, Problem Solving, Theorem Proving.
SIMILAR PATTERNS	This pattern is related to the Equating Arguments Pattern, and to Building a hierarchy of arguments.

L. Saitta and J.-D. Zucker, *Abstraction in Artificial Intelligence and Complex Systems*,
DOI: 10.1007/978-1-4614-7052-6,

In Table F.2 we provide the template for aggregating objects.

Table F.2 AGGREGATION—Aggregation Pattern that forms new objects starting from existing ones

NAME	AGGREGATION
ALSO KNOWN	In Machine Learning the *aggregation* operator is known as "predicate invention", "predicate construction", or "term construction", whereas in Data Mining it is related to "motif discovery". In general, it is the basis of the "constructive induction" approach to learning. In Planning, Problem Solving, and Reinforcement Learning it includes "state aggregation" and "spatial and/or temporal aggregation".
GOAL	It aims at working, in any field, with "high level" constructs in the description of data and in theories, in order to reduce the computational cost and increasing the meaningfulness of the results.
TYPICAL APPLICATIONS	Finding regions and objects in the visual input, representing physical apparata at various levels of details by introducing composite components.
IMPLEMENTATION ISSUES	Implementing the grouping operator may require even complex algorithms, and the cost of aggregation has to be weighted against the advantages in the use of the abstract representation.
KNOWN USES	Even though not always under the name of abstraction, aggregation and feature construction is very much used in computer vision, description of physical systems, Machine Learning, Data Mining, and Artificial Intelligence in general.

Appendix G
Abstraction of Michalski's "Train" Problem

In this appendix the detailed application of the introduce operators in Michalski's "train" problem are reported. The results of the method are described in Chap. 9. In Table G.1 the method $\mathsf{meth}\big(\mathcal{P}_g, \omega_{aggr}(\{\texttt{car, load}\}, \texttt{loadedcar})\big)$ is reported.

The parameters, which are listed in Table G.2, specify how objects are actually aggregated and how attributes and relations change as a consequence.

Finally, Table G.3 describes the actual algorithm performing the aggregation abstraction.

Table G.1 Method $\mathsf{meth}\big(\mathcal{P}_g, \omega_{aggr}(\{\texttt{car, load}\}, \texttt{loadedcar})\big)$

NAME	$\mathsf{meth}\big(\mathcal{P}_g, \omega_{aggr}(\{\texttt{car, load}\}, \texttt{loadedcar})\big)$
INPUT	$\mathcal{P}_g, \{\texttt{car, load}\}, \texttt{loadedcar}$, $g : \mathcal{O}_{\texttt{car}} \times \mathcal{O}^n_{\texttt{load}} \rightarrow \mathcal{O}_{\texttt{loadedcar}}$ $(n \geqslant 0)$ $g(y, x_1, \ldots, x_n) =$ **if** $[y \in \mathcal{O}_{\texttt{car}}] \wedge [x_1, \ldots, x_n \in \mathcal{O}_{\texttt{load}}] \wedge$ $[(x_i, y) \in RCOV(R_{Inside})\ (1 \leqslant i \leqslant n)] \wedge$ $[y, x_1, \ldots, x_n]$ are labelled with the same example] **then** z
OUTPUT	$\mathcal{P}_a, R_{Partof} \subseteq (\mathcal{O}_{\texttt{load}} \cup \mathcal{O}_{\texttt{car}}) \times \mathcal{O}_{\texttt{loadedcar}}$
APPL-CONDITIONS	$\exists \texttt{c} \in \mathcal{O}_{\texttt{car}}$ $\exists$ (*different*) $\ell_1, \ldots, \ell_n \in \mathcal{O}_{\texttt{load}}$ $\texttt{c}, \ell_1, \ldots, \ell_n$ are labelled with the same example $(\ell_i, \texttt{c}) \in RCOV(R_{Inside})\ (1 \leqslant i \leqslant n)$
PARAMETERS	See Table G.2
MEMORY	$\Delta^{(\mathcal{P})}, RCOV(R_{Partof})$
BODY	See Table G.3

DOI: 10.1007/978-1-4614-7052-6,

Table G.2 Parameters of the method meth $\left(\mathcal{P}_g, \omega_{aggr}(\{\texttt{car, load}\}, \texttt{loadedcar})\right)$

$\alpha(x, y) \Rightarrow$	$LCshape^{(a)}(z) = Cshape(y)$
	$LClength^{(a)}(z) = Clength(y)$
	$LCwall^{(a)}(z) = Cwall(y)$
	$LCwheels^{(a)}(z) = Cwheels(y)$
$\gamma(x_1, x_2) \Rightarrow$	R_{Inside} is NA
	if $\exists\, y'$ **s.t.** $(y', y) \in RCOV(R_{Infrontof})$ **then** $(y', z) \in RCOV(R^{(a)}_{Infrontof})$
	if $\exists\, y'$ **s.t.** $[(y, y') \in RCOV(R_{Infrontof})$ **then** $(z, y') \in RCOV(R^{(a)}_{Infrontof})$

Table G.3 Pseudo-code for the method meth $\left(\mathcal{P}_g, \omega_{aggr}(\{\texttt{car, load}\}, \texttt{loadedcar})\right)$

METHOD meth $\left(\mathcal{P}_g, \omega_{aggr}(\{\texttt{car, load}\}, \texttt{loadedcar})\right)$
Let $R_{partof} \subseteq (\mathcal{O}_{\texttt{load}} \cup \mathcal{O}_{\texttt{car}}) \times \mathcal{O}_{\texttt{loadedcar}}$ *be a new predicate*
Let $\sigma = \{\ell_1, ..., \ell_n \mid \ell_i \in \mathcal{O}_{\texttt{load}}, (\ell_i, \texttt{c}) \in RCOV(R_{Inside})\ (1 \leqslant i \leqslant n)\}$
$\mathcal{O}_a = \mathcal{O}_g, \mathcal{A}_a = \mathcal{A}_g, \mathcal{R}_a = \mathcal{R}_g$
$\Delta^{(\mathcal{P})}_{\mathcal{O}} = \Delta^{(\mathcal{P})}_{\mathcal{A}} = \Delta^{(\mathcal{P})}_{\mathcal{R}} = \emptyset$
$RCOV(R_{partof}) = \emptyset$
Build up $\texttt{d} = g(\texttt{c}, \ell_1, \ldots, \ell_n)$
$RCOV(R_{partof}) = RCOV(R_{partof}) \cup \{(\texttt{c}, \texttt{d})\}$
for $i = 1, n$ **do**
  $RCOV(R_{partof}) = RCOV(R_{partof}) \cup \{(\ell_i, \texttt{d})\}$
end
$\mathcal{O}_a = \mathcal{O}_a - \{\texttt{c}, \ell_1 2, \ldots, \ell_n\} \cup \{\texttt{d}\}$
$\Delta^{(\mathcal{P})}_{\mathcal{O}} = \{\texttt{c}, \ell_2, \ldots, \ell_n\}$
$\mathcal{A}_a = \mathcal{A}_a - \{(\texttt{c}, \texttt{car}, Cshape(\texttt{c}), Clength(\texttt{c}), Cwall(\texttt{c}), Cwheels(\texttt{c}))\} -$
$\{(\ell_i, \texttt{load}, Lshape(\ell_i) | (1 \leqslant i \leqslant n)\}$
$\mathcal{A}_a = \mathcal{A}_a \cup \{(\texttt{d}, \texttt{loadedcar}, Cshape(\texttt{c}), Clength(\texttt{c}), Cwall(\texttt{c}), Cwheels(\texttt{c}))\}$,
$\Delta^{(\mathcal{P})}_{\mathcal{A}} = \Delta^{(\mathcal{P})}_{\mathcal{A}} \cup \{(\texttt{c}, \texttt{car}, Cshape(\texttt{c}), Clength(\texttt{c}), Cwall(\texttt{c}), Cwheels(\texttt{c}))\} \cup$
$\{(\ell_i, \texttt{load}, Lshape(\ell_i) | (1 \leqslant i \leqslant n)\}$
forall$(y', \texttt{c}) \in RCOV(R_{Infrontof})$ **do**
  $RCOV(R^{(a)}_{Infrontof}) = RCOV(R_{Infrontof}) - \{(y', \texttt{c})\} \cup \{(y', \texttt{d})\}$
  $\Delta^{(\mathcal{P})}_{\mathcal{R}} = \Delta^{(\mathcal{P})}_{\mathcal{R}} \cup \{(y', \texttt{c})\}$
end
forall$(\texttt{c}, y') \in RCOV(R_{Infrontof})$ **do**
  $RCOV(R^{(a)}_{Infrontof}) = RCOV(R_{Infrontof}) - \{(\texttt{c}, y')\} \cup \{(\texttt{d}, y')\}$
  $\Delta^{(\mathcal{P})}_{\mathcal{R}} = \Delta^{(\mathcal{P})}_{\mathcal{R}} \cup \{(\texttt{c}, y')\}$
end
$\Delta^{(\mathcal{P})} = \Delta^{(\mathcal{P})}_{\mathcal{O}} \cup \Delta^{(\mathcal{P})}_{\mathcal{A}} \cup \Delta^{(\mathcal{P})}_{\mathcal{R}} \cup RCOV(R_{partof})$

Appendix H
Color Figures

In this appendix, some of the figures appearing in the book are reported with their original colors.

Fig. H.1 Vasilij Kandinsky, Composition VII, 1913. The Tretyakov Gallery, Moscow

DOI: 10.1007/978-1-4614-7052-6,

Fig. H.2 *Nocturne in Black and Gold* by J. McNeill Whistler (1875). It is considered as a first step toward abstraction in painting

Fig. H.3 K. Malevich's *Portrait of Ivan Klioune* (1911). The State Russian Museum, St. Petersburg

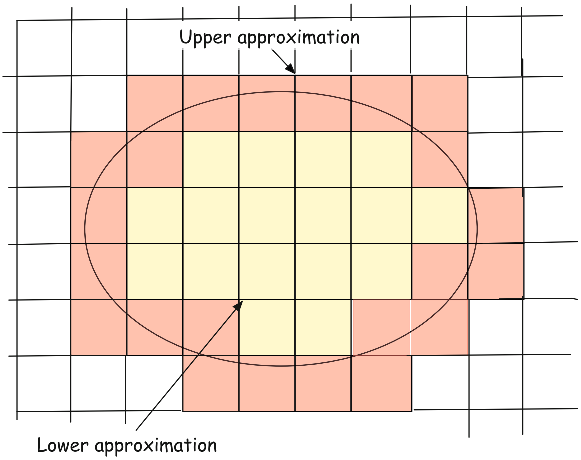

Fig. H.4 Upper (*pink* + *yellow* regions) and lower (*yellow* region) approximations of a concept $\mathcal{X}$ = `Oval`, defined as a region in the 2D plane

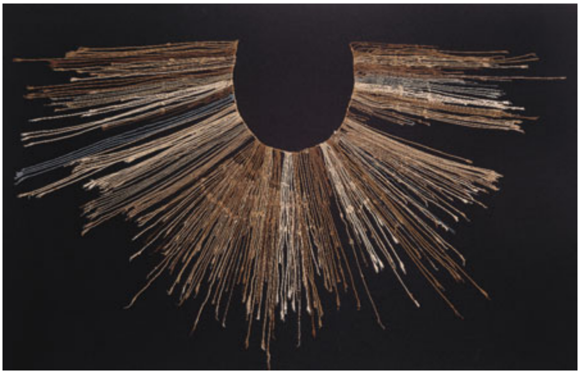

Fig. H.5 Incas used *quipus* to memorize numbers. A quipu is a cord with nodes that assume position-dependent values. An example of the complexity a quipu may reach. (Reprinted with permission from *Museo Larco*, Pueblo Libre, Lima, Peru.)

(a) (b)

Fig. H.6 **a** Picture of a poppy field. If we only have this picture, it is impossible to say whether it is concrete or abstract. **b** The same picture in black and white. By comparison, this last is less informative than the colored one, because the information referring to the color has been removed; then picture **b** is more abstract than picture **a**

Fig. H.7 A color picture has been transformed into a *black* and *white* one. If the color is added again, there is no clue for performing this addition correctly, if it is not know how the color was originally removed

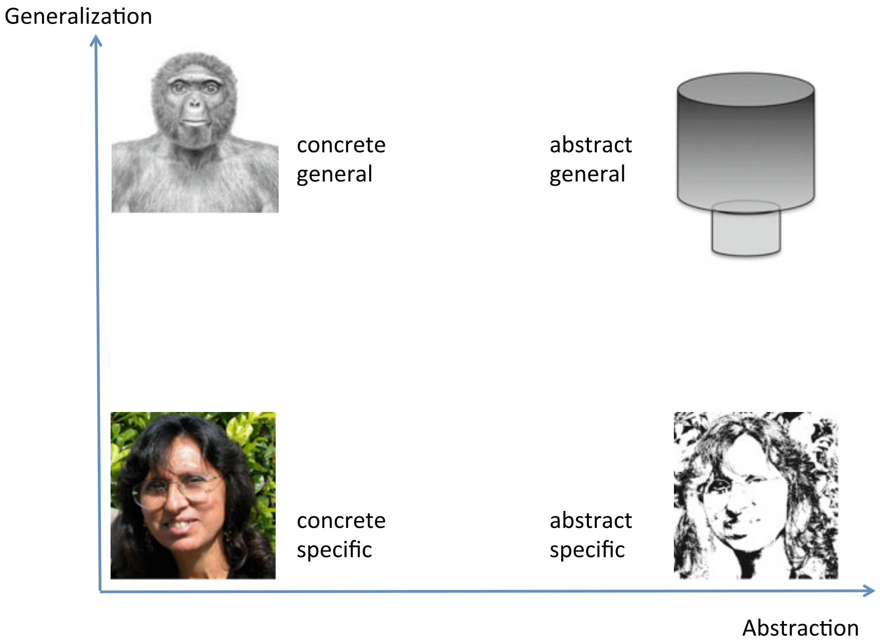

Fig. H.8 Abstraction and generalization can be combined in every possible way. In the *left-bottom* corner there is picture of one of the authors, which is specific (only one instance) and concrete (all the skin, hair, face, ... details are visible). In the *right-bottom* corner there is a version of the picture which is specific (only one instance, as the person is still recognizable) and abstract (most details of the appearance are hidden). In the *top-left* corner the chimpanzee–human last common ancestor is represented with many physical details, making thus the picture still concrete; however many monkeys and hominides satisfy the same description, so that this is an example of a concrete but general concept. Finally, in the *top-right* corner there is a representation of a human head according to Marr [353] (see Fig. 2.13); the head is abstract (very few details of the appearance) and general (any person could be an instance)

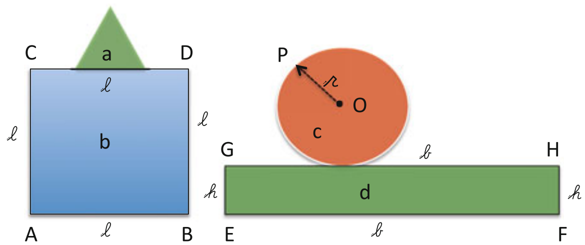

Fig. H.9 A geometrical scenario with various geometrical elements

Fig. H.10 Example of method $\mathsf{meth}(\mathcal{P}_g, \omega_{hattr}\ ((A_m, \Lambda_m))$. The attribute $A_m = Color$ is hidden from the *left* picture giving a *gray*-level picture (*right*). Each pixel shows a value of the light intensity, but this last is no more distributed over the R,G,B channels

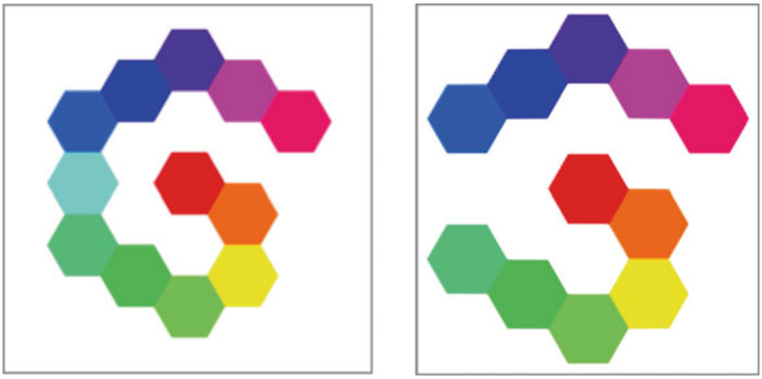

Fig. H.11 Example of application of the method $\mathsf{meth}[\mathcal{P}_g, \omega_{hattrval}((Color, \Lambda_{Color}), \texttt{turquoise})]$. The value `turquoise` is *hidden* from the left picture; a less colorful picture is obtained (*right*), where objects of color `turquoise` become transparent (`UN`)

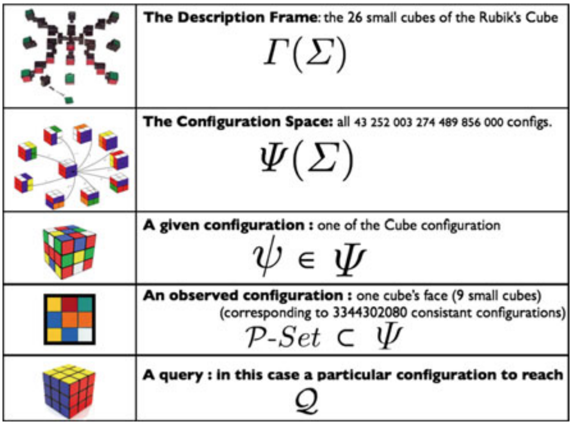

Fig. H.12 The Rubik's cube can be described in terms of the 26 small component cubes, which give rise to the description frame Γ. Each arrangement of the cubes generates a specific configuration ψ; the configuration set, Ψ, is very large. A configuration is a complete description of the positions of the small cubes, so that it is unique. If Rubik's cube is observed only partially, for instance by looking only at one face, the observation corresponds to many configurations, each one obtained by completing the invisible faces of the cube in a different way; in this case we have a $\mathcal{P}$-*Set* $\mathcal{P}$, which is a set of configurations. The query $\mathcal{Q}$ can be represented by a particular configuration to be reached starting from an initial one

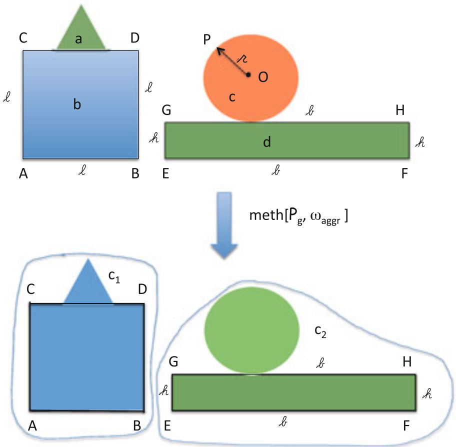

Fig. H.13 Application of method $\text{meth}\big[\mathcal{P}_g, \omega_{aggr}((\texttt{figure}, \texttt{figure}), \texttt{tower})\big]$. Objects a and b are aggregated to obtain object c_1, and objects c and d are aggregated to obtain object c_2. The color of c_1 is *blue*, because b is larger than a, whereas the color of c_2 is *green*. Both composite objects are large. The new object c_1 is at the *left* of c_2

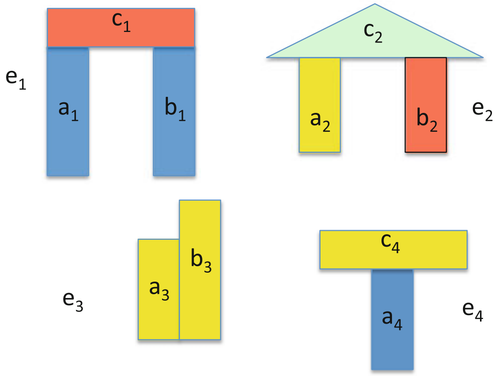

Fig. H.14 Examples of four structured objects, used to learn the concept of an "arch". Each component has a shape (rectangle or triangle) and a color (*blue*, *red*, *yellow*, or *green*). They are linked by two relations, namely R_{ontop} and $R_{adjacent}$

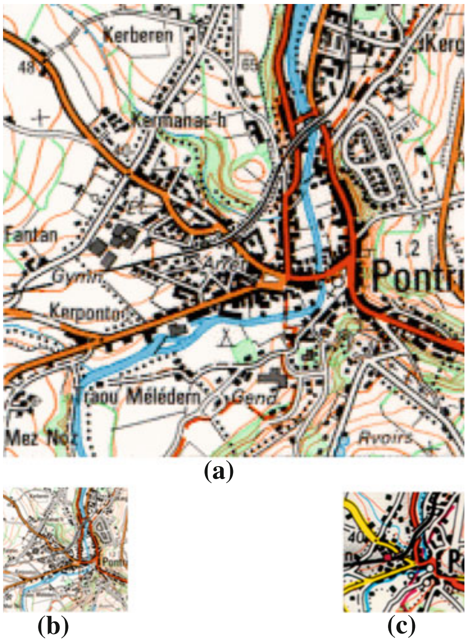

Fig. H.15 **a** Part of a map at 1/25000 scale. **b** A 16-fold reduction of the map. **c** Cartographic generalization of the map at the 1/100 000 scale. By comparing **b** and **c** the differences between simply reducing and generalizing are clearly apparent

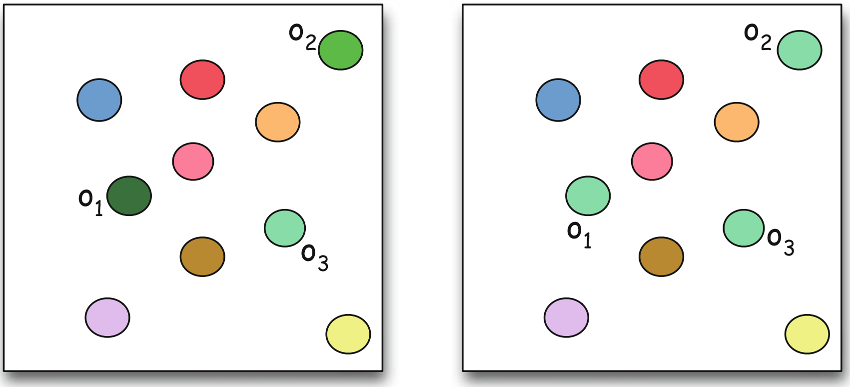

Fig. H.16 Application of method $\mathsf{meth}(\mathcal{P}_g, \omega_{eqattrval}\left((Color, \Lambda_{Color}), V_{id}, v^{(a)}\right)$ to the figure on the *left*. Let V_{id} = {`olive-green`, `sea-green`, `lawn-green`, `light-green`, `dark-green`}. Objects $\mathrm{o}_1, \mathrm{o}_2$, and o_3 have color `dark-green`, `lawn-green`, and `sea-green`, respectively. After equating all shades of *green* to `sea-green`, the color of all three considered objects becomes `sea-green`

References

1. A. Aamodt, E. Plaza, Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Comm. **7**, 39–59 (1994)
2. R. Abbott et al., Bits don't have error bars: Upward conceptualization and downward approximation, in *Philosophy and Engineering*, ed. by I. van de Poel, D. Goldberg (Springer, Berlin, 2010), pp. 285–294
3. S. Abdallah, M. Plumbley, Predictive information, Multi-information, and Binding Information. Technical Report C4DM-TR10-10, Centre for Digital Music, Queen Mary University of London, (2010).
4. D. Achlioptas, L. Kirousis, E. Kranakis, D. Krinzac, M. Molloy, Y. Stamatiou, Random constraint satisfaction: a more accurate picture. Lecture notes in Computer Science **1330**, 107–120 (1997)
5. D. Aha, Incremental constructive induction: an instance-based approach. in Proceedings of the 8th International Workshop on Machine Learning, (Evanston, USA, 1991), pp. 117–121.
6. H. Ajroud, A. Jaoua, Abstraction of objects by conceptual clustering. Inf. Sci. **109**, 79–94 (1998)
7. C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, S. Angel, *A Pattern Language-Towns, Buildings, Construction* (Oxford University Press, New York, 1977)
8. J.F. Allen, Maintaining knowledge about temporal intervals. Commun. ACM **26**, 832–843 (1983)
9. E. Alphonse, C. Rouveirol, Selective propositionalization for relational learning. Lect. Notes Comput. Sci. **1704**, 271–276 (1999)
10. E. Alphonse, C. Rouveirol, Lazy propositionalisation for relational learning. in Proceedings of the 14th European Conference on Artificial Intelligence, (Berlin, Germany, 2000), pp. 256–260.
11. D. Alvaro, P. Pazo-Alvarez, A. Capilla, E. Amenedo, Oscillatory brain activity in the time frequency domain associated to change blindness and change detection awareness. J. Cogn. Neurosci. **24**, 337–350 (2012)
12. E. Amaldi, V. Kann, On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. Theor. Comput. Sci. **209**, 237–260 (1998)
13. K. Amaratunga, R. Sudarshan, Multiresolution modeling with operator-customized wavelets derived from finite elements. Comput. Methods Appl. Mech. Eng. **195**, 2509–2532 (2006)
14. S. Amarel, On representations of problems of reasoning about actions. Mach. Intell. **3**, 131–171 (1968)

L. Saitta and J.-D. Zucker, *Abstraction in Artificial Intelligence and Complex Systems*,
DOI: 10.1007/978-1-4614-7052-6,

15. A. Amir, M. Lindenbaum, Grouping-based hypothesis verification in object recognition. in Proceedings of the 13th Israeli Conference on Artificial Intelligence and Computer Vision, (Tel-Aviv, Israel, 1998).
16. A. Amir, M. Lindenbaum, Quantitative analysis of grouping processes. IEEE Trans. PAMI **20**, 168–185 (1998)
17. D. Andre, Learning hierarchical behaviors. in Proceedings of the NIPS Workshop on Abstraction and Hierarchy in Reinforcement Learning, (Amherst, USA, 1998).
18. D. Andre, S.J. Russell, State abstraction for programmable Reinforcement Learning agents. in Proceedings of the 18th National Conference on Artificial Intelligence, (Menlo Park, USA, 2002), pp. 119–125.
19. G. Antonelli, The nature and purpose of numbers. J. Philos. **107**, 191–212 (2010)
20. G. Antonelli, Numerical abstraction via the Frege quantifier. Notre Dame J. Formal Logic **51**, 161–179 (2010)
21. N. Archer, M. Head, Y. Yuan, Patterns in information search for decision making: The effects of information abstraction. Int. J. Hum. Comput. Stud. **45**, 599–616 (1996)
22. A. Arenas, J. Duch, A. Fernández, S. Gómez, Size reduction of complex networks preserving modularity. New J. Phys. **9**, 176 (2007)
23. A. Arenas, A. Fernández, S. Gómez, Analysis of the structure of complex networks at different resolution levels. New J. Phys. **10**, 053039 (2008)
24. Aristotle, Introduction and commentary. Prior and Posterior Analytics, ed. by W.D. Ross, (Oxford, UK, 1949).
25. M. Asadi, M. Huber, Action dependent state space abstraction for hierarchical learning systems. in Artificial Intelligence and Applications, ed. by M.H. Hamza, (IASTED/ACTA Press, 2005), pp. 421–426.
26. N. Ay, M. Müller, A. Szkola, Effective complexity of stationary process realizations. Entropy **13**, 1200–1211 (2011)
27. F. Bacchus, Q. Yang, Downward refinement and the efficiency of hierarchical problem solving. Artif. Intell. **71**, 43–100 (1994)
28. C. Bäckström, Planning with abstraction hierarchies can be exponentially less efficient. in Proceedings of the 14th Int Joint Conference on Artificial Intelligence, (Montreal, Canada, 1995), pp. 1599–1604.
29. A. Baker, Simplicity. in The Stanford Encyclopedia of Philosophy, ed. by E.N. Zalta (2010).
30. T. Ball, B. Cook, S. Lahiri, L. Zhang, Zapato: automatic theorem proving for predicate abstraction refinement. in Proceedings of the 16th Int. Conference on Computer-Aided Verification, (Boston, USA, 2004), pp. 457–461.
31. D. Ballard, C. Brown, *Computer Vision* (Prentice Hall, New Jersey, 1982)
32. S. Bang, A hub-protein based visualization of large protein-protein interaction networks. in Proceedings of the IEEE Conference of the Engineering in Medicine and Biology Society, (Lyon, France, 2007), pp. 1217–1220.
33. J. Barron, J. Malik, Discovering efficiency in coarse-to-fine texture classification. Technical Report UCB/EECS-2010-94, EECS Department, University of California, Berkeley (2010).
34. L. Barsalou, On the indistinguishability of exemplar memory and abstraction in category representation. in ed. by T. Srull, R. Wyer Jr. Adavances in Social Cognition, Vol. III: Content and Process Specificity in the Effects of Prior Experience, (Lawrence Erlbaum, Hillsdale, 1990), pp. 61–88.
35. L. Barsalou, Abstraction as dynamic interpretation in perceptual symbol systems. in ed. by L. Gershkoff-Stowe, D. Rakison Building Object Categories, (Erlbaum, NJ, 2005), pp. 389–431.
36. L. Barsalou, Simulation, situated conceptualization, and prediction. Phil. Trans. Roy. Soc. B **364**, 1281–1289 (2009)
37. L. Barsalou, K. Wiemer-Hastings, Situating abstract concepts, in *Grounding Cognition: The Role of Perception and Action in Memory, Language, and Thought*, ed. by D. Pecher, R. Zwaan (Cambridge University Press, New York, 2005), pp. 129–163
38. L. Barsalou, C. Wilson, W. Hasenkamp, On the vices of nominalization and the virtues of contextualizing, in *The Mind in Context*, ed. by E. Smith (Guilford Press, NY, 2010), pp. 334–360

39. A.G. Barto, S. Mahadevan, Recent advances in hierarchical Reinforcement Learning. Discrete Event Dyn. Syst. **13**, 41–77 (2003)
40. J. Barwise, J. Seligman, *Information Flow: The Logic of Distributed Systems* (Cambridge University Press, New York, 1997)
41. M. Basseville, A. Benveniste, K.C. Chou, S.A. Golden, R. Nikoukhah, A.S. Willsky, Modeling and estimation of multiresolution stochastic processes. IEEE Trans. Inf. Theor. **38**, 766–784 (1992)
42. M. Bassok, K. Dunbar, K. Holyoak, Introduction to the special section on the neural substrate of analogical reasoning and metaphor comprehension. J. Exp. Psychol. Learn. Mem. Cogn. **38**, 261–263 (2012)
43. J. Bauer, I. Boneva, M. Kurbán, A. Rensink, A modal-logic based graph abstraction. Lect. Notes Comput. Sci. **5214**, 321–335 (2008)
44. K. Bayer, M. Michalowski, B. Choueiry, C. Knoblock, Reformulating constraint satisfaction problems to improve scalability. in Proceedings of the 7th International Symposium on Abstraction, Reformulation, and Approximation, (Whistler, Canada, 2007), pp. 64–79.
45. A. Belussi, C. Combi, G. Pozzani, Towards a formal framework for spatio-temporal granularities. in Proceedings of the 15th International Symposium on Temporal Representation and Reasoning, (Montreal, Canada, 2008), pp. 49–53.
46. P. Benjamin, M. Erraguntla, D. Delen, R. Mayer, Simulation modeling and multiple levels of abstraction. in Proceedings of the Winter Simulation Conference, (Piscataway, New Jersey, 1998), pp. 391–398.
47. J. Benner, *The Ancient Hebrew Lexicon of the Bible* (Virtualbookworm, College Station, 2005)
48. C. Bennett, Dissipation, information, computational complexity and the definition of organization, in *Emerging Syntheses in Science*, ed. by D. Pines (Redwood City, USA, Addison-Wesley, 1987), pp. 215–234
49. C. Bennett, Logical depth and physical complexity. in ed. by R. Herken The Universal Turing Machine: A Half-Century Survey, (Oxford University Press, Oxford, 2011), pp. 227–257.
50. A. Berengolts, M. Lindenbaum, On the performance of connected components grouping. Int. J. Comput. Vis. **41**, 195–216 (2001)
51. A. Berengolts, M. Lindenbaum, On the distribution of saliency. in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, (Washington, USA, 2004), pp. 543–549.
52. F. Bergadano, A. Giordana, L. Saitta, *Machine Learning: An Integrated Framework and its Application* (Ellis Horwood, Chichester, 1991)
53. G. Berkeley, Of the Principles of Human Knowledge. (Aaron Rahmes for Jeremy Pepyat, Skynner Row, 1710)
54. S. Bertz, W. Herndon, The similarity of graphs and molecules, in *Artificial Intelligence Applications to Chemistry*, ed. by T. Pierce, B. Hohne (ACS, USA, 1986), pp. 169–175
55. T. Besold, Computational Models of Analogy-Making: An Overview Analysis of Computational Approaches to Analogical Reasoning, Ph.D. thesis (University of Amsterdam, NL, 2011).
56. C. Bessiere, P.V. Hentenryck, To be or not to be ... a global constraint. in Proceedings of the 9th International Conference on Principles and Practices of Constraint Programming, (Kinsale, Ireland, 2003).
57. W. Bialek, I. Nemenman, N. Tishby, Predictability, complexity, and learning. Neural Comput. **13**, 2409–2463 (2001)
58. M. Biba, S. Ferilli, T. Basile, N.D. Mauro, F. Esposito, Induction of abstraction operators using unsupervised discretization of continuous attributes. in Proceedings of The International Conference on Inductive Logic Programming, (Santiago de Compostela, Spain, 2006), pp. 22–24.
59. I. Biederman, Recognition-by-components: a theory of human image understanding. Psychol. Rev. **94**, 115–147 (1987)
60. A. Bifet, G. Holmes, R. Kirkby, B. Pfahringer, Moa: Massive online analysis. J. Mach. Learn. Res. **99**, 1601–1604 (2010)

61. P. Binder, J. Plazas, Multiscale analysis of complex systems. Phys. Rev. E 63, 065203(R) (2001).
62. J. Bishop, Data Abstraction in Programming Languages, (Addison-Wesley, Reading, 1986).
63. S. Bistarelli, P. Codognet, F. Rossi, Abstracting soft constraints: Framework, properties, examples. Artif. Intell. **139**, 175–211 (2002)
64. A. Blum, P. Langley, Selection of relevant features and examples in machine learning. Artif. Intell. **97**, 245–271 (1997)
65. S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D. Hwang, Complex networks: structure and dynamics. Phys. Rep. **424**, 175–308 (2006)
66. D. Bonchev, G. Buck, Quantitative measures of network complexity, in *Complexity in Chemistry, Biology, and Ecology*, ed. by D. Bonchev, D. Rouvray (Springer, USA, 2005), pp. 191–235
67. I. Boneva, A. Rensink, M. Durban, J. Bauer, *Graph abstraction and abstract graph transformation (Centre for Telematics and Information Technology* (University of Twente Enschede, Technical report , 2007)
68. G. Booch, Object-Oriented Analysis and Design with Applications, (Addison-Wesley, Reading, 2007).
69. M. Botta, A. Giordana, Smart+: A multi-strategy learning tool. in Proceedings of the 13th International Joint Conference on Artificial Intelligence, (Chambéry, France, 1993), pp. 203–207.
70. M. Botta, A. Giordana, L. Saitta, M. Sebag, Relational learning as search in a critical region. J. Mach. Learn. Res. **4**, 431–463 (2003)
71. P. Bottoni, L. Cinque, S. Levialdi, P. Musso, Matching the resolution level to salient image features. Pattern Recogn. **31**, 89–104 (1998)
72. I. Bournaud, M. Courtine, J.-D. Zucker, Propositionalization for clustering symbolic relational descriptions. in Proceedings of the 12th International Conference on Inductive Logic Programming, (Szeged, Hungary, 2003), pp. 1–16.
73. E. Bourrel, V. Henn, Mixing micro and macro representations of traffic flow: a first theoretical step. in Proceedings of the 9th Meeting of the Euro Working Group on, Transportation, pp. 10–13 (2002).
74. O. Bousquet, Apprentissage et simplicité. Diploma thesis, (Université de Paris Sud, Paris, France, 1999), In French.
75. C. Boutilier, R. Dearden, M. Goldszmidt, Exploiting structure in policy construction. in Proceedings of the 14th International Joint Conference on Artificial Intelligence, (Montréal, Canada, 1995), pp. 1104–1111.
76. C. Boutilier, R. Dearden, M. Goldszmidt, Stochastic dynamic programming with factored representations. Artif. Intell. **121**, 49–107 (2000)
77. J. Boyan, A. Moore, Generalization in reinforcement learning: safely approximating the value function. Adv. Neural Inf. Process. Syst. **7**, 369–376 (1995)
78. K. Brassel, R. Weibel, A review and conceptual framework of automated map generalization. Int. J. Geogr. Inf. Syst. **2**, 229–244 (1988)
79. N. Bredèche, Y. Chevaleyre, J.-D. Zucker, A. Drogoul, G. Sabah, A meta-learning approach to ground symbols from visual percepts. Robot. Auton. Syst. **43**, 149–162 (2003)
80. H. Brighton, C. Mellish, Advances in instance selection for instance-based learning algorithms. Data Min. Knowl. Discov. **6**, 153–172 (2002)
81. A. Brook, Approaches to abstraction: a commentary. Int. J. Educ. Res. **27**, 77–88 (1997)
82. R. Brooks, Elephants don't play chess. Robot. Auton. Syst. **6**, 3–15 (1990)
83. R. Brooks, Intelligence without representation. Artif. Intell. **47**, 139–159 (1991)
84. V. Bulitko, N. Sturtevant, J. Lu, T. Yau, Graph abstraction in real-time heuristic search. J. Artif. Intell. Res. **30**, 51–100 (2007)
85. N. Busch, I. Fruend, C. Herrmann, Electrophysiological evidence for different types of change detection and change blindness. J. Cogn. Neurosci. **22**, 1852–1869 (2010)
86. T.D.V. Cajetanus, *De Nominum Analogia (1498)* (Zammit, Rome (Italy, 1934)

87. T. Calders, R. Ng, J. Wijsen, Searching for dependencies at multiple abstraction levels. ACM Trans. Database Syst. **27**, 229–260 (2002)
88. G. Cantor, *Contributions to the Founding of the Theory of Transfinite Numbers* (Dover Publications, UK, 1915)
89. R. Cavendish, *The Black Arts* (Perigee Books, USA, 1967)
90. G. Chaitin, On the length of programs for computing finite binary sequences: statistical considerations. J. ACM **16**, 145–159 (1969)
91. D. Chalmers, R. French, D. Hofstadter, High-level perception, representation, and analogy: a critique of Artificial Intelligence methodology. J. Exp. Theor. Artif. Intell. **4**, 185–211 (1992)
92. V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey. ACM Comput. Survey **41**, 1–58 (2009)
93. J. Charnley, S. Colton, I. Miguel, Automated reformulation of constraint satisfaction problems. in Proceedings of the Automated Reasoning Workshop, (Bristol, UK, 2006), pp. 128–135.
94. A. Chella, M. Frixione, S. Gaglio, A cognitive architecture for artificial vision. Artif. Intell. **89**, 73–111 (1997)
95. A. Chella, M. Frixione, S. Gaglio, Understanding dynamic scenes. Artif. Intell. **123**, 89–132 (2000)
96. A. Chella, M. Frixione, S. Gaglio, Conceptual spaces for computer vision representations. Artif. Intell. Rev. **16**, 87–118 (2001)
97. C. Cheng, Y. Hu, Extracting the abstraction pyramid from complex networks. BMC Bioinform. **11**, 411 (2010)
98. Y. Chevaleyre, F. Koriche, J.-D. Zucker, Learning linear classifiers with ternary weights from metagenomic data. in Proceedings of the Conférence Francophone sur l'Apprentissage Automatique, (Nancy, France, 2012), In French.
99. L. Chittaro, R. Ranon, Hierarchical model-based diagnosis based on structural abstraction. Artif. Intell. **155**, 147–182 (2004)
100. T. Chothia, D. Duggan, Abstractions for fault-tolerant global computing. Theor. Comput. Sci. **322**, 567–613 (2004)
101. B. Choueiry, B. Faltings, R. Weigel, Abstraction by interchangeability in resource allocation. in Proceedings of the 14th International Joint Conference on Artificial Intelligence, (Montreal, Canada, 1995), pp. 1694–1701.
102. B. Choueiry, Y. Iwasaki, S. McIlraith, Towards a practical theory of reformulation for reasoning about physical systems. Artif. Intell. **162**, 145–204 (2005)
103. B. Choueiry, A. Davis, Dynamic bundling: Less effort for more solutions. in Proceedings of the 5th International Symposium on Abstraction, Reformulation and Approximation, (Kananaskis, Alberta, Canada, 2002), pp. 64–82.
104. B. Choueiry, G. Noubir, On the computation of local interchangeability in discrete constraint Satisfaction problems. in Proceedings of the 15th National Conference on Artificial Intelligence, (Madison, USA, 1998), pp. 326–333.
105. J. Christensen, A hierarchical planner that creates its own hierarchies. in Proceedings of the 8th National Conference on Artificial Intelligence, (Boston, USA, 1990), pp. 1004–1009.
106. R. Cilibrasi, P. Vitànyi, Clustering by compression. IEEE Trans. Inform. Theor. **51**, 1523–1545 (2005)
107. A. Cimatti, F. Giunchiglia, M. Roveri, Abstraction in planning via model checking. in Proceedings of the 8th International Symposium on Abstraction, Reformulation, and Approximation, (Asilomar, USA, 1998), pp. 37–41.
108. E. Clarke, B. Barton, Entropy and MDL discretization of continuous variables for Bayesian belief networks. Int. J. Intell. Syst. 15, 61, 92 (2000).
109. E. Codd, Further normalization of the data base relational model. in Courant Computer Science Symposium 6: Data Base Systems, (Prentice-Hall, Englewood Cliff, 1971), pp. 33–64.
110. W. Cohen, Fast effective rule induction. in Proceedings of the 12th International Conference on Machine Learning, (Lake Tahoe, USA, 1995), pp. 115–123.
111. T. Colburn, G. Shute, Abstraction in computer science. Minds Mach. **17**, 169–184 (2007)

112. E. Colunga, L. Smith, The emergence of abstract ideas: evidence from networks and babies. Phil. Trans. Roy. Soc. B **358**, 1205–1214 (2003)
113. L. Console, D. Theseider-Dupré, Abductive reasoning with abstraction axioms. Lect. Notes Comput. Sci. **810**, 98–112 (1994)
114. S. Cook, The complexity of theorem proving procedures. in Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, (Shaker Heights, USA, 1971), pp. 151–158.
115. S. Coradeschi, A. Saffiotti, Anchoring symbols to sensor data: preliminary report. in Proceedings of the 17th National Conference on Artificial Intelligence, (Austin, USA, 2000), pp. 129–135.
116. L. Costa, F. Rodrigues, A. Cristino, Complex networks: the key to systems biology. Genet. Mol. Biol. **31**, 591–601 (2008)
117. P. Cousot, R. Cousot, Basic concepts of abstract interpretation. Build. Inf. Soc. **156**, 359–366 (2004)
118. V. Cross, Defining fuzzy relationships in object models: Abstraction and interpretation. Fuzzy Sets Syst. **140**, 5–27 (2003)
119. J. Crutchfield, N. Packard, Symbolic dynamics of noisy chaos. Physica D **7**, 201–223 (1983)
120. W. Daelemans, Abstraction considered harmful: lazy learning of language processing. in Proceedings of 6th Belgian-Dutch Conference on Machine Learning, (Maastricht, NL, 1996), pp. 3–12.
121. L. Danon, J. Duch, A. Diaz-Guilera, A. Artenas, Comparing community structure identification. J. Stat. Mech. Theor. Exp. P09008 (2005).
122. J. Davis, V. Costa, S. Ray, D. Page, An integrated approach to feature invention and model construction for drug activity prediction. in Proceedings of the 24th International Conference on Machine Learning, (Corvallis, USA, 2007), pp. 217–224.
123. P. Davis, R. Hillestad, Families of models that cross levels of resolution: issues for design, calibration and management. in Proceedings of the 25th Conference on Winter Simulation, (Los Angeles, USA, 1993), pp. 1003–1012.
124. R. Davis, Diagnostic reasoning based on structure and behavior. Artif. Intell. **24**, 347–410 (1984)
125. F. de Goes, S. Goldenstein, M. Desbrun, L. Velho, Exoskeleton: curve network abstraction for 3D shapes. Comput. Graph. **35**, 112–121 (2011)
126. J. de Kleer, B. Williams, Diagnosing multiple faults. Artif. Intell. **32**, 97–130 (1987)
127. M. de Vries, Engineering science as a “discipline of the particular”? Types of generalization in Engineering sciences. in ed. by I. van de Poel, D. Goldberg Philosophy and Engineering: An Emerging Agenda, (Springer, 2010), pp. 83–93.
128. T. Dean, R. Givan, Model minimization in Markov decision processes. in Proceedings of the National Conference on Artificial Intelligence, (Providence, USA, 1997), pp. 106–111.
129. D. DeCarlo, A. Santella, Stylization and abstraction of photographs. ACM Trans. Graph. **21**, 769–776 (2002)
130. M. Dehmer, L. Sivakumar, Recent developments in quantitative graph theory: information inequalities for networks. PLoS ONE **7**, e31395 (2012)
131. O. Dekel, S. Shalev-shwartz, Y. Singer, The Forgetron: A kernel-based perceptron on a fixed budget. in In Advances in Neural Information Processing Systems 18, (MIT Press, 2005), pp. 259–266.
132. A. Delorme, G. Richard, M. Fabre-Thorpe, Key visual features for rapid categorization of animals in natural scenes. Front. Psychol. **1**, 0021 (2010)
133. D. Dennett, *The Intentional Stance* (MIT Press, Cambridge, 1987)
134. K. Devlin, Why universities require computer science students to take math. Commun. ACM **46**, 37–39 (2003)
135. T. Dietterich, R. Michalski, Inductive learning of structural description. Artif. Intell. **16**, 257–294 (1981)
136. T. Dietterich, R. Michalski, A comparative review of selected methods for learning from examples, in Machine Learning: An Artificial Intelligence Approach, ed. by J. Carbonell, R. Michalski, T. Mitchell (Tioga Publishing, Palo Alto, 1983).

137. T. Dietterich, An overview of MAXQ hierarchical reinforcement learning. Lect. Notes Comput. Sci. **26–44**, 2000 (1864)
138. T. Dietterich, Machine Learning for sequential data: A review. in Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, (London, UK, 2002), pp. 15–30.
139. R. Dorat, M. Latapy, B. Conein, N. Auray, Multi-level analysis of an interaction network between individuals in a mailing-list. Ann. Telecommun. **62**, 325–349 (2007)
140. J. Dougherty, R. Kohavi, M. Sahami, Supervised and unsupervised discretization of continuous features. in Proceedings of the 12th International Conference on Machine Learning, (Tahoe City, USA, 1995), pp. 194–202.
141. G. Drastal, G. Czako, S. Raatz, Induction in an abstraction space. in Proceedings of the 11th International Joint Conference on Artificial Intelligence, (Detroit, USA, 1989), pp. 708–712.
142. E. Dubinsky, Reflective abstraction in advanced mathematical thinking, in *Advanced Mathematical Thinking*, ed. by D. Tall (Kluwer, Dordrecht, 1991), pp. 95–123
143. R. Durbin, S. Eddy, A. Krogh, G. Mitchison, *Biological Sequence Analysis* (Cambridge University Press, Cambridge, 1998)
144. P. Duygulu, M. Bastan, Multimedia translation for linking visual data to semantics in videos. Mach. Vis. Appl. **22**, 99–115 (2011)
145. S. Dzeroski, P. Langley, L. Todorovski, Computational discovery of scientific knowledge. Lect. Notes Artif. Intell. **4660**, 1–14 (2007)
146. T. Ellman, Synthesis of abstraction hierarchies for constraint satisfaction by clustering approximately equivalent objects. in Proceedings of the 10th International Conference on Machine Learning, (Amherst, USA, 1993), pp. 104–111.
147. F. Emmert-Streib, Statistic complexity: combining Kolmogorov complexity with an ensemble approach. PLoS ONE **5**, e12256 (2010)
148. F. Emmert-Streib, M. Dehmer, Exploring statistical and population aspects of network complexity. PlosOne **7**, e34523 (2012)
149. H. Enderton, A Mathematical Introduction to Logic, (Academic Press, 1972).
150. E. Engbers, M. Lindenbaum, A. Smeulders, An information-based measure for grouping quality. in Proceedings of the European Conference on Computer Vision, (Prague, Czech Republic, 2004), pp. 392–404.
151. D. Ensley, A hands-on approach to proof and abstraction. ACM SIGCSE Bull. **41**, 45–47 (2009)
152. S. Epstein, X. Li, Cluster graphs as abstractions for constraint satisfaction problems. in Proceedings of the 8th International Symposium on Abstraction, Reformulation, and Approximation, (Lake Arrowhead, USA, 2009), pp. 58–65.
153. J. Euzenat, On a purely taxonomic and descriptive meaning for classes. in Proceedings of the IJCAI Workshop on Object-Based Representation Systems, (Chambéry, France, 1993), pp. 81–92.
154. J. Euzenat, Représentation des connaissances: De l' Approximation à la Confrontation. Ph.D. thesis, (Université Joseph Fourier, Grenoble, France, 1999).
155. J. Euzenat, Granularity in relational formalisms with application to time and space representation. Comput. Intell. **17**, 703–737 (2001)
156. J. Euzenat, A. Montanari, Time granularity, in *Handbook of Temporal Reasoning in Artificial Intelligence*, ed. by M. Fisher, D. Gabbay, L. Vila (Elsevier, Amsterdam, 2005), pp. 59–118
157. P. Expert, T. Evans, V. Blondel, R. Lambiotte, Beyond space for spatial networks. PNAS **108**, 7663–7668 (2010)
158. M. Fabre-Thorpe, Visual categorization: accessing abstraction in non-human primates. Phil. Trans. Roy. Soc. B **358**, 1215–1223 (2003)
159. B. Falkenhainer, K. Forbus, D. Gentner, The structure-mapping engine: algorithm and examples. Artif. Intell. **41**, 1–63 (1989)
160. J. Fan, R. Samworth, Y. Wu, Ultrahigh dimensional feature selection: Beyond the linear model. J. Mach. Learn. Res. **10**, 2013–2038 (2009)

161. A. Feil, J. Mestre, Change blindness as a means of studying expertise in Physics. J. Learn. Sci. **19**, 480–505 (2010)
162. J. Feldman, How surprising is a simple pattern? Quantifying "Eureka!". Cognition **93**, 199–224 (2004)
163. A. Felner, N. Ofek, Combining perimeter search and pattern database abstractions. in Proceedings of the 7th International Symposium on Abstraction, Reformulation and Approximation, (Whistler, Canada, 2007), pp. 155–168.
164. S. Ferilli, T. Basile, N.D. Mauro, F. Esposito, On the learnability of abstraction theories from observations for relational learning. in Proceedings of European Conference on Machine Learning, (Porto, Portugal, 2005), pp. 120–132.
165. G. Ferrari, Vedi cosa intendo?, in Percezione, liguaggio, coscienza, Saggi di filosofia della mente, ed. by M. Carenini, M. Matteuzzi (Quodlibet, Macerata, Italy, 1999), pp. 203–224. In Italian.
166. P. Ferrari, Abstraction in mathematics. Phil. Trans. Roy. Soc. B **358**, 1225–1230 (2003)
167. R. Fikes, N. Nilsson, Strips: a new approach to the application of theorem proving to problem solving. Artif. Intell. **2**, 189–208 (1971)
168. K. Fine, *The Limit of Abstraction* (Clarendon Press, Oxford, 2002)
169. S. Fine, Y. Singer, N. Tishby, The hierarchical hidden markov model: analysis and applications. Mach. Learn. **32**, 41–62 (1998)
170. K. Fisher, J. Mitchell, On the relationship between classes, objects, and data abstraction. Theor. Pract. Object Syst. **4**, 3–25 (1998)
171. R. Fitch, B. Hengst, D. Šuc, G. Calbert, J. Scholz, Structural abstraction experiments in Reinforcement Learning. Lect. Notes Artif. Intell. **3809**, 164–175 (2005)
172. P. Flach, Predicate invention in inductive data Engineering. in Proceedings of the European Conference on Machine Learning, (Wien, Austria, 1993), pp. 83–94.
173. P. Flach, N. Lavrač, The role of feature construction in inductive rule learning. in Proceedings of the ICML Workshop on Attribute-Value and Relational Learning: Crossing the Boundaries, (Stanford, USA, 2000), pp. 1–11.
174. P. Flener, U. Schmid, Predicate invention, in *Encyclopedia of Machine Learning*, ed. by C. Sammut, G.I. Webb (Springer, USA, 2010), pp. 537–544
175. L. Floridi, The method of levels of abstraction. Minds Mach. **18**, 303–329 (2008)
176. L. Floridi, J. Sanders, The method of abstraction, in *Yearbook of the Artificial*, vol. 2, ed. by M. Negrotti (Peter Lang AG, Germany, 2004), pp. 178–220
177. G. Forman, An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. pp. 1289–1305 (2003).
178. S. Fortunato, C. Castellano, Community structure in graphs. Networks **814**, 42 (2007)
179. A. Frank, *A* (Asuncion, UCI Machine Learning Repository, 2010)
180. A. Frank, An operational meta-model for handling multiple scales in agent-based simulations. in Proceedings of the Dagstuhl Seminar, (Dagstuhl, Germany, 2012), pp. 1–6.
181. G. Frege, Rezension von E. Husserl: philosophie der Arithmetik. Zeitschrift fúr Philosophie und Philosophische Kritik 103, 313–332 (1894).
182. E. Freuder, Eliminating interchangeable values in constraint satisfaction problems. in Proceedings of the 9th National Conference of the American Association for Artificial Intelligence, (Anaheim, USA, 1991), pp. 227–233.
183. E. Freuder, D. Sabin, Interchangeability supports abstraction and reformulation for multi-dimensional constraint satisfaction. in Proceedings of 13th National Conference of the American Association for Artificial Intelligence, (Portland, USA, 1996), pp. 191–196.
184. G. Friedrich, Theory diagnoses: a concise characterization of faulty systems. in Proceedings of the 13th International Joint Conference on Artificial Intelligence, (Chambéry, France, 1993), pp. 1466–1471.
185. L. Frommberger, *Qualitative Spatial Abstraction in Reinforcement Learning*, (Springer, 2010)
186. M. Gabbrielli, S. Martini, Data abstraction, in *Programming Languages: Principles and Paradigms*, ed. by A. Tucker, R. Noonan (Springer, Heidelberg, 2010), pp. 265–276

187. U. Galassi, M. Botta, A. Giordana, Hierarchical hidden markov models for user/process profile learning. Fundam. Informaticae **78**, 487–505 (2007)
188. U. Galassi, A. Giordana, L. Saitta, Structured hidden markov models: a general tool for modeling agent behaviors, in *Soft Computing Applications in Business*, ed. by B. Prasad (Springer, Heidelberg, 2008), pp. 273–292
189. J. Gama, R. Sebastião, P. Rodrigues, Issues in evaluation of stream learning algorithms. in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (New York, USA, 2009), pp. 329–338.
190. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design patterns: abstraction and reuse of object-oriented design. Lect. Notes Comput. Sci. **707**, 406–431 (1993)
191. E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns (Elements of Reusable Object-Oriented Software* (Addison-Wesley Professional, Boston, 2005)
192. M. Garland, Multiresolution Modeling: Survey and Future Opportunities. Eurographics ’99-State of the Art Reports, pp. 111–131 (1999).
193. P. Gärdenfors, Language and the evolution of cognition. in Lund University Cognitive Studies, vol. 41. (Lund University Press, 1995).
194. P. Gärdenfors, *Conceptual Spaces* (MIT Press, Cambridge, 2004)
195. M. Gell-Mann, S. Lloyd, Effective complexity, in *Nonextensive Entropy-Interdisciplinary Applications*, ed. by M. Gell-Mann, C. Tsallis (Oxford University Press, Oxford, 2003), pp. 387–398
196. I. Gent, T. Walsh, Phase transitions from real computational problems. in Proceedings of the 8th International Symposium on Artificial Intelligence, (Monterrey, Mexico, 1995), pp. 356–364.
197. D. Gentner, Structure-mapping: a theoretical framework for analogy. Cogn. Sci. **7**, 155–170 (1983)
198. D. Gentner, Analogical reasoning, Psychology, in *Encyclopedia of Cognitive Science*, ed. by L. Nadel (Nature Publishing Group, London, 2003), pp. 106–112
199. D. Gentner, L. Smith, Analogical reasoning, in *Encyclopedia of Human Behavior*, ed. by V.S. Ramachandran 2nd, edn. (Elsevier, Oxford, 2012), pp. 130–136
200. L. Getoor, B. Taskar, *Introduction to Statistical Relational Learning* (The MIT Press, Cambridge, 2007)
201. C. Ghezzi, M. Jazayeri, D. Mandrioli, *Fundamentals of Software Engineering*, 2nd edn. (Pearson, NJ, 2003)
202. C. Ghidini, F. Giunchiglia, Local models semantics, or contextual reasoning = locality + compatibility. Artif. Intell. **127**, 221–259 (2001)
203. C. Ghidini, F. Giunchiglia, A semantic for abstraction. in Proceedings of 16th European Conf. on Artificial Intelligence, (Valencia, Spain, 2004), pp. 338–342.
204. M. Gick, K. Holyoak, Analogical problem solving. Cogn. Psychol. **12**, 306–355 (1980)
205. A. Gilpin, T. Sandholm, Lossless abstraction of imperfect information games. J. ACM **54**, 1–32 (2007)
206. A. Giordana, G. Lobello, L. Saitta, Abstraction in propositional calculus. in Proceedings of the Workshop on Knowledge Compilation and Speed Up Learning, (Amherst, USA, 1993), pp. 56–64.
207. A. Giordana, G. Peretto, D. Roverso, L. Saitta, Abstraction: An alternative view of concept acquisition, in *Methodologies for Intelligent Systems*, vol. 5, ed. by Z.W. Ras, M.L. Emrich (Elsevier, New York, 1990), pp. 379–387
208. A. Giordana, L. Saitta, Abstraction: a general framework for learning. in Working Notes of the AAAI Workshop on Automated Generation of Approximations and Abstractions, (Boston, USA, 1990), pp. 245–256.
209. A. Giordana, L. Saitta, Phase transitions in relational learning. Mach. Learn. **41**, 217–251 (2000)
210. A. Giordana, L. Saitta, D. Roverso, Abstracting concepts with inverse resolution. in Proceedings of the 8th International Machine Learning Workshop, (Evanston, USA, 1991), pp. 142–146.

211. P. Girard, R. Koenig-Robert, Ultra-rapid categorization of Fourier-spectrum equalized natural images: Macaques and humans perform similarly. Plos One **6**, e16453 (2011)
212. M. Girvan, M. Newman, Community structure in social and biological networks. PNAS **99**, 7821–7826 (2002)
213. F. Giunchiglia, A. Villafiorita, T. Walsh, Theories of abstraction. AI Commun. **10**, 167–176 (1997)
214. F. Giunchiglia, T. Walsh, A theory of abstraction. Artif. Intell. **57**, 323–389 (1992)
215. R. Givan, T. Dean, M. Greig, Equivalence notions and model minimization in Markov decision processes. Artif. Intell. **147**, 163–223 (2003)
216. R. Goldstone, L. Barsalou, Reuniting perception and conception. Cognition **65**, 231–262 (1998)
217. R. Golsdstein, V. Storey, Data abstraction: why and how? Data Knowl. Eng. **29**, 293–311 (1999)
218. M. Gordon, G. Scantlebury, Non-random polycondensation: statistical theory of the substitution effect. Trans. Faraday Soc. **60**, 604–621 (1964)
219. B. Gortais, Abstraction and art. Phil. Trans. Roy. Soc. B **358**, 1241–1249 (2003)
220. F. Gosselin, P.G. Schyns, Bubbles: a technique to reveal the use of information in recognition tasks. Vis. Res. **41**, 2261–2271 (2001)
221. P. Grassberger, Toward a quantitative theory of self-generated complexity. J. Theor. Phys. **25**, 907–938 (1986)
222. A. Grastien, G. Torta, A theory of abstraction for diagnosis of discrete-event systems. in Proceedings of the 9th International Symposium on Abstraction, Reformulation, and Approximation, (Cardona, Spain, 2011), pp. 50–57.
223. P. Grünwald, P. Vitànyi, Kolmogorov complexity and information theory. J. Logic Lang. Inf. **12**, 497–529 (2003)
224. M. Grimaldi, P. Cunningham, A. Kokaram, An evaluation of alternative feature selection strategies and ensemble techniques for classifying music. in Proceedings of the ECML Workshop on Multimedia Discovery and Mining, (Dubrovnik, Croatia, 1991).
225. C. Guestrin, M. Hauskrecht, B. Kveton, Solving factored MDPs with continuous and discrete variables. in Proceedings of the 20th International Conference on Uncertainty in Artificial Intelligence, (Portland, Oregon, USA, 2004), pp. 235–242.
226. C. Guestrin, D. Koller, R. Parr, Efficient solution algorithms for factored MDPs. J. Artif. Intell. Res. **19**, 399–468 (2003)
227. J. Guttag, Abstract data types and the development of data structures. Commun. ACM **20**, 396–404 (1977)
228. I. Guyon, A. Elisseef, An introduction to feature extraction, in *Feature Extraction, Foundations and Applications*, ed. by I. Guyon, S. Gunn, M. Nikravesh, L. Zadeh, Series Studies, in Fuzziness and Soft Computing, (Springer, New York, 2005), pp. 1–24
229. I. Guyon, A. Elisseeff, An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
230. B. Hale, *Abstract Objects* (Basil Blackwell, Oxford, UK, 1987)
231. G. Halford, W. Wilson, S. Phillips, Abstraction: nature, costs, and benefits. Int. J. Educ. Res. **27**, 21–35 (1997)
232. B. Hanczar, J.-D. Zucker, C. Henegar, L. Saitta, Feature construction from synergic pairs to improve microarray-based classification. Bioinformatics **23**, 2866–2872 (November 2007)
233. L. Harrie, R. Weibel, Modelling the overall process of Generalisation. in Generalisation of Geographic Information: Cartographic Modelling and Applications, ed. by W. Mackaness, A. Ruas, T. Sarjakoski (Elsevier, 2007), pp. 67–88.
234. D. Harry, D. Lindquist, *Graph Abstraction Through Centrality Erosion and k-Clique Minimization Technical Report* (Olin College, Needham, USA, 2004)
235. T. Hartley, N. Burgess, Models of spatial cognition. in Encyclopedia of Cognitive Science, (MacMillan, London, 2003), pp. 111–119.
236. L. Hartwell, J. Hopfield, S. Leibler, A. Murray, From molecular to modular cell biology. Nature **402**, C47–52 (1999)

237. P. Haslum, A. Botea, M. Helmert, B. Bonet, S. Koenig, Domain-independent construction of pattern database heuristics for cost-optimal planning. in Proceedings of the 22nd National Conference on Artificial Intelligence, (Vancouver, Canada, 2007), pp. 1007–1012.
238. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd edn. (Springer, New York, 2008)
239. X. He, S. Zemel, S. Richard, M. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (Washington, USA, 2004), pp. 695–703.
240. G. Hegel, *Phänomenologie des Geistes* (Bamberg und Würzburg, Germany, 1807)
241. C. Helma, R. King, S. Kramer, The predictive toxicology Challenge 2000–2001. Bioinformatics (Oxford, England) 17, 107–108 (2001).
242. M. Helmert, The fast downward planning system. J. Artif. Intell. Res. **26**, 191–246 (2006)
243. M. Helmert, P. Haslum, J. Hoffmann, Explicit-state abstraction: A new method for generating heuristic functions. in Proceedings of the 23rd National Conference on Artificial Intelligence, vol. 3, (Chicago, USA, 2008), pp. 1547–1550.
244. T. Hendriks, The impact of independent model formation on model-based service interoperability. in Proceedings of 7th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering, and Data Bases, (Cambridge, UK, 2008), pp. 434–441.
245. C. Henegar, R. Cancello, S. Rome, H. Vidal, K. Clement, J. Zucker, Clustering biological annotations and gene expression data to identify putatively co-regulated biological processes. J. Bioinform. Comput. Biol. **4**, 582–833 (2006)
246. B. Hengst, Safe state abstraction and reusable continuing subtasks in hierarchical Reinforcement Learning. Lect. Notes Comput. Sci. **4830**, 58–67 (2007)
247. B. Hengst, Generating hierarchical structure in Reinforcement Learning from state variables. in Proceedings of the Pacific Rim International Conference on Artificial Intelligence, (Malbourne, Australia, 2000), pp. 533–543.
248. P. Higgs, Broken symmetries, massless particles and gauge fields. Phys. Lett. **12**, 132 (1964)
249. J. Hill, B. Houle, S. Merritt, A. Stix, Applying abstraction to master complexity: The comparison of abstraction ability in computer science majors with students in other disciplines. in Proceedings of the 2nd International Workshop on The Role of Abstraction in Software Engineering, (Leipzig, Germany, 2008), pp. 15–21.
250. H. Hirsh, N. Japkowicz, Bootstrapping training-data representations for inductive learning: a case study in molecular biology. in Proceedings of the 12th National Conference on Artificial Intelligence, (Seattle, Washington, USA, 1994).
251. C. Hoare, Notes on data structuring. in APIC Studies in Data Processing, Structured Programming, vol. 8, (Academic Press, New York, 1972), pp. 83–174.
252. J. Hobbs, Granularity. in Proceedings of the 9th International Joint Conference on Artificial Intelligence, (Los Angeles, USA, 1985), pp. 432–435.
253. V. Hodge, J. Austin, A survey of outlier detection methodologies. Artif. Intell. Rev. **22**, 85–126 (2004)
254. J. Hoffmann, M. Zießler, The integration of visual and functional classification in concept formation. Phychol. Res. **48**, 69–78 (1986)
255. D. Hofstadter, *Fluid Concepts and Creative Analogies* (Harvester Wheatsheaf, London, 1995)
256. L. Holder, D. Cook, S. Djoko, Substructure discovery in the SUBDUE system. in Proceedings of the AAAI Workshop on Knowledge Discovery in Databases, (Seattle, USA, 1994), pp. 169–180.
257. R. Holte, B. Choueiry, Abstraction and reformulation in artificial intelligence. Phil. Trans. Roy. Soc. B **358**, 1197–1204 (2003)
258. R. Holte, J. Grajkowski, B. Tanner, Hierarchical heuristic search. in Proceedings of the 6th International Symposium on Abstraction, Approximation and Reformulation, (Airth Castle, Scotland, UK, 2005), pp. 121–133.
259. R. Holte, T. Mkadmi, R. Zimmer, A. MacDonald, Speeding up problem solving by abstraction: a graph oriented approach. Artif. Intell. **85**, 321–361 (1996)

260. R. Holte, M. Perez, R. Zimmer, A. MacDonald, Hierarchical A*: Searching abstraction hierarchies efficiently. in Proceedings of the National Conference on Artificial Intelligence, (Portland, USA, 1996), pp. 530–535.
261. K. Holyoak, P. Thagard, *Mental Leaps: Analogy in Creative Thought* (MIT Press, Cambridge, 1995)
262. A. Horn, On sentences which are true of direct unions of algebras. J. Symb. Logic **16**, 14–21 (1951)
263. D. Hostadter, Analogy as the core of cognition, in *The Analogical Mind: Perspectives from Cognitive Science*, ed. by D. Gentner, K.J. Holyoak, B.N. Kokinov (The MIT Press/Bradford Book, Cambridge, 2001), pp. 499–538
264. Z. Hu, J. Mellor, J. Wu, M. Kanehisa, J. Stuart, C. DeLisi, Towards zoomable multidimensional maps of the cell. Nat. Biotechnol. **25**, 547–554 (2007)
265. D. Huang, W. Pan, Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. Bioinformatics **22**, 1259–1268 (2006)
266. D. Huang, P. Wei, W. Pan, Combining gene annotations and gene expression data in model-based clustering: weighted method. OMICS **10**, 28–39 (2006)
267. J. Hunt, *Guide to the Unified Process featuring UML, Java and Design Patterns* (Springer, Heidelberg, 2003)
268. E. Husserl, *Philosophie der Arithmetik* (Pfeffer, Halle, 1891)
269. T. Imielinski, Domain abstraction and limited reasoning. in Proceedings of the 10th International Joint Conference on Artificial Intelligence, (Milan, Italy, 1987), pp. 997–1003.
270. I. Inza, P. Larrañaga, R. Blanco, A. Cerrolaza, Filter versus wrapper gene selection approaches in DNA microarray domains. Artif. Intell. Med. **31**, 91–103 (2004)
271. E. Itkonen, Analogy as Structure and Process. Approaches in Linguistics, Cognitive Psychology and Philosophy of Science, (John Benjamins, Amsterdam, 2005).
272. L. Itti, P. Baldi, Bayesian surprise attracts human attention. Vis. Res. **49**, 1295–1306 (2009)
273. A. Jain, Data clustering: 50 years beyond k-Means. Pattern Recognit. Lett. **31**, 651–666 (2010)
274. W. James, *The Principles of Psychology* (Dover Publications, New York, 1890)
275. N. Japkowicz, H. Hirsh, *Towards a bootstrapping approach to constructive induction* (In Working Notes of the Workshop on Constructive Induction and Change of Representation, New Brunswick, USA, 1994)
276. T. Joachims, Text categorization with support vector machines: learning with many relevant features. Lect. Notes Comput. Sci. **1398**, 137–142 (1998)
277. N.K. Jong, P. Stone, State abstraction discovery from irrelevant state variables. in Proceedings of the 19th International Joint Conference on Artificial Intelligence, (Edinburgh, Scotland, 2005), pp. 752–757.
278. T. Kahle, E. Olbrich, J. Jost, N. Ay, Complexity measures from interaction structures. Phys. Rev. E **79**, 026201 (2009)
279. V. Kandinsky, Point et ligne sur le plan, (Éditions Gallimard, Paris, France, 1991), In French.
280. I. Kant, Die Kritik der reinen Vernunft, (Johann Friedrich Hartknoch, Riga, Germany, 1781), In German
281. I. Kant, Critik der Urtheilskraft, (Lagarde und Friederich, Berlin, Germany, 1790), In German
282. E. Katsiri, A. Mycroft, Knowledge representation and scalable abstract reasoning for sentient computing using First-Order Logic. in Proceedings 1st Workshop on Challenges and Novel Applications for Automated Reasoning, (Miami, USA, 2002), pp. 73–82.
283. D. Kayser, Abstraction and natural language semantics. Phil. Trans. Roy. Soc. B **358**, 1261–1268 (2003)
284. S. Keller, On the use of case-based reasoning in generalization. in Proceedings of International Conference on Spatial Data Handling, (Edinburgh, Scotland, 1994), pp. 1118–1132.
285. K. Khan, S. Muggleton, R. Parson, Repeat learning using predicate invention. in Proceedings of the 8th International Workshop on Inductive Logic Programming, (Berlin, Germany, 1998), pp. 165–174.
286. J. Kietz, K. Morik, A polynomial approach to the constructive induction of structural knowledge. Mach. Learn. **14**, 193–217 (1994)

287. R.D. King, A. Srinivasan, L. Dehaspe, Warmr: a data mining tool for chemical data. J. Comput. Aided Mol. Des. **15**, 173–181 (2001)
288. Y. Kinoshita, K. Nishizawa, An algebraic semantics of predicate abstraction for PML. Inform. Media Technol. **5**, 48–57 (2010)
289. C. Knoblock, Automatically generating abstractions for planning. Artif. Intell. **68**, 243–302 (1994)
290. C. Knoblock, S. Minton, O. Etzioni, Integrating abstraction and explanation-based learning in PRODIGY. in Proceedings of the 9th National Conference on Artificial Intelligence, (Menlo Park, USA, 1991), pp. 541–546.
291. R. Kohavi, G. John, Wrappers for feature subset selection. Artif. Intell. **97**, 273–324 (1997)
292. R. Kohavi, M. Sahami, Error-based and entropy-based discretization of continuous features. in Proceedings of the 2nd Knowledge Discovery and Data Mining Conference, (Portland, USA, 1996), pp. 114–119.
293. S. Kok, P. Domingos, Statistical predicate invention. in Proceedings of the 24th International Conference on Machine Learning, (Corvallis, USA, 2007), pp. 433–440.
294. D. Koller, M. Sahami, Toward optimal feature selection. in Proceedings of the 13th International Conference on Machine Learning, (Bari, Italy, 1996), pp. 284–292.
295. A. Kolmogorov, Three approaches to the quantitative definition of information. Probl. Inf. Trans. **1**, 4–7 (1965)
296. M. Koppel, Complexity, depth, and sophistication. Complex Syst. **1**, 1087–1091 (1987)
297. R. Korf, Toward a model of representation changes. Artif. Intell. **14**, 41–78 (1980)
298. S. Kotsiantis, D. Kanellopoulos, Discretization techniques: a recent survey. GESTS Int. Trans. Comput. Sci. Eng. **32**, 47–58 (2006)
299. R. Kowalski, *Logic for Problem-Solving* (North-Holland Publising, Amsterdam, 1986)
300. O. Kozlova, O. Sigaud, C. Meyer, Texdyna: hierarchical Reinforcement Learning in factored MDPs. Lect. Notes Artif. Intell. **6226**, 489–500 (2010)
301. J. Kramer, Is abstraction the key to computing? Commun. ACM **50**, 37–42 (2007)
302. S. Kramer, Predicate invention: A comprehensive view. Technical Report OFAI-TR-95-32, Austrian Research Institute for Artificial Intelligence, (Vienna, 1995).
303. S. Kramer, N. Lavrač, P. Flach, Propositionalization approaches to relational data mining, in *Relational Data Mining*, ed. by S. Dzeroski, N. Lavrač (Springer, Berlin, 2001), pp. 262–291
304. S. Kramer, B. Pfahringer, C. Helma, Stochastic propositionalization of non-determinate background knowledge. Lect. Notes Comput. Sci. **1446**, 80–94 (1998)
305. M. Krogel, S. Rawles, F. Železný, P. Flach, N. Lavrač, S. Wrobel, Comparative evaluation of approaches to propositionalization. in Proceedings of the 13th International Conference on Inductive Logic Programming, (Szeged, Hungary, 2003), pp. 194–217.
306. Y. Kudoh, M. Haraguchi, Y. Okubo, Data abstractions for decision tree induction. Theor. Comput. Sci. **292**, 387–416 (2003)
307. P. Kuksa, Y. Qi, B. Bai, R. Collobert, J. Weston, V. Pavlovic, X. Ning, Semi-supervised abstraction-augmented string kernel for multi-level bio-relation extraction. in Proceedings of the European Conference on Machine Learning, (Barcelona, Spain, 2010), pp. 128–144.
308. M. Kurant, P. Thiran, Layered complex networks. Phys. Rev. Lett. **96**, 138701 (2006)
309. R. López-Ruiz, Statistical complexity and Fisher-Shannon information: Applications, in *K*, ed. by Statistical Complexity (New York, Sen (Springer, 2011), pp. 65–127
310. R. Lambiotte, Multi-scale modularity in complex networks. in it Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (Avignon, France, 2010) pp. 546–553.
311. A. Lancichinetti, S. Fortunato, Consensus clustering in complex networks. Sci. Rep. **2**, 336–342 (2012)
312. A. Lancichinetti, S. Fortunato, J. Kertesz, Detecting the overlapping and hierarchical community structure of complex networks. New J. Phys. **11**, 033015 (2009)
313. A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms. Phys. Rev. E **78**, 046110 (2008)
314. T. Lang, Rules for the robot droughtsmen. Geogr. Mag. **42**, 50–51 (1969)

315. P. Langley, *Scientific Discovery: Computational Explorations of the Creative Processes* (MIT Press, Cambridge, 1987)
316. P. Langley, The computer-aided discovery of scientific knowledge. Int. J. Hum-Comput. Stud. **53**, 393–410 (2000)
317. Y. Lasheng, J. Zhongbin, L. Kang, Research on task decomposition and state abstraction in Reinforcement Learning. Artif. Intell. Rev. **38**, 119–127 (2012)
318. N. Lavrač, P. Flach, An extended transformation approach to Inductive Logic Programming. ACM Trans. Comput. Log. **2**, 458–494 (2001)
319. N. Lavrač, J. Fürnkranz, D. Gamberger, Explicit feature construction and manipulation for covering rule learning algorithms, in *Advances in Machine Learning I*, ed. by J. Koronacki, Z. Ras, S. Wierzchon (Springer, New York, 2010), pp. 121–146
320. N. Lavrač, D. Gamberger, P. Turney, A relevancy filter for constructive induction. IEEE Intell. Syst. Their Appl. **13**, 50–56 (1998)
321. H. Laycock, Notes to object. in, Stanford Encyclopedia of Philosophy, 2010.
322. A. Lazaric, M. Ghavamzadeh, R. Munos, Analysis of a classification-based policy iteration algorithm. in Proceedings of the 27th International Conference on Machine Learning (Haifa, Israel, 2010), pp. 607–614.
323. H. Leather, E. Bonilla, M. O'Boyle, Automatic feature generation for Machine Learning based optimizing compilation. in Proceedings of International Symposium on Code Generation and Optimization (Seattle, 2009), pp. 81–91.
324. C. Lecoutre, Constraint Networks: Techniques and Algorithms (Wiley, 2009).
325. E. Leicht, M. Newman, Community structure in directed networks. Phys. Rev. Lett. **100**, 118703 (2008)
326. U. Leron, Abstraction barriers in Mathematics and Computer Science. in Proceedings of 3rd International Conference on Logo and Math Education (Montreal, Canada, 1987).
327. D. Levin, D. Simons, Failure to detect changes to attended objects in motion pictures. Psychon. Bull. Rev. **4**, 501–506 (1997)
328. A. Levy, Creating abstractions using relevance reasoning. in Proceedings of the 12th National Conference on Artificial Intelligence (Seattle, 1994), pp. 588–594.
329. D. Lewis, *On the Plurality of Worlds* (Basil Blackwell, Oxford, 1986)
330. L. Li, T. Walsh, M.L. Littman, Towards a unified theory of state abstraction for MDPs. in Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics (Fort Lauderdale, 2010), pp. 531–539.
331. M. Li, P. Vitànyi, *An Introduction to Kolmogorov Complexity and its Applications*, 2nd edn. (Springer, New York, 1997)
332. S. Li, M. Ying, Soft constraint abstraction based on semiring homomorphism. Theor. Comput. Sci. **403**, 192–201 (2008)
333. B. Liskov, J. Guttag, *Abstraction and Specification in Program Development* (MIT Press, Cambridge, 1986)
334. H. Liu, H. Motoda, On issues of instance selection. Data Min. Knowl. Discov. **6**, 115–130 (2002)
335. H. Liu, H. Motoda, R. Setiono, Z. Zhao, Feature selection: An ever evolving frontier in Data Mining. J. Mach. Learn. Res. **10**, 4–13 (2010)
336. H. Liu, R. Setiono, Feature selection via discretization. IEEE Trans. Knowl. Data Eng. **9**, 642–645 (1997)
337. H. Liu, F. Hussain, C.L. Tan, M. Dash, Discretization: An enabling technique. Data Min. Knowl. Discov. **6**, 393–423 (2002)
338. S. Lloyd, H. Pagels, Complexity as thermodynamic depth. Ann. Phys. **188**, 186–213 (1988)
339. J. Locke, Essay Concerning Human Understanding (Eliz Holt for Thomas Basset, London, 1690)
340. R. López-Ruiz, H. Mancini, X. Calbet, A statistical measure of complexity. Phys. Lett. A **209**, 321–326 (1995)
341. A. Lovett, K. Forbus, Modeling multiple strategies for solving geometric analogy problems. in Proceedings of the 34th Annual Conference of the Cognitive Science Society (Sapporo, Japan, 2012).

342. M. Lowry, The abstraction/implementation model of problem reformulation. in Proceedings of the 10th International Joint Conference on Artificial Intelligence (Milan, Italy, 1987), pp. 1004–1010.
343. M. Lowry, Algorithm synthesis through problem reformulation. in Proceedings of the 6th National Conf. on Artificial Intelligence (Seattle, 1987), pp. 432–436.
344. M. Lowry, Strata: Problem reformulation and ADT. in Proceedings of the 1st Workshop on Change of Representation and Inductive Bias (Briarcliff, 1988), pp. 29–50.
345. M. Lowry, M. Subramaniam, Abstraction for analytic verification of concurrent software systems. in Proceedings of the 8th Symposium on Abstraction, Reformulation, and Approximation (Asilomar, 1998), pp. 85–94.
346. S. Lozano, A. Arenas, A. Sànchez, Community connectivity and heterogeneity: Clues and insights on cooperation on social networks. J. Econ. Interact. Coord. **3**, 183–199 (2008)
347. H. Lu, D. Chen, K. Holyoak, Bayesian analogy with relational transformations. Psychol. Rev. **119**, 617–648 (2012)
348. D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson, R. Huebner, Level of Detail for 3D Graphics (Morgan Kaufmann, 2003).
349. W. Mackaness, A. Ruas, L. Sarjakoski, *Generalisation of Geographic Information: Cartographic Modelling and Applications* (Elsevier Science, Oxford, 2007)
350. S. Mannor, I. Menache, A. Hoze, U. Klein, Dynamic abstraction in Reinforcement Learning via clustering. in Proceedings of the 21st International Conference on Machine Learning, (Banff, Canada, 2004), pp. 71–78.
351. S. Markovitch, D. Rosenstein, Feature generation using general constructor functions. Mach. Learn. **49**, 59–98 (2002)
352. D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* (W.H. Freeman and Company, New York, 1982)
353. D. Marr, H. Nishihara, Representation and recognition of the spatial organization of three-dimensional shapes. Phil. Trans. Roy. Soc. B **200**, 269–294 (1978)
354. L. Martin, C. Vrain, Systematic predicate invention in inductive logic programming. in Proceedings of the International Workshop on Inductive Logic Programming, (Prague, Czech Republic, 1997), pp. 189–204.
355. K. Marx, *A Contribution to the Critique of Political Economy* (H. Kerr, Chicago, USA, 1904)
356. K. Marx, *Foundations of the Critique of Political Economy* (Harmondsworth, England, 1973)
357. C. Matheus, L. Rendell, Constructive induction on decision trees. in Proceedings of the 11th International Joint Conference on Artificial Intelligence, (Detroit, MI, USA, 1989), pp. 645–650.
358. M. Mazurowski, J. Malof, G. Tourassi, Comparative analysis of instance selection algorithms for instance-based classifiers in the context of medical decision support. Phys. Med. Biol. **56**, 473 (2010)
359. D. McDonald, A. Leung, W. Ferguson, T. Hussain, An abstraction framework for cooperation among agents and people in a virtual world. in Proceedings of Conference on Artificial Intelligence for the Interactive Digital Entertainment, (Marina del Rey, USA, 2006), pp. 54–59.
360. R. McMaster, Knowledge acquisition for cartographic generalization : experimental methods, in *GIS and Generalization*, ed. by J. Müller, R. Weibel, J. Lagrange (Taylor and Francis, London, 1995), pp. 161–180
361. R. McMaster, K. Shea, *Generalization in Digital Cartography* (Association of American Geographers, Washington, USA, 1992)
362. P. Mehra, L. Rendell, B. Wah, Principled constructive induction. in Proceedings of the 11th International Joint Conference on Artificial Intelligence, (Detroit, USA, 1989), pp. 651–656.
363. F. Melo, S. Meyn, M. Ribeiro, An analysis of Reinforcement Learning with function approximation. in Proceedings of the 25th International Conference on Machine Learning, (Helsinki, Finland, 2008), pp. 664–671.
364. C. Menschke, Robust elements in rough set abstractions. Lect. Notes Comput. Sci. 5548, 114–129

365. R. Michalski, R. Stepp, Revealing conceptual structure in data by inductive inference, in *Machine Intelligence*, vol. 10, ed. by J. Hayes, D. Michie, Y. Pao (Chichester, UK, Horwood, 1982), pp. 173–196
366. R. Michalski, Pattern recognition as knowledge-guided computer induction. Technical Report 927, Department of Computer Science, (University of Illinois, Urbana-Champaign, 1978).
367. R. Michalski, Pattern recognition as a rule-guided inductive inference. IEEE Trans. Pattern Anal. Mach. Intell. **2**, 349–361 (1980)
368. R. Michalski, K. Kaufman, Learning patterns in noisy data: The AQ approach, in *Machine Learning and Its Applications*, ed. by G. Paliouras, V. Karkaletsis, C. Spyropoulos (Springer, New York, 2001), pp. 22–38
369. D. Michie, S. Muggleton, D. Page, A. Srinivasan, To the international computing community: A new east-west challenge. in Research Report, (Oxford University Computing Laboratory, Oxford, 1994).
370. L. Miclet, S. Bayoudh, A. Delhay, Analogical dissimilarity. J. Artif. Intell. Res. **32**, 793–824 (2008)
371. J. Miles-Smith, D. Smith, Database abstraction: aggregation. Commun. ACM **20**, 405–413 (1977)
372. J. Miles-Smith, D. Smith, Data base abstractions: aggregation and generalization. ACM Trans. Database Syst. **2**, 105–133 (1977)
373. J. Mill, *A System of Logic* (University Press of the Pacific, Honolulu, USA, 2002)
374. M. Minsky, Steps toward artificial intelligence. Proc. IRE **49**, 8–30 (1961)
375. T.M. Mitchell, *Machine Learning* (McGraw-Hill, New York, 1997)
376. A. W. Moore, L. Baird, L.P. Kaelbling, Multi-value-functions: efficient automatic action hierarchies for multiple goal MDPs. in Proceedings of the International Joint Conference on Artificial Intelligence, (Stockholm, Sweden, 1999), pp. 1316–1323.
377. D. Morgan, The rise and fall of abstraction in the 18th Century art theory. Eighteenth-Century Stud. **27**, 449–478 (1994)
378. H. Motoda, H. Liu, Feature selection, extraction and construction. in Proceedings of the 6th Pacific-Asian Conference on Knowledge Discovery and Data Mining, (Taipei, Taiwan, 2002), pp. 67–72.
379. I. Mozetič, Hierarchical model-based diagnosis. Int. J. Man Mach. Stud. **35**, 329–362 (1991)
380. J. Mpindi, H.S. Haapa-Paananen, S. Kilpinen, T. Pisto, E. Bucher, K. Ojala, K. Iljin, P. Vainio, M. Björkman, S. Gupta, P. Kohonen, M. Nees, O. Kallioniemi, GTI: A novel algorithm for identifying outlier gene expression profiles from integrated microarray datasets. PLoS ONE **6**, e17259 (2011)
381. S. Muggleton, DUCE, an oracle-based approach to constructive induction. in Proceedings of the 10th International Joint Conference on Artificial Intelligence, (Milan, Italy, 1987), pp. 287–292.
382. S. Muggleton (ed.), *Inductive Logic Programming* (Academic Press, London, UK, 1992)
383. S. Muggleton, W. Buntine, Machine invention of first-order predicates by inverting resolution. in Proceedings of the 5th International Conference on Machine Learning, (Ann Arbor, USA, 1988), pp. 339–352.
384. S. Muggleton, L. de Raedt, Inductive logic programming: theory and methods. J. Logic Program. **19**, 629–679 (1994)
385. S. Muggleton, L. Raedt, D. Poole, I. Bratko, P. Flach, K. Inoue, A. Srinivasan, ILP turns 20. Mach. Learn. **86**, 3–23 (2011)
386. M. Mukherji, D. Kafura, A process-calculus-based abstraction for coordinating multi-agent groups. Theor. Comput. Sci. **192**, 287–314 (1998)
387. K. Murphy, M. Paskin, Linear time inference in hierarchical HMMs. Adv. Neural Inf. Process. Syst. **14**, 833–840 (2001)
388. S.K. Murthy, S. Kasif, S.L. Salzberg, A system for induction of oblique decision trees. J. Artif. Intell. Res. **2**, 1–32 (1994)
389. S. Mustière, GALBE: Adaptive generalization. The need for an adaptive process for automated generalization: An example on roads. in Proceedings of the GISPlaNet Conference, (Lisbon, Portugal, 1998).

390. S. Mustière, Apprentissage supervisé pour la généralisation cartographique. Ph.D. thesis, (University Pierre et Marie Curie, Paris, France, 2001), In French.
391. S. Mustière, Cartographic generalization of roads in a local and adaptive approach: a knowledge acquisition problem. Int. J. Geogr. Inf. Sci. **19**, 937–955 (2005)
392. S. Mustière, L. Saitta, J.-D. Zucker, Abstraction in cartographic generalization. Lect. Notes Artif. Intell. **638–644**, 2000 (1932)
393. S. Mustière, J.-D. Zucker, L. Saitta, An abstraction-based machine learning approach to cartographic generalization. in Proceedings of the International Conference on Spatial Data Handling, (Beijing, China, 2000), pp. 50–63.
394. L. Navarro, F. Flacher, V. Corruble, Dynamic level of detail for large scale agent-based urban simulations. in Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems, (Taipei, Taiwan, 2011), pp. 701–708.
395. P. Nayak, A. Levy, A semantic theory of abstractions. in Proceedings of the 14th International Joint Conference on Artificial Intelligence, (Montreal, Canada, 1995), pp. 196–203.
396. N. Neagu, S. Bistarelli, B. Faltings, Experimental evaluation of interchangeability in soft CSPs. Lect. Notes Artif. Intell. **3010**, 140–153 (2004)
397. A. Newell, Limitations of the current stock of ideas about problem solving. in Proceedings of Conference on Electronic Information Handling, (Washington, USA, 1965), pp. 195–208.
398. A. Newell, *Unified Theories of Cognition* (Harvard University Press, Cambridge, 1990)
399. M. Newman, The structure and function of complex networks. SIAM Rev. **45**, 167–256 (2003)
400. M. Newman, Fast algorithm for detecting community structure in networks. Phys. Rev. E **69**, 066133 (2004)
401. T.A.N. Nguyen, J.-D. Zucker, N.H. Du, A. Drogoul, D.-A. Vo, A hybrid macro-micro pedestrians evacuation model to speed up simulation in road networks. Lect. Notes Comput. Sci. **7068**, 371–383 (2011)
402. N. Nilsson, *Artificial Intelligence: A New Synthesis* (Morgan Kaufmann, San Francisco, USA, 1998)
403. H. Noonan, Count nouns and Mass nouns. Analysis **38**, 167–172 (1978)
404. D. Nyerges, Representing geographic meaning, in *Map Generalization*, ed. by B. Buttenfield, R. McMaster, H. Freeman (Essex, UK, Longman Scientific and Technical, Harlow , 1991), pp. 59–85
405. D. O'Donoghue, M. Keane, A creative analogy machine: Results and challenges. in Proceedings of the Internernational Conference on Computational Creativity, (Dublin, Ireland, 2012), pp. 17–24.
406. K. Oehler, R. Gray, Combining image compression and classification using vector quantization. IEEE Trans. Pattern Anal. Mach. Intell. **17**, 461–473 (1995)
407. S. Oliveira, S. Seok, A multilevel approach to identify functional modules in a yeast protein-protein interaction network. Lect. Notes Comput. Sci. **3992**, 726–733 (2006)
408. C. Ortiz-Hill, Frege's attack on Husserl and Cantor. Monist **77**, 345–357 (1994)
409. F. Pachet, P. Roy, Analytical features: a knowledge-based approach to audio feature generation. EURASIP J. Audio Speech Music Process. **2009**, 1–23 (2009)
410. G. Pagallo, D. Haussler, Two algorithms that learn DNFs by discovering relevant features. in Proceedings of the 6th International Workshop on Machine Learning, (Ithaca, New York, USA, 1989), pp. 119–123.
411. B. Pang, R. Holte, Multimapping abstractions and hierarchical heuristic search. in Proceedings of the 5th Symposium on Combinatorial Search, (Niagara Falls, Canada, 2012).
412. S. Pantazi, J. Arocha, J. Moehr, Case-based medical informatics. in Intelligent Paradigms in Healthcare Enterprises, ed. by B. Silverman, A. Jain, A. Ichalkaranje, L. Jain (Springer, 2005), pp. 31–65.
413. R. Parr, A unifying framework for temporal abstraction in stochastic processes. in Proceedings of the 8th International Symposium on Abstraction, Reformulation, and Approximation, (Asilomar, USA, 1998), pp. 95–102.
414. Z. Pawlak, Rough sets. Int. J. Parallel Programm. **11**, 341–356 (1982)

415. J. Pearl, On the connection between the complexity and the credibility of inferred models. Int. J. Gen. Syst. **4**, 255–264 (1978)
416. C. Perlich, F. Provost, Distribution-based aggregation for relational learning with identifier attributes. Mach. Learn. **62**, 65–105 (2006)
417. J. Piaget, *Genetic epistemology* (Columbia University Press, New York, 1968)
418. S. Piramuthu, R.T. Sikora, Iterative feature construction for improving inductive learning algorithms. Expert Syst. Appl. **36**, 3401–3406 (2009)
419. D. Plaisted, Theorem proving with abstraction. Artif. Intell. **16**, 47–108 (1981)
420. Plato. $\Pi o\lambda\iota\tau\epsilon\iota\alpha$ (Republic), 7.514a. 380 BC
421. J. Platt, Prediction of isomeric differences in paraffin properties. J. Phys. Chem. **56**, 328–336 (1952)
422. C. Plazanet, Enrichissement des bases de données géographiques : Analyse de la géométrie des objets linéaires pour la généralisation cartographique (Application aux routes). Ph.D. thesis, (University Marne-la-Vallée, France, 1996), In French.
423. G. Plotkin, A further note on inductive generalization. in Machine Intelligence, vol 6, (Edinburgh University Press, 1971).
424. G. Polya, *How to Solve It: A New Aspect of Mathematical Methods* (Princeton University Press, Princeton, 1945)
425. M. Ponsen, M. Taylor, K. Tuyls, Abstraction and generalization in reinforcement learning: a summary and framework. Lect. Notes Comput. Sci. **5924**, 1–32 (2010)
426. K. Popper, *The Logic of Scientific Discovery* (Harper Torch, New York, 1968)
427. M. Poudret, A. Arnould, J. Comet, P.L. Gall, P. Meseure, F. Képès, Topology-based abstraction of complex biological systems: application to the Golgi apparatus. Theor. Biosci. **127**, 79–88 (2008)
428. W. Prenninger, A. Pretschner, Abstraction for model-based testing. Electron. Notes Theore. Comput. Sci. **116**, 59–71 (2005)
429. A. Prieditis, Machine discovery of admissible heuristics. Mach. Learn. **12**, 117–142 (1993)
430. E. Prifti, J.D. Zucker, K. Clement, C. Henegar, Interactional and functional centrality in transcriptional co-expression networks. Bioinform. **26**(24), 3083–3089 (2010)
431. P. Prosser, An empirical study of phase transitions in constraint satisfaction problems. Artif. Intell. **81**, 81–109 (1996)
432. G. Provan, Hierarchical model-based diagnosis. in Proceedings of the 12th International Workshop on Principles of Diagnosis, (Murnau, Germany, 2001), pp. 167–174.
433. J. Provost, B.J. Kuipers, R. Miikkulainen, Developing navigation behavior through self-organizing distinctive state abstraction. Connection Sci. **18**, 159–172 (2006)
434. L. Pyeatt, A. Howe, Decision tree function approximation in Reinforcement Learning. in Proceedings of the 3rd International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Models, (Havana, Cuba, 2001), pp. 70–77.
435. Z. Pylyshyn, What the mind's eye tells the mind's brain: a critique of mental imagery. Psychol. Bull. **80**, 1–24 (1973)
436. Z. Pylyshyn, *Computation and Cognition: Toward a Foundation for Cognitive Science* (MIT Press, Cambridge, 1984)
437. W. Quine, *Word and Object* (MIT Press, Cambridge, 1960)
438. J. Quinlan, R. Cameron-Jones, Induction of logic programs: Foil and related systems. New Gen. Comput. **13**, 287–312 (1995)
439. J.R. Quinlan, R.M. Cameron-Jones, Foil: A midterm report. Lect. Notes Comput. Sci. **667**, 3–20 (1993)
440. R. Quinlan, Induction of decision trees. Mach. Learn. **1**, 81–106 (1986)
441. L. Rabiner, A tutorial on Hidden Markov Models and selected applications in speech recognition. Proc. IEEE **77**, 257–286 (1989)
442. M. Ramscar, D. Yarlett, Semantic grounding in models of analogy: an environmental approach. Cogn. Sci. **27**, 41–71 (2003)
443. E. Ravasz, A. Barabasi, Hierarchical organization in complex networks. Phys. Rev. E **67**, 026112 (2003)

444. B. Ravindran, A. Barto, Model minimization in hierarchical reinforcement learning. Lect. Notes Comput. Sci. **2371**, 196–211 (1985)
445. G. Rücker, C. Rücker, Substructure, subgraph, and walk counts as measures of the complexity of graphs and molecules. J. Chem. Inf. Comput. Sci. **41**, 1457–1462 (2001)
446. N. Regnauld, Généralisation du bâti: Structure spatiale de type graphe et représentation cartographique, Ph.D. thesis, (Université de Provence-Aix-Marseille 1, 1998), In French.
447. N. Regnauld, R. McMaster, A synoptic view of generalisation operators. Generalisation Geogr. Inf. **37–66**, 2007 (2007)
448. R. Reiter, On closed world data bases, in *Logic and Data Bases*, ed. by H. Gallaire, J. Minker (Plenum Press, New York, 1978), pp. 119–140
449. R. Reiter, A theory of diagnosis from first principles. Artif. Intell. **32**, 57–96 (1987)
450. L. Rendell, A scientific approach to practical induction, in *Machine learning: A guide to current research*, ed. by T. Mitchell, J. Carbonell, R. Michalski (Kluwer Academic Publishers, Norwell, USA, 1986), pp. 269–274
451. A. Rendl, I. Miguel, P. Gent, P. Gregory, Common subexpressions in constraint models of planning problems. in Proceedings of the 8th International Symposium on Abstraction, Reformulation, and Approximation, (Lake Arrowhead, USA, 2009), pp. 143–150.
452. A. Rensink, E. Zambon, Neighbourhood abstraction in GROOVE. Electron. Commun. EASST **32**, 1–13 (2010)
453. R. Rensink, J. O'Regan, J. Clark, On the failure to detect changes in scenes cross brief interruptions. Visual Cognition **7**, 127–146 (2000)
454. L. Rising, Understanding the power of abstraction in patterns. IEEE Softw. **24**, 46–51 (2007)
455. G. Roşu, Behavioral abstraction is hiding information. Theor. Comput. Sci. **327**, 197–221 (2004)
456. P. Ronhovde, Z. Nussinov, Multiresolution community detection for megascale networks by information-based replica correlations. Phys. Rev. E **80**, 016109 (2009)
457. G. Rosen, Abstract objects. in The Stanford Encyclopedia of Philosophy, ed. by E. Zalta (2009).
458. F. Rossi, K. Venable, T. Walsh, *A Short Introduction to Preferences: Between Artificial Intelligence and Social Choice* (Morgan and Claypool Publishing, San Rafael, USA, 2011)
459. A. Ruas, Modèles de Généralisation de Données Géographiques: Base de Contraintes et d'Autonomie, Ph.D. thesis, (University of Marne-la-Vallée, France, 1999), In French.
460. A. Ruas, Automatic Generalization Project: Learning Process from Interactive Generalization (OEEPE Official Publication n. 39, 2001).
461. L. Sacchi, C. Larizza, C. Combi, R. Bellazzi, Data Mining with temporal abstractions: learning rules from time series. Data Min. Knowl. Discov. **15**, 217–247 (2007)
462. E. Sacerdoti, Planning in a hierarchy of abstration spaces. Artif. Intell. **5**, 115–135 (1974)
463. M. Sachenbacher, P. Struss, Task-dependent qualitative domain abstraction. Artif. Intell. **162**, 121–143 (2004)
464. S.D. Saeger, A. Shimojima, Channeling abstraction. in Proceedings of the International Symposium on Abstraction, Reformulation and Approximation, (Whistler, Canada, 2007), pp. 133–147.
465. Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics. Bioinformatics **23**, 2507 (2007)
466. L. Saitta, C. Henegar, J. Zucker, Abstracting complex interaction networks. in Proceedings of the 8th Symposium on Abstraction, Reformulation, and Approximation, (Lake Arrowhead, USA, 2009), pp. 821–825.
467. L. Saitta, P. Torasso, G. Torta, Formalizing the abstraction process in model-based diagnosis. Lect. Notes Comput. Sci. **4612**, 314–328 (2007)
468. L. Saitta, J.-D. Zucker, Semantic abstraction for concept representation and learning. in Proceedings of the International Symposium on Abstraction, Approximation and Reformulation, (Pacific Grove, USA, 1998), pp. 103–120.
469. L. Saitta, J.-D. Zucker, Abstraction and phase transitions in relational learning. Lect. Notes Comput. Sci. **1864**, 291–302 (2000)

470. L. Saitta, J.-D. Zucker, A model of abstraction in visual perception. Int. J. Appl. Intell. **80**, 134–155 (2001)
471. L. Saitta, J.-D. Zucker, Abstraction and complexity measures. Lect. Notes Comput. Sci. **4612**, 375–390 (2007)
472. L. Saitta, C. Vrain, Abstracting Markov networks. in Proceedings of the AAAI Workshop on Abstraction, Reformulation, and Approximation, (Atlanta, Georgia, USA, 2010).
473. L. Saitta (ed.), The abstraction paths. Special Issue of the Philos. Trans. Roy. Soc. B 358, 1435 (2003).
474. M. Sales-Pardo, R. Guimerà, A. Moreira, L.N. Amaral, Extracting the hierarchical organization of complex systems. PNAS **104**, 15224–15229 (2007)
475. C. Sammut (ed.), *Encyclopedia of Machine Learning* (Springer, New York, 2011)
476. J. Schlimmer, Learning and representation change. in Proceedings of the 6th National Conference on, Artificial Intelligence, pp. 511–535 (1987).
477. J. Schmidhuber, Low-complexity art. J. Int. Soc. Arts Sci. Technol. **30**, 97–103 (1997)
478. H. Schmidtke, W. Woo, A size-based qualitative approach to the representation of spatial granularity. in Proceedings of the 20th International Joint Conference on Artificial Intelligence, (Bangalore, India, 2007), pp. 563–568.
479. R. Schrag, D. Miranker, Abstraction in the CSP phase transition boundary. in Proceedings of the 4th International Symposium on Artificial Intelligence and Mathematics, (Ft. Lauderdale, USA, 1995), pp. 126–133.
480. J. Seligman, From logic to probability. Lect. Notes Comput. Sci. **5363**, 193–233 (2009)
481. R. Serna Oliver, I. Shcherbakov, G. Fohler, An operating system abstraction layer for portable applications in wireless sensor networks. in Proceedings of the ACM Symposium on Applied Computing, (Sierre, Switzerland, 2010), pp. 742–748.
482. A. Sfard, L. Linchevsky, The gains and pitfalls of reification-the case of alegbra. Educ. Stud. Math. **26**, 191–228 (1994)
483. C. Shannon, The mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423 (1948)
484. A. Sharpanskykh, Agent-based modeling and analysis of socio-technical systems. Cybern. Syst. **42**, 308–323 (2011)
485. S. Shekhar, C. Lu, P. Zhang, A unified approach to detecting spatial outliers. GeoInformatica **7**, 139–166 (2003)
486. J. Shi, M. Littman, Abstraction methods for game theoretic poker. in Proceedings of the 2nd International Conference on Computers and Games, (Hamamatsu, Japan, 2001), pp. 333–345.
487. J. Shiner, M. Davison, P. Landsberg, Simple measure of complexity. Phys. Rev. E **39**, 1459–1464 (1999)
488. G. Silverstein, M. Pazzani, Relational clichés: Constraining constructive induction during relational learning. in Proceedings of the 8th International Workshop on Machine Learning, (Evanston, USA, 1991), pp. 203–207.
489. G. Simmons, *Shapes, Part Structure and Object Concepts, in Proceedings of the ECAI Workshop on Parts and Wholes: Conceptual Part-Whole Relationships and Formal Mereology* (Nederlands, Amsterdam, 1994)
490. H. Simon, *The Sciences of the Artificial*, 3rd edn. (MIT Press, Cambridge, 1999)
491. D. Simons, Current approaches to change blindness. Vis. Cogn. **7**, 1–15 (2000)
492. D. Simons, C. Chabris, T. Schnur, Evidence for preserved representations in change blindness. Conscious. Cogn. **11**, 78–97 (2002)
493. Ö. Simsek, Workshop summary: abstraction in reinforcement learning. in Proceedings of the International Conference on Machine Learning, (Montreal, Canada, 2009), p. 170.
494. M. Sizintsev, R. Wildes, Coarse-to-fine stereo vision with accurate 3D boundaries. Image Vis. Comput. **28**, 352–366 (2010)
495. B. Smith, M. Dyer, Locating the phase transition in binary constraint satisfaction problems. Artif. Intell. **81**, 155–181 (1996)
496. R.M. Smullyan, *First-Order Logic* (Dover Publications, Mineola, 1995)

497. N.N. Soja, S. Carey, E. Spelke, Ontological categories guide young children's inductions of word meaning: Object terms and substance terms. Cognition **38**, 179–211 (1991)
498. R. Solomonoff, A formal theory of inductive inference-Part I. Inf. Contl. **7**, 1–22 (1964)
499. R. Solomonoff, A formal theory of inductive inference-Part II. Inf. Contl. **7**, 224–254 (1964)
500. J. Sowa, A. Majumdar, Conceptual structures for knowledge creation and communication. in Proceedings of the International Conference on Conceptual Structures, (Dresden, Germany, 2012), pp. 17–24.
501. A. Srinivasan, S. Muggleton, M. Bain, Distinguishing exceptions from noise in non-monotonic learning. in Proceedings of the 2nd International Workshop on Inductive Logic Programming, (Tokyo, Japan, 1992), pp. 203–207.
502. S. Srivastava, N. Immerman, S. Zilberstein, Abstract planning with unknown object quantities and properties. in Proceedings of the 8th Symposium on Abstraction, Reformulation, and Approximation, (Lake Arrowhead, USA, 2009), pp. 143–150.
503. M. Stacey, C. McGregor, Temporal abstraction in intelligent clinical data analysis: a survey. Artif. Intell. Med. **39**, 1–24 (2007)
504. I. Stahl, Predicate invention in ILP-An overview. in Proceedings of the European Conference on Machine Learning, (Vienna, Austria, 1993), pp. 311–322.
505. I. Stahl, The appropriateness of predicate invention as bias shift operation in ILP. Mach. Learn. **20**, 95–117 (1995)
506. F. Staub, E. Stern, Abstract reasoning with mathematical constructs. Int. J. Educ. Res. **27**, 63–75 (1997)
507. M. Stefik, Planning with constraints (MOLGEN: Part 1). Artif. Intell. **16**, 111–139 (1981)
508. M. Stolle, D. Precup, Learning options in reinforcement learning. Lect. Notes Comput. Sci. **2371**, 212–223 (2002)
509. J.V. Stone, *Computer vision: What is the object? in Proceedings of the Artificial Intelligence and Simulation of Behaviour Conference* (Birmingham, UK, 1993), pp. 199–208
510. P. Struss, A. Malik, M. Sachenbacher, Qualitative modeling is the key to automated diagnosis. in Proceedings of the 13st World Congress of the International Federation of Automatic Control, (San Francisco, USA, 1996).
511. S. Stylianou, M.M. Fyrillas, Y. Chrysanthou, Scalable pedestrian simulation for virtual cities. in Proceedings of the ACM Symposium on Virtual Reality software and technology, (New York, USA, 2004), pp. 65–72.
512. D. Subramanian, A theory of justified reformulations. in Change of Representation and Inductive Bias, ed. by P. Benjamin (Kluwer Academic Press, 1990), pp. 147–168.
513. D. Subramanian, R. Greiner, J. Pearl, The relevance of relevance (editorial). Artif. Intell. **97**, 1–2 (1997)
514. S. Sun, N. Wang, Formalizing the multiple abstraction process within the G-KRA model framework. in Proceedings of the International Conference on Intelligent Computing and Integrated Systems, (Guilin, China, 2010), pp. 281–284.
515. S. Sun, N. Wang, D. Ouyang, General KRA abstraction model. J. Jilin Univ. 47, 537–542 (2009). In Chinese.
516. R. Sutton, D. Precup, S. Singh, Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. Artif. Intell. **112**, 181–211 (1999)
517. R. S. Sutton, E. J. Rafols, A. Koop, Temporal abstraction in temporal-difference networks. in Proceedings of the NIPS-18, (Vancouver, Canada, 2006), pp. 1313–1320.
518. R. Sutton, Generalization in Reinforcement Learning: Successful examples using sparse coarse coding. Advances in Neural Information Processing Systems, pp. 1038–1044 (1996).
519. R. Sutton, A. Barto, *Reinforcement Learning* (MIT Press, Cambridge, 1998)
520. R. Sutton, D. McAllester, S. Singh, Y. Mansour, Policy gradient methods for Reinforcement Learning with function approximation. Adv. NIPS **12**, 1057–1063 (2000)
521. A. Swearngin, B. Choueiry, E. Freuder, A reformulation strategy for multi-dimensional CSPs: The case study of the SET game. in Proceedings of the 9th International Symposium on Abstraction, Reformulation and Approximation, (Cardona, Spagna, 2011), pp. 107–116.

522. B. Sylvand, Une brève histoire du concept de "concept". Ph.D. thesis, (Université La Sorbonne, Paris, France, 2006), In French.
523. C. Szepesvári, Algorithms for Reinforcement Learning, (Morgan & Claypool, 2010).
524. M.E. Taylor, P. Stone, Transfer learning for reinforcement learning domains: a survey. J. Mach. Learn. Res. **10**, 1633–1685 (2009)
525. J. Tenenberg, Abstraction in Planning, Ph.D. thesis, (Universtiy of Rochester, USA, 1988).
526. J. Tenenberg, Preserving consistency across abstraction mappings. in Proceedings 10th International Joint Conference on Artificial Intelligence (Milan, Italy, 1987), pp. 1011–1014.
527. B. ter Haar Romeny, Designing multi-scale medical image analysis algorithms. in Proceedings of the International Conference on Pattern Recognition (Tutorial) (Istanbul, Turkey, 2010).
528. C. Thinus-Blanc, *Animal Spatial Cognition* (World Scientific Publishing, Singapore, 1996)
529. R. Tibshirani, Regression shrinkage and selection via the lasso. J. Roy. Stat. Soc. Ser. B (Methodological) **58**, 267–288 (1996)
530. A. Tollner-Burngasser, M. Riley, W. Nelson, Individual and team susceptibility to change blindness. Aviation Space Environ. Med. **81**, 935–943 (2010)
531. P. Torasso, G. Torta, Automatic abstraction of time-varying system models for model based diagnosis. Lect. Notes Artif. Intell. **3698**, 176–190 (2005)
532. G. Torta, P. Torasso, Automatic abstraction in component-based diagnosis driven by system observability. in Proceedings of the 18th International Joint Conference on Artificial Intelligence, (Acapulco, Mexico, 2003), pp. 394–400.
533. G. Torta, P. Torasso, Qualitative domain abstractions for time-varying systems: an approach based on reusable abstraction fragments. in Proceedings of the 17th International Workshop on Principles of Diagnosis (Peñaranda de Duero, Spain, 2006), pp. 265–272.
534. V. Truppa, E.P. Mortari, D. Garofoli, S. Privitera, E. Visalberghi, Same/different concept learning by Capuchin monkeys in matching-to-sample tasks. PLoS One **6**, e23809 (2011)
535. J. Tsitsiklis, B. Van Roy, An analysis of temporal-difference learning with function approximation. IEEE Trans. Autom. Contl. **42**, 674–690 (1997)
536. P. Turney, A uniform approach to analogies, synonyms, antonyms, and associations. in Proceedings of the International Conference on Computational Linguistics, vol. 1, (Manchester, UK, 2008), pp. 905–912.
537. E. Tuv, A. Borisov, G. Runger, K. Torkkola, Feature selection with ensembles, artificial variables, and redundancy elimination. J. Mach. Learn. Res. **10**, 1341–1366 (2009)
538. B. Tversky, K. Hemenway, Objects, parts, and categories. J. Exp. Phychol. Gen. **113**, 169–193 (1984)
539. J. Ullman, *Principles of Databases* (Computer Science, Baltimore, 1982)
540. J. Ullman, Implementation of logical query languages for databases. ACM Trans. Database Syst. **10**, 298–321 (1985)
541. S. Ullman, Visual routines. Cognition **18**, 97–159 (1984)
542. P.E. Utgoff, D.J. Stracuzzi, Many-layered learning. Neural Comput. **14**, 2497–2529 (2002)
543. R. Valdés-Pérez, Principles of human-computer collaboration for knowledge discovery in science. Artif. Intell. **107**, 335–346 (1999)
544. M. Valtorta, A result on the computational complexity of heuristic estimates for the A^* algorithm. Inf. Sci. **34**, 48–59 (1984)
545. D. van Dalen, *Logic and Structure*, 4th edn. (Springer, New York, 2004)
546. P. Vitányi, Meaningful information. in Proceedings of the 13th International Symposium on Algorithms and Computation, (Vancouver, Canada, 2002), pp. 588–599.
547. D. Vo, A. Drogoul, J.-D. Zucker, An operational meta-model for handling multiple scales in agent-based simulations. in Proceedings of the International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (Ho Chi Minh City, Vietnam, 2012), pp. 1–6.
548. P. Vogt, The physical symbol grounding problem. Cogn. Syst. Res. **3**, 429–457 (2002)
549. F. Wang, On the abstraction of conventional dynamic systems: from numerical analysis to linguistic analysis. Inf. Sci. **171**, 233–259 (2005)

550. N. Wang, D. Ouyang, S. Sun, Formalizing ontology-based hierarchical modeling process of physical world. Lect. Notes Comput. Sci. **6319**, 18–24 (2010)
551. N. Wang, D. Ouyang, S. Sun, Hierarchical abstraction process in model-based diagnosis. Chinese J. Comput. 34, 383–394 (2011). In Chinese.
552. S. Watanabe, *Knowing and Guessing: Quantitative Study of Inference and Information* (Wiley, New York, 1969)
553. R. Weibel, D. Burghardt, On-the-fly generalization. Encyclopedia of GIS (Springer, New York, 2008), pp. 339–344.
554. R. Weibel, S. Keller, T. Reichenbacher, Overcoming the knowledge acquisition bottleneck in map generalization: the role of interactive systems and computational intelligence. in Proceedings of the Conference on Spatial Information Theory (Semmering, Austria, 1995), pp. 139–156.
555. W. Weigel, B. Faltings, B. Choueiry, Context in discrete Constraint Satisfaction Problems. in Proceedings of the European Conference on Artificial Intelligence (Budapest, Hungay, 1996), pp. 205–209.
556. H. Welling, Four mental operations in creative cognition: the importance of abstraction. Creativity Res. J. **19**, 163–177 (2007)
557. T. Werschlein, R. Weibel, Use of neural networks in line generalisation. in Proceedings of 5th European Conference and Exhibition on Geographical Information Systems (Paris, France, 1994), pp. 77–85.
558. M. Wertheimer, über Gestalttheorie. Philosophische Zeitschrift für Forschung und Aussprache, 1:39–60 (1925). In German.
559. B. Weslake, Explanatory depth. Philos. Sci. **77**, 273–294 (2010)
560. J. Weston, A. Bordes, L. Bottou. Online (and offline) on an even tighter budget. in Proceedings of the 10th International Workshop on Artificial Intelligence and, Statistics pp. 413–420 (2005).
561. C. Williams, T. Hogg, Exploiting the deep structure of constraint problems. Artif. Intell. **70**, 73–117 (1994)
562. D. Wilson, T. Martinez, Reduction techniques for instance-based learning algorithms. Mach. Learn. **38**, 257–286 (2000)
563. P. Winston, Learning structural descriptions from examples. in The Psychology of Computer Vision, ed. by P. Winston (McGraw-Hill, 1975), pp. 157–209.
564. R. Wirth, Completing logic programs by inverse resolution. in Proceedings of the 4th European Working Session on Learning (Montpellier, France, 1989), pp. 239–250.
565. I. Witten, E. Frank, M. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. (Morgan Kaufman, 2011).
566. J. Wneck, R. Michaslki, Hypothesis-driven constructive induction in AQ17-HCI: a method and experiments. Mach. Learn. **14**, 139–168 (1994)
567. J. Wogulis, P. Langley, Efficiency by learning intermediate concepts. in Proceedings of the 6th International Conference on Machine Learning (Ithaca, USA, 1989), pp. 78–80.
568. D. Wolpert, W. Macready, Self-dissimilarity: An empirically observable complexity measure. in Proceedings of the International Conference on Complex Systems (Nashua, USA, 1997), pp. 625–643.
569. C. Wright, *Frege's Conception of Numbers as Objects* (Aberdeen University Press, Aberdeen, Scotland, 1983)
570. K. Xu, W. Li, Exact phase transitions in random constraint satisfaction problems. J. Artif. Intell. Res. **12**, 93–103 (2000)
571. K. Xu, W. Li, Many hard examples in exact phase transitions. Theor. Comput. Sci. **355**, 291–302 (2006)
572. B. Yang, M. Zhang, G. Xie, Abstract interpretation theory and its application. Comput. Eng. Appl. **46**, 16–20 (2010)
573. J. Yang, W. Wang, P. Wu, Discovering high-order periodic patterns. Knowl. Inf. Syst. **6**, 243–268 (2004)

574. Q. Yang, Intelligent Planning: A Cecomposition and Abstraction Based Approach (Springer, 1997).
575. K. Yip, F. Zhao, Spatial aggregation: theory and applications. J. Artif. Intell. Res. **5**, 1–26 (1996)
576. L. Zadeh, Fuzzy sets. Inf. Contl. **8**, 338–353 (1965)
577. L. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-I. Inf. Sci. **8**, 199–249 (1975)
578. M. Záková, F. Zelezný, Exploiting term, predicate, and feature taxonomies in propositionalization and propositional rule learning. Lect. Notes Comput. Sci. **4701**, 798–805 (2007)
579. S. Zeki, The visual image in mind and brain. Sci. Am. **267**, 68–76 (1992)
580. S. Zeki, *A Vision of the Brain* (Blackwell, Oxford, 1993)
581. S. Zeki, *Inner Vision* (Oxford University Press, Oxford, 1999)
582. S. Zeki, *Splendors and Miseries of the Brain* (Wiley-Blackwell, Oxford, 2009)
583. F. Železný, N. Lavrač, Propositionalization-based relational subgroup discovery with RSD. Mach. Learn. **62**, 33–63 (2006)
584. C. Zeng, S. Arikawa, Applying inverse resolution to EFS language learning. in Proceedings of the International Conference for Young Computer Scientists (Shanghai, China, 1999), pp. 480–487.
585. S. Zhang, X. Ning, X. Zhang, Graph kernels, hierarchical clustering, and network community structure: experiments and comparative analysis. Eur. Phys. J. B **57**, 67–74 (2007)
586. F. Zhou, S. Mahler, H. Toivonen, Review of network abstraction techniques. in Proceedings of the ECML Workshop on Explorative Analytics of Information Networks (Bristol, UK, 2009).
587. S. Zilles, R. Holte, The computational complexity of avoiding spurious states in state space abstraction. Artif. Intell. **174**, 1072–1092 (2010)
588. R. Zimmer, Abstraction in art with implications for perception. Phil. Trans. Roy. Soc. B **358**, 1285–1291 (2003)
589. L. Zuck, A. Pnueli, Model checking and abstraction to the aid of parameterized systems (a survey). Comput. Lang. Syst. Struct. **30**, 139–169 (2004)
590. J.-D. Zucker, A grounded theory of abstraction in artificial intelligence. Phil. Trans. Roy. Soc. Lond. B **358**, 1293–1309 (2003)
591. J.-D. Zucker, J.-G. Ganascia, Selective reformulation of examples in concept learning. in Proceedings of the 11th International Conference on Machine Learning (New Brunswick, USA, 1994), pp. 352–360.
592. J.-D. Zucker, J.-G. Ganascia, Changes of representation for efficient learning in structural domains. in Proceedings of the 13th International Conference on Machine Learning (Bari, Italy, 1996), pp. 543–551.

Index

L. Saitta and J.-D. Zucker, *Abstraction in Artificial Intelligence and Complex Systems*,
DOI: 10.1007/978-1-4614-7052-6, © Springer Science+Business Media New York 2013