PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 42, NO. 1, PP. 147–154 (1998)

A GENERIC MODEL FOR KNOWLEDGE BASES

R.F. ALBRECHT* and Gábor NÉMETH**

* University of Innsbruck
Austria
** Technical University of Budapest
H–1521 Budapest, Hungary

Received: October 30, 1997

Abstract

A knowledge base system is a database system with logical, temporal and topological structures together with operations on these structures. We provide the necessary mathematical concepts for modeling such a system. These are parametrized hierarchical relations, logic functions, hierarchies of variables with their hierarchical control operators, and neighborhood/similarity structures. These concepts are then applied to define a model of a knowledge module. By composition of knowledge modules we obtain the knowledge system model.

Keywords: knowledge bases, system theory.

Introduction

In our model a knowledge base system KBS consists of the following components:

S a set of primitive objects,
$\mathrm{S}(S)$ a hierarchy of relations $\mathbf{S}$, all parametrized (referenced) by indices of hierarchically structured index sets,
$\mathrm{F}(S)$ an explicitly given part $S(\mathbf{S})$, the 'facts',
$\mathrm{D}(S) = \mathrm{def}\, \mathrm{S}(S)\backslash \mathrm{F}(S)$ the implicitly given part of $\mathrm{S}(S)$, obtainable by composite applications of functions of R, a set of (inference, deduction) 'rules', the application of which is in general subject to constraints, conditions, grammatical rules, collected in a set,
$\Gamma(\mathrm{R})$ the grammar of R. The representation of KBS is facilitated by use of variables on sets of components on all hierarchical levels. Assignments to variables and reciprocal, reassignments to substitutable components are performed by
C a hierarchy of control functions and their reciprocals, whereby a control function $\mathrm{val} : P \times \{\mathrm{var}\, x\} \to X$ is associated with each variable $\mathrm{var}\, x$, and where $X = \{x[p] \mid p \in P\}$ is the variability domain ('type') of $\mathrm{var}\, x$ and P is a set of control parameters p. An assignment to $\mathrm{var}\, x$ is then expressed by $\mathrm{val}\,(p, \mathrm{var}\, x) = x[p]$, usually written $\mathrm{var}\, x : (p)x[p]$. Domains of variables can contain variables of lower hierarchical level

and variables can defined on sets of lower level control parameters of variables [ALBRECHT 1995, 1996, 1997].

To operate on the components of KBS, a set of operations

OP has to be given (e.g. selectors like subset forming, projections, cuts, selection of substructures by properties, constructors like set forming, set products, set union, set intersection, concatenation of relations, and transformations of objects and indices, counting cardinalities [ALBRECHT 1995, 1996, 1997]. Again a grammar

Γ(OP) for the application of operations of OP may be given.

To express structural properties of KBS, we need

P a set of predicates, e.g. generalized quantors, 'is part of' property, etc.

Given a partially or linearly ordered logical or physical model time

$(T, <)$ [ALBRECHT 1995, 1996], all components of KBS can be indexed by time points $t \in T$ and processes $(\mathrm{KBS}_t)_{t \in U \subseteq T}$ with varying states $\mathrm{KBS}_{[t]}$ at time points $t \in U \subseteq T$ can be considered. Temporal properties can be adjoined to P.

Finally, on each hierarchical level, sets of objects, rules and parameters of variables can be topologized, mostly by introducing a uniform topological structure. Topological properties can be adjoined to P, for example general distance or similarity measures [ALBRECHT, 1997]. In engineering science topological structures are used under the name 'fuzzy'.

Knowledge Representation

Knowledge we have in mind in form of memorized perceptions, concepts, behavioral processes, intellectual processes, is physically represented by structured physical objects in space-time dimension which we are able to 'interpret'. Mathematically, these objects are abstracted and represented by normed symbols in mathematical space, subject to mathematical operations (aggregations, partitionings, substitutions, combinations, referencing, etc.). The time dimension is mapped onto orderings in space. We use parametrized sets (families, relations) and operations on these for the mathematical description.

The Hierarchy of Parametrized Relations

Let there be given a set S, $S \neq \emptyset$, of elements s, considered primitive with respect to the hierarchy. S is isomorphic with the family $(s)_{s \in S}$ (canonical indexing).

For a given index set $I^{(1)}$, $I^{(1)} \neq \emptyset$, let there be $S^{(1)} \subseteq S \cup \bigcup_{J \subseteq I^{(1)}} S^J$, $S^{(1)} \neq \emptyset$. An element $(s_i)_{i \in J}$ of $S^{(1)}$ is a family or relation on level 1 with 1-dimensional index set J, with ind $J \to S$, $i \to s_{[i]}$, $s_i = \text{def } (i, s_{[i]})$.

For a given index set $I^{(2)}$, $I^{(2)} \neq \emptyset$, let there be $S^{(2)} \subseteq S^{(1)} \cup \bigcup_{J \subseteq I^{(2)}} \times \times (S^{(1)})^J$, $S^{(2)} \neq \emptyset$. An element of $S^{(2)} \backslash S^{(1)} \neq \emptyset$ is a family of families $((s_{i^{(2)} i^{(1)}})_{(i^{(2)}, i^{(1)}) \in J^{(1)}[i^{(2)}]})_{i^{(2)} \in J^{(2)}}$, or a relation on level 2 with two 1-dimensional index sets $J^{(1)} \subseteq I^{(1)}$, $J^{(2)} \subseteq I^{(2)}$, whereby we use the notational conventions $J^{(1)}[i^{(2)}] = \text{def } \{i^{(2)}\} \times J^{(1)}$, $J^{(2)}[i^{(1)}] = \text{def} J^{(1)} \times \{i^{(2)}\}$, $J^{(2,1)} = \{(i^{(2)}, i^{(1)}) \mid i^{(2)} \in J^{(2)} \wedge i^{(1)} \in J^{(1)}[i^{(2)}]\} = \{(i^{(2)}, i^{(1)}) \mid i^{(1)} \in J^{(1)} \wedge i^{(2)} \in J^{(2)}[i^{(1)}]\} \subseteq I^{(2)} \times I^{(1)}$. Then the position of the indices expresses their membership in the hierarchy of index sets.

Concatenation of the family of families yields

$\mathbf{K}((s_{i^{(2)} i^{(1)}})_{(i^{(2)}, i^{(1)}) \in J^{(1)}[i^{(2)}]})_{i^{(2)} \in J^{(2)}} = (s_{i^{(2)} i^{(1)}})_{(i^{(2)}, i^{(1)}) \in J^{(2,1)}}$, a family with 2-dimensional index set.

On the other hand, $(\text{cut } (\{i^{(2)}\}) \ (s_{i^{(2)}, i^{(1)}})_{(i^{(2)}, i^{(1)}) \in J^{(2,1)}}))_{i^{(2)} \in J^{(2)}} = ((s_{i^{(2)} i^{(1)}})_{(i^{(2)}, i^{(1)}) \in J^{(1)}[i^{(2)}]})_{i^{(2)} \in J^{(2)}}$, and for the transposed case, $(\text{cut } (\{i^{(1)}\}) \ (s_{i^{(2)} i^{(1)}})_{(i^{(2)}, i^{(1)}) \in J^{(2,1)}}))_{i^{(1)} \in J^{(1)}} = ((s_{i^{(2)} i^{(1)}})_{i^{(2)}, i^{(1)}) \in J^{(2)}[i^{(1)}]})_{i^{(1)} \in J^{(1)}}$.

Applying induction with respect to $n \in \mathbf{N}$ we have for given $S^{(n)}$, $I^{(n+1)}$, $I^{(n+1)} \neq \emptyset$, $S^{(n+1)} \subseteq S^{(n)} \cup \bigcup_{J \subseteq I^{(n+1)}} (S^{(n)})^J$ in general on hierarchical level $n+1$. An element of highest hierarchical level of $S^{(n+1)}$ is then of the form $(\ldots((s_{i^{(n+1)} \ldots i^{(1)}})_{i^{(n+1)} \ldots i^{(1)} \in J^{(1)}[i^{(n+1)}, \ldots i^{(2)}]})_{i^{(n+1)} \ldots i^{(2)} \in J^{(2)}[i^{(n+1)}, \ldots i^{(3)}]}) \ldots \ldots)_{i^{(n+1)} \in J^{n+(1)}}$, or if concatenated $(s_{i^{(n+1)} \ldots i^{(1)}})_{i^{(n+1)} \ldots i^{(1)} \in J^{(n+1, \ldots 1)}}$ with $J^{(n+1, \ldots 1)} \subseteq J^{(n+1)} \times \ldots \leq J^{(1)}$. The structural complexity of the objects is mirrored in the structural complexity of the indices after concatenation.

EXAMPLE 1 Construction of valuated objects, especially logics [see ALBRECHT, 1997]: We assume $S = A \cup V$ is a partition, $I^{(1)} = (\{1, 2\}, <)$, $S^{(1)} \subseteq A \times V$ with elements (a, v), the indices suppressed, $I^{(2)}$ a finite set, $S^{(2)} \subseteq \prod_{J \subseteq I^{(2)}} (S^{(1)})^J$ with elements $(a_i, v_i)_{i \in J}$, $I^{(3)} = (\{1, 2\}, <)$, $S^{(3)} \subseteq S^{(2)} \times V$ with elements $((a_i, v_i)_{i \in J}, v)$, whereby $v = \varphi_{\text{card } J}((v_{[i]})_{i \in J})$, $\varphi_{\text{card } J} : V^{\text{card } J} \to V$, which is in particular a logic function for V a lattice. In this example $\prod$, $\times$, $\varphi_{\text{card } J}$ are elements of OP, the applications of $\prod$, $\times$ are restricted, the restrictions are elements of $\Gamma(\text{OP})$.

Rules

At logical time t let there be given a part $D \subseteq \mathrm{F}$. A rule $f \in \mathrm{R}$ is then a surjective function $f : D \to W$ with $d \mapsto w = f(d)$. If the grammar $\Gamma(\text{OP})$ admits for F and w a concatenation, then at logical time t', $t < t'$, $\mathrm{D}' = \text{def } \mathbf{K}((F, w), E(C), C)$ (for concatenations I refer to [ALBRECHT, 1995]). If $i(w)$ is the (composite) index of w, then $w = \text{def } pr(i(w))\mathrm{D}'$. Rule applications can be composed. We distinguish rule applications from operations op $\in$ OP. However, both can be combined.

EXAMPLE 2 ('formal languages'): $K = \{1,2,3,4\}$, $D = \{(a_{ik})_{k \in K} \mid i \in I \wedge \bigwedge i \in I(a_{[i]3} = a_3 \wedge \bigwedge k \in K(a_{[ik]} \in A))\}$, A a given set, 'production rule' $f(a_3) = (b_{31}b_{32})$, $\mathbf{K}(pr(\{i1, i2, i4\})D$, $\{(b_{31}, b_{32})\} = \{(a_{i1}, a_{i2}, b_{31}, b_{32}, a_{i4})\}$ (concatenation with replacement, $a_3 ::= (b_{31}, b_{32})$, in context $\bigwedge i \in (a_{[i1]}, a_{[i2]}, a_{[i4]} \in A))$.

EXAMPLE 3 ('reasoning' inference rules): $K = \{1,2,3,4\}$, $V = \{'t', 'f'\}$, $D = \{((a_k, v_k)_{k \in K}, v = \varphi_4((v_{[k]})_{k \in K}) \mid v \in V \wedge (v_{[k]})_{k \in K} \in V^4\}$, inference rule $f : D \to \{((b_1, w_1)_{l \in \{1,2\}}, w = \psi_2((w_{[1]})_{l \in \{1,2\}}) \mid w \in V \wedge (w_{[1]})_{l \in \{1,2\}} \in V^2\}$, φ_4, ψ_2 logic functions, such that $((a_k, v_k)_{k \in K}, \varphi_4((v_{[k]})_{k \in K}) = 't'/'f')$ $\mapsto ((b_1, w_1)_{l \in \{1,2\}}, \psi_2((w_{[1]})_{l \in \{1,2\}}) = 't'/'f')$. An example in usual notation is: 'if' $((a_{[1]} \wedge a_{[2]} \wedge a_{[3]}) \vee a_{[4]})$ 'then' $(b_{[1]} \vee b_{[2]})$ 'else' $\neg(b_{[1]} \vee b_{[2]})$.

Utilisation of Variables

We can represent a KBS by a hierarchy of variables and their control functions/operators: Starting on 'top', we consider var KBS $=$ (var S, var S (var S), var F(var S), var D (var F), var R (var F), var Γ (var R), var OP, var Γ (var OP), var P, var $(T, <)$). All variables var X range on given domains X parametrized by $P_{[x]}$ and have control functions val: $P_{[x]} \to X$ with control parameters $p_{[x]} \in P_{[x]}$.

The assignment steps in logical time are:

var $S := S \neq \emptyset$, selection of the primitive objects: var $I := I \neq \emptyset$, selection of the primitive indices;

var OP := OP, var Γ(var OP) := Γ(OP), selection of admitted structors for S;

var P := P, selection of structural predicates; for bottom up construction of the hierarchy F up to F(N);

var $F^{(0)}$: pow $S \backslash \emptyset$, selection of var $F^{(0)} := F^{(0)}$; $\mathrm{F}(0) = \emptyset$;

var N : **N**. var $N := N$, for $n = 0, 1, 2, \ldots N - 1$;

var $I^{(n+1)}$: pow $I \backslash \emptyset$, selection of var $I^{(n+1)} := I^{(n+1)}$;

var $F^{(n+1)}$: pow $\bigcup_{J \subseteq \mathrm{var}\, I^{(n+1)}} (F^{(n)})^J \backslash \emptyset$, selection of var $F^{(n+1)} := F^{(n+1)}$;

var $\mathrm{F}(n+1) =: \mathrm{F}(n) \cup F^{(n+1)}$;

var R(F(N)) := R(F(N)), var Γ(R(F(N))) := R(F(N)), selection of admitted rules.

This results in var KBS := KBS. We suppressed the assignment parameters. Assignments to composite variables can be performed in partial steps [see e.g. ALBRECHT 1997]. Analogously, assignments to time variables and topological structure variables can be made.

The deduction steps in logical time are:

$\text{var}\, f(\text{var}\, D)$: R with $\Gamma(\text{R})$. Selection of f : $\text{var}\, f := f$, follows $\text{var}\, D := D$, $\text{var}\, W := f(D)$. Selection of an argument: $\text{var}\, d : D$, $\text{var}\, d := d$, evaluation of $\text{var}\, w := w = f(d)$.

Decision on operation on (F, w) : $\text{var}\, op(\text{F}, w)$: OP with $\Gamma(\text{OP})$, $\text{var}\, op(\text{F}, w) := op(\text{F}, w)$.

So far we made the assumption, that 'someone' made the selection of all the control parameters $p_{[x]}$ involved, whereby the sets $P_{[x]}$ of control parameters are in hierarchical dependence. Parametrizing these sets by higher order parameters $Q_{[x]}$, we can define variables on the lower order parameter sets $P_{[x]}$: $\text{var}\, p_{[x]} : \{p_{[q \cdot x]} \mid p_{[q \cdot x]} \in P_{[x]} \wedge q \in Q_{[x]}\}$ and control functions $\text{val} : Q_{[x]} \times \{\text{var}\, p_{[x]}\} \to P_{[x]}$. These higher order control functions can depend on the results of previous lower order assignments ('feedback') and on currently given external parameters ('goals') and are assumed to represent 'higher intelligence' for forthcoming decisions. The hierarchy of higher order control functions can be extended. More details are given in [ALBRECHT, 1997].

Binary Knowledge Modules

As a particular but important case we consider knowledge represented by valuated binary relations. For example, if $y = f(x)$, (y, x) is a pair, if (y, x) is a proposition (object y has property x), it can be valuated for example by $v \in V = \{\text{'}t\text{'}, \text{'}f\text{'}\}$ to give $((y, x), v)$. This includes of course composite objects (relations) y and composite properties (relations) x and arbitrary sets V with any structures.

Deterministic Case with Discrete Topology

Given a non-empty set Y of elements y named 'objects' and a non-empty set X of elements x named 'properties', bijective parametrizations $\text{ind} : J \leftrightarrow Y$, $\text{ind}' : I \leftrightarrow X$, and a relation $R = (y_j, x_i)_{(j,i) \in U}$, $U \subseteq J \times I$, with $\text{pr}_1 U = J$, $\text{pr}_2 U = I$. As well we could have named X the set of objects and Y the set of properties. If V is a non-empty set and $\varphi : R \to V$ is a valuation, then R can be represented by $M = \text{def}\, (v_{ji})_{(j,i) \in U}$ with $v_{ji} = \text{def}\, \varphi((y_j, x_i))$. We consider $\wedge j \in J(\text{cut}\ (\{j\})\ M = (v_{ji})_{i \in I_{[j]}})$, $\wedge i \in I(\text{cut}\ \{i\})\ M = (v_{ji})_{j \in J_{[i]}})$, which define $I_{[j]}$, $J_{[i]}$, and we assume $\wedge j, j' \in J(j \neq j' \Rightarrow (v_{ji})_{i \in I_{[j]}} \neq (v_{j'i})_{i \in I_{[j']}})$, $\wedge i, i' \in I(i \neq i' \Rightarrow (v_{ji})_{j \in J_{[i]}} \neq (v_{ji'})_{j \in J_{[i']}})$. We name $((y_j, x_i), v_{ji})_{ji \in U}$ a 'knowledge module' KM.

Let there be given $\eta : V \times V \to \mathbf{B} = (\{\text{'}t\text{'}, \text{'}f\text{'}\}, \sqcap, \sqcup)$ with $\eta(\text{diag}\, V \times V) = \{\text{'}t\text{'}\}$, $\eta(V \times V \setminus \text{diag}\, V \times V) = \{\text{'}f\text{'}\}$. For $\text{card}\, K \leq \text{card}\, U$ we consider $\varphi_{\text{card}\, K} \in (\mathbf{B}^{\text{card}\, K} \to \mathbf{B})$. For $\tilde{I} \subseteq I$ and $\tilde{J} \subseteq J$ we define $J[\tilde{I}] =$

def $\bigcap_{i\in\tilde{I}} J_{[i]} = \{j \mid j \in J \wedge \tilde{I} \subseteq I_{[j]}\}$, $I[\tilde{J}] =$ def $\bigcap_{j\in\tilde{J}} I_{[j]} = \{i \mid i \in I \wedge \tilde{J} \subseteq J_{[i]}\}$, and for $\tilde{v}_{[ji]} \in V$, $\beta \in \mathbf{B}$, $Y((\tilde{v}_i)_{i\in\tilde{I}}, \varphi_{\mathrm{card}\,\tilde{I}}, \beta) =$ def $\{y_j \mid j \in J[\tilde{I}] \wedge \varphi_{\mathrm{card}\,\tilde{I}}((\eta(\tilde{v}_{[i]}, v_{[ji]}))_{i\in\tilde{I}}) = \beta\}$,

$$X((\tilde{v}_j)_{j\in\tilde{J}}, \varphi_{\mathrm{card}\,\tilde{J}}, \beta) = \mathrm{def}\ \{x_i \mid i \in I[\tilde{J}] \wedge \varphi_{\mathrm{card}\,\tilde{J}}((\eta(\tilde{v}_{[j]}, v_{[ji]}))_{j\in\tilde{J}}) = \beta\}\ .$$

We have

$$Y((\tilde{v}_i)_{i\in\tilde{I}}, \varphi_{\mathrm{card}\,\tilde{I}}, `t\text{'}) = \Phi_{\mathrm{card}\,\tilde{I}}((Y(\tilde{v}_i, `t\text{'}))_{i\in\tilde{I}})\ ,$$

$$Y(\tilde{v}_i, `t\text{'}) = \mathrm{def} Y((\tilde{v})_i)_{i\in\{i\}}, \eta(\tilde{v}_{[i]}, v_{[ji]}) = `t\text{'})\ ,$$

$$X((\tilde{v}_j)_{j\in\tilde{J}}, \varphi_{\mathrm{card}\,\tilde{J}}, `t\text{'}) = \Phi_{\mathrm{card}\,\tilde{J}}((X(\tilde{v}_j, `t\text{'}))_{j\in\tilde{J}})\ ,$$

$$X(\tilde{v}_j, `t\text{'}) = \mathrm{def} X((\tilde{v})_j)_{j\in\{j\}}, \eta(\tilde{v}_{[j]}, v_{[ji]}) = `t\text{'})\ ,$$

$\Phi_{\mathrm{card}\,\tilde{I}}$, $\Phi_{\mathrm{card}\,\tilde{J}}$ the set functions corresponding to the boolean functions $\varphi_{\mathrm{card}\tilde{I}}$, $\varphi_{\mathrm{card}\tilde{J}}$, respectively. Further,

$$\wedge j \in J(\mathrm{Y}_j = \mathrm{def}\{Y((\tilde{v}_{[j]i})_{i\in\tilde{I}}, \sqcap_{\mathrm{card}\,\tilde{I}}, `t\text{'}) \mid \tilde{I} \in (\mathrm{pow}\, I_{[j]})\backslash\emptyset\}\ ,$$

$$\text{is a filter base for } \tilde{I} \to I_{[j]})\ ,$$

$$\wedge i \in I(\mathrm{X}_i = \mathrm{def}\{X((\tilde{v}_{[j]i})_{j\in\tilde{J}}, \sqcap_{\mathrm{card}\,\tilde{J}}, `t\text{'}) \mid \tilde{J} \in (\mathrm{pow}\, J_{[i]})\backslash\emptyset\}\ ,$$

$$\text{is a filter base for } \tilde{J} \to J_{[i]})\ .$$

This expresses the 'inheritance' principle: the larger the set of common properties/objects, the smaller the set of objects/properties possessing these properties/objects. If $\wedge j \in J(\lim \mathrm{Y}_j = \{y_j\})$ and $\wedge i \in I(\lim \mathrm{X}_i = \{x_i\})$, then we say $(v_{ji})_{i\in I_{[j]}}$ and $(v_{ji})_{j\in J_{[i]}}$ 'characterize' y_j and x_i, respectively. Under this assumption, there may exist 'coarser' filter bases Y_j^* and X_i^* also converging to $\{y_j\}$ and $\{x_i\}$, respectively [see e.g. ALBRECHT 1994]. It can be of practical importance to find such Y_j^* and X_i^* of maximal coarseness (minimal characterizations). For all $I[j]^*$ being characterizations and for a given $\{y_j \mid j \in \hat{J}\}$ we have

$$\{y_j \mid j \in \hat{J}\} = Y(((v_{[ji]})_{i\in I[j]^*})_{j\in\hat{J}}, \sqcup_{\mathrm{card}\,\hat{J}}(\sqcap_{\mathrm{card}\,I[j]^*}), `t\text{'}) =$$

$$= \bigcup_{j\in\hat{J}} \bigcap_{i\in I^*[j]} Y(v_{[ji]}, `t\text{'})\ ,$$

and an analogue result for the transposed equation. Considering the dual

$$\wedge j \in J(\mathrm{Y}_j = \mathrm{def}\{Y((\tilde{v}_{[j]i})_{i\in\tilde{I}}, \sqcup_{\mathrm{card}\,\tilde{I}}, `t\text{'}) \mid \tilde{I} \in (\mathrm{pow}\, I_{[j]})\backslash\emptyset\}\ ,$$

$$\text{is an ideal base for } \tilde{I} \to I_{[j]})\ ,$$

$$\wedge i \in I(\mathrm{X}_i = \mathrm{def}\{X((\tilde{v}_{[j]i})_{j\in\tilde{J}}, \sqcup_{\mathrm{card}\,\tilde{J}}, `t') \mid \tilde{J} \in (\mathrm{pow}\, J_{[i]})\backslash\emptyset\}\,,$$
$$\text{is an ideal base for } \tilde{J} \to J_{[i]})\,,$$

we have the dual to the inheritance principle: the larger the set of alternative properties/objects, the larger the set of objects/properties possessing these properties/objects.

'Queries' to the knowledge module KM are then formulated by applications of operations of OP. For example:

1. given: $\emptyset \subset K \subseteq U$, $(\tilde{V}_{ji})_{ji\in K}$, $\emptyset \subset \tilde{V}_{[ji]} \subseteq V$; find: cut $((\tilde{V}_{ji})_{ji\in K})M$, find card cut $((\tilde{V}_{ji})_{ji\in K})M$, find
$$\{v_{j'i'} \mid v_{j'i'} \in M \wedge \vee v_{ji} \in \mathrm{cut}((\tilde{v}_{ji})_{ji\in K})M(v_{[j'i']} = v_{[ji]})\}\,.$$
2. given: $\wedge(j,i) \in K(V_{[ji]} \subseteq V)$, $\varphi_{\mathrm{card}\,K}, \beta$, find: $\wedge(j,i) \in K \wedge \tilde{v}_{ji} \in V_{ji}(R((\tilde{v}_{ji})_{ji\in K}, \varphi_{\mathrm{card}\,K}, \beta)$, with
$$R((\tilde{v}_{ji})_{ji\in K}, \varphi_{\mathrm{card}\,K}, \beta) = \mathrm{def}\ \{v_{ji} \mid (j,i) \in$$
$$K \wedge \varphi_{\mathrm{card}\,\tilde{J}}((\eta(\tilde{v}_{[ji]}, v_{[ji]}))_{ji\in K})) = \beta\}\,.$$
3. given: $(\tilde{v}_i)_{i\in\tilde{I}}, \varphi_{\mathrm{card}\,\tilde{I}}, \beta$, find $(y_j)_{j\in J^*} = \mathrm{def} Y((\tilde{v}_i)_{i\in\tilde{I}}, \varphi_{\mathrm{card}\,\tilde{I}}, \beta)$; given $(y_j)_{j\in J^*}$: find (all) $(\tilde{v}_i)_{i\in\tilde{I}}, \varphi_{\mathrm{card}\,\tilde{I}}, \beta)$ such that
$$(y_j)_{j\in J^*} = Y((\tilde{v}_i)_{i\in\tilde{I}}, \varphi_{\mathrm{card}\,\tilde{I}}, \beta)\,.$$
4. For $\emptyset \subset \tilde{I} \subset I_{[j]}$ let be $(y_{j'})_{j'\in J^*} = Y((v_{[j]i})_{i\in\tilde{I}}, \sqcap_{\mathrm{card}\,\tilde{I}}; \beta)$. This defines deduction rules, implicitly given by M:
$$(x_i, v_{[j]i})_{i\in\tilde{I}} \Rightarrow (y_{j'})_{j'\in J^*} \Rightarrow ((x_i, v_{ji})_{i\in I^*(j)})_{j\in J^*} = (x_i, (v_{ji})_{j\in J^*(i)})_{i\in I^*}$$
with
$$I^*(j) = I_{[j]}\backslash\tilde{I}\ I^* = \bigcup_{j\in J^*} I^*(j)\,,\quad J^*(i) = \{j \mid j \in J^* \wedge i \in I^*(j)\}\,.$$
The conclusion can be repeated for $(x_i, (v_{ji})_{j\in J^*(i)})_{i\in I^*}$.

Deterministic Case with General Topologies

We assume that $(V, \leq, \sqcup_v, \sqcap_v)$ is a complete atomic boolean lattice. Then the families $(x_i, v_{[j]i})_{i\in I[j]}$ define functions $f_{[j]} : I_{[j]} \to V$. The set extensions of the $f_{[j]}$ are homomorphisms, i.e. for $I' \subseteq I'' \subseteq I_{[j]}$ holds $f_{[j]}(I') \leq f_{[j]}(I'')$. Then a filter/ideal base $\mathrm{I} = \{I_{[jk]} \mid k \in K\}$ on pow $I_{[j]}$ maps onto a filter/ideal base $\mathrm{V} = \{f_{[j]}(I_{[jk]}) \mid k \in K\}$ on pow V. Filter/ideal bases can express neighbourhood/similarity relations between objects y and between properties x. To measure neighbourhood and similarity of values $v \in V$ we introduce a uniform topological structure on V by a filter base $\mathrm{B} = \{D_{[q]} \mid q \in Q\}$ on pow $(V \times V)$ with $V \times V \in \mathrm{B}$, $\mathrm{diag}(V \times V) \subseteq \bigcap_{q \in Q} D_{[q]}$, and $D_{[q]}^{-1} = D_{[q]}$ and assume B is itself a complete lattice. Then for any pair $(\tilde{v}, v) \in V \times V$ uniform, generalized, multivalued distances $d_{\cup}(\tilde{v}, v)$ and $d_{\cap}(\tilde{v}, v) \in \mathrm{B}$ with $d_{\cup}(\tilde{v}, v) \subset d_{\cap}(\tilde{v}, v)$ can be introduced [see ALBRECHT, 1997]. We set $\eta(\tilde{v}, v) = d_{\cap}(\tilde{v}, v)$ and have B (or any isomorphic complete lattice) as generalization of $\mathbf{B} = \{‘t’, ‘f’\}$, the latter corresponding to $\mathrm{B} = \{V \times V, \mathrm{diag}(V \times V)\}$. This makes it possible to measure the distance of any $(v_{ji})_{ji \in K} = pr(K)M$ from a given $(\tilde{v}_{ji})_{ji \in K}$ by $(\beta_{[ji]} = \mathrm{def}\ \eta(\tilde{v}_{[ji]}, v_{[ji]}))_{ji \in K}$ and to valuate the $v_{ji} : (v_{ji}, \beta_{ji})_{ji \in K}$. We then can apply a logic function $\varphi_{\mathrm{card}\,K} \in \Phi_{\mathrm{card}\,K} = (\mathrm{B}^{\mathrm{card}\,K} \to \mathrm{B})$ for a valuation $((v_{ji}, \beta_{ji})_{ji \in K}, \varphi_{\mathrm{card}\,K}$ $((\beta_{[ji]})_{ji \in K})$ and can formally proceed as in the boolean case before.

Knowledge Modules with Variables

If the knowledge module contains variables, e.g. $((y_j, x_i), \mathrm{var}\, v_{ji})$, they express indeterminacy in the sense that the domain (‘type’) of the variable is known but the value to be assigned is not yet determined. This case has to be distinguished from elements not appearing in the module, e.g. index pairs $(j', i') \in (J \times I) \backslash U$. ‘Queries’ with variables to a module with variables in general result in ‘answers’ with variables.

Composition of Knowledge Modules

A knowledge module can be seen as an input/output system and hence modules can be composed to a knowledge base system by feeding (part of) the answer of one module as (part of a) query to the same or another module. This composition is analogue to the composition of functional modules in computer architecture.

References

[1] ALBRECHT, R. F. (1994): Some Basic Concepts of Objectoriented Databases, *System Science*, Vol. 20/1, Wroclaw, pp. 17–30.

[2] ALBRECHT, R. F. (1995): On the Structure of Discrete Systems, *Lect. Notes in Comp. Sc.*, Vol. 1030, *Comp. Aided Systems Theory*, Springer, pp. 3–18.

[3] ALBRECHT, R. F. (1996): The Structure of Discrete Systems, Trends in Theoretical Informatics, (eds. R.F. Albrecht, H. Herre), Österreichische Computer Gesellschaft, pp. 127–144.

[4] ALBRECHT, R. F. (1997): Systems with Topological Structures, submitted to *Int. Conf. on Computing Anticipatry Systems, CASYS97*, Liège.