Structures with many−valued information and their relational proof theory

Ivo Düntsch
School of Information and Software Engineering
University of Ulster at Jordanstown
Newtownabbey, BT 37 0QB, N. Ireland
I.Duentsch@ulst.ac.uk

Wendy MacCaull
Department of Mathematics, Statistics and Computer Science
St. Francis Xavier University
P.O.Box 5000, Antigonish, NS, B2G 2W5, Canada
wmaccaul@stfx.ca

Ewa Orlowska
Institute of Telecommunications
Szachowa 1, 04–894 Warsaw, Poland
orlowska@itl.waw.pl

ABSTRACT

We present a uniform relational framework for developing proof systems for theories of manyvaluedness that may have the form of a logical system, of a class of algebra or of an information system. We outline a construction of proof systems for SH_n logics, mv−algebras and many−valued information systems.

1. Introduction

The structures that deal with multiple−valued information are of various kinds, in particular they may be:
(I) Logical systems: e.g. ukasiewicz logics (ukasiewicz 1920), Post logics (Post (1920, 1921), symmetric Heyting logics (Iturrioz 1983), 4−valued Belnap logic (Belnap 1977), etc.
(II) Algebraic structures: most often they are the counterparts to logical systems, e.g. mv−algebras − the counterpart to the infinite−valued ukasiewicz logic (Chang 1958), Post algebras (Epstein (1960), SH_n algebras (Iturrioz 1983), etc.
(III) Structures that arise in connection with knowledge representation: e.g. many−valued information systems (Düntsch et al.).

These structures usually consist of the two parts:
− A static part for the representation of information

− A dynamic part for processing of information.

We show that one formalism, that of algebras of relations, can be uniformly applied to both the representation of and the processing of information for structures from all the groups listed above. We present relational formalisms for three structures, each of which is a representative example of one of the above classes: SH_n logics (group I), mv−algebras (group II), many−valued information systems (group III).

2. The basic relational formalism

Basic relational structures are the algebras Re(U) of binary relations on a set U, with the operations of union ($\cup$), intersection ($\cap$), complement ($-$), composition (;) and converse ($^{-1}$) of relations and with two constants 1 and 1', interpreted as the universal relation $U\times U$, and the identity relation $\{(x,x): x\in U\}$, respectively. We recall that composition and converse of relations are defined as follows:

$$P;R=\{(x,y)\in U\times U: \text{there is } z\in U \text{ such that } (x,z)\in P \text{ and } (z,y)\in R\}$$

$$R^{-1}=\{(x,y)\in U\times U: (y,x)\in R\}.$$

The formal system associated with those structures consists of a set of expressions and a set of inference rules. The expressions are of the form xRy, where R is a term built from relation variables and/or constants with the relational operations and x, y are variables representing elements of U. The expression xRy is satisfied by the elements a, b of U if $(a,b)\in R$. xRy is true in an algebra $\mathbb{A}$ of relations whenever it is satisfied by any a, $b\in U$, and xRy is valid in the theory of algebras of relations whenever it is true in every algebra of relations. It follows that xRy is true in an algebra $\mathbb{A}$ iff the equation R=1 is true in the equational formalism of algebras of relations.

The rules of the formal system of algebras of relations are of the two kinds. The first group consists of the rules that reflect definitions of the relational operations and the rules of the second group reflect properties of relational constants. The rules apply to finite sequences of relational expressions. An application of a rule results in a new sequence of expressions or a pair of sequences (branching is denoted by |). In the rules listed below K and H denote finite (possibly empty) sequences of relational expressions. A variable is said be *restricted* in a rule if it does not appear in any expression of the upper part of the rule.

(RO) **Rules for relational operations**

($\cup$) K, $xP\cup Ry$, H

K, xPy, xRy, H

($-\cup$) K, $x-(P\cup R)y$, H

K, $x-Py$, H | K, $x-Ry$, H

($\cap$) K, $xP\cap Ry$, H

($-\cap$) K, $x-(P\cap R)y$, H

K, xPy, H | K, xRy, H   K, x−Py, x−Ry, H

(−−) K, x−−Ry, H

K, xRy, H

($^{-1}$) K, xR^{-1}y, H

K, yRx, H

(−$^{-1}$) K, x−(R^{-1})y, H

K, y−Rx, H

(;) K, xP;Ry, H

K, xPz, H, xP;Ry | K, zRy, H, xP;Ry

where z is any variable

(−;) K, x−(P;R)y, H

K, x−Pz, z−Ry, H

where z is a restricted variable

(RC) **Rules for relational constants**

($1'_1$) K, xRy, H

K, x1’z, H, xRy | K, zRy, H, xRy

where z is any variable,

($1'_2$) K, xRy, H

K, xRz, H, xRy | K, z1’y, H, xRy

where z is any variable

($1'_3$) K, x1’y, H

K, y1’x, H, x1’y

(AX) **Axiomatic sequences**

A sequence of formulas is said to be *axiomatic* whenever it contains a subsequence of either of the following forms:

(a1) xRy, x−Ry

(a2) x1y

(a3) x1’x

A sequence K of relational expressions is *valid* iff for every algebra of relations $\mathbb{A}$=Re(U) and for any elements a, b of U there is an expression xRy in K which is satisfied by a, b. It follows that sequences of formulas are interpreted as metalevel disjunctions of their elements. A rule of the form $K/\{H_t: t\in T\}$ is said to be *admissible* whenever sequence K is valid iff for all $t\in T$, sequence H_t is valid. Hence, branching is interpreted as a metalevel conjunction. It is easy to see that the axiomatic sequences are valid and all the rules given above are admissible. Observe that the admissibility of the rules from group (RC) for any relational constant is a consequence of the properties of constant 1'. Namely, we have:

Rule ($1'_1$) is admissible iff $1';R\subseteq R$; rule ($1'_2$) is admissible iff $R;1'\subseteq R$; rule ($1'_3$) is admissible iff 1' is symmetric. Transitivity of 1' is reflected by an instance of rule ($1'_1$) with R=1'. Reflexivity of 1' is reflected by validity of the axiomatic sequence (a3).

Given an expression xRy, successive application of the rules results in a tree whose root is xRy and whose nodes are labelled with finite sequences of relational expressions. We stop applying rules to the expressions in a node if it is labelled with an axiomatic sequence. If the tree obtained in this way is finite and all of its leaves are labelled with axiomatic sequences, then xRy is valid. This follows easily from the admissibility of the rules and validity of axiomatic sequences. It is also known that the converse holds: namely, if xRy is valid, then there is a finite tree obtained from xRy in the way described above such that all of its leaves are labelled with the axiomatic sequences. Hence, the basic relational formalism is complete. A detailed presentation of this formalism can be found in (Orlowska 1988, 1996). Details of a working prototype implementation ReVAT (Relational Validator via Semantic Tableaux) may be found in (Spencer and MacCaull 2000). The basic relational formalism in undecidable. Several of its decidable subclasses are presented in (Buszkowski and Orlowska 1997) and (Dobrowolska−Buffoli 1998).

Sometimes relational formalisms require a cut rule in order to get completeness result. The cut rule has the following form:

(cut) $\dfrac{K}{K,\ xRy \mid K,\ x{-}Ry}$

where R is any relational expression appearing in the sequence K and x,y are any variables

Clearly, if a relational formalism requires n−ary relations for n≥3, the rules ($\cup$), ($-\cup$),($\cap$),($-\cap$), ($--$), (cut) are of the form presented above with the respective relational expressions denoting n−ary relations.

3. Relational proof theory

In this section we outline general principles of construction of relational proof systems for theories which might be presented in a variety of forms. The principles provide a methodology for developing proof theory depending on a presentation of the theory in question.

Case 1

Let us assume that a theory is given in a form of a class of algebras of relations. The algebras of such a class are usually extensions of (a reduct of) the basic algebras Rel(U) of relations. That is, besides the standard relational operations and constants, some new relational operations and/or constants satisfying some constraints expressible in a first order language might be added. In that case we would extend the language of the basic relational formalism with some relational operations and/or constants specific for the given class of algebras. Next, we would extend the set of rules and/or axiomatic sequences of the basic relational formalism with rules and/or axiomatic sequences that characterise the new operations or constants. This means that each constraint postulated for the given class of algebras should have associated with it some rule(s) and/or axiomatic sequence(s) in such a way that the following property holds.

Constraint−rule and constraint−axiomatic sequence adequacy:
The rule is admissible and/or the axiomatic sequence is valid iff the respective constraint holds.

An example of the relational proof theory for a class of algebras of relations can be found in (Frias and Orlowska 1995, Buszkowski and Orlowska 1997, Düntsch and Orlowska 1999).

Below we list some rule generation principles that lead from a constraint to an adequate rule. For the sake of simplicity the principles are formulated for a formalism with binary relations, but the analogous principles are valid for any n−ary relations, $n \geq 3$.

By a *relational literal* we mean a relation variable and/or constant or a complemented relation variable and/or constant. For the sake of simplicity, any expression xRy, such that R is a relational literal, is also referred to as a literal. If a literal A is xRy, where R is a relational literal, then we denote the literal x−Ry by $\neg A$.

Rule generation principles

Principle I

A constraint is given in the form of a formula:

(c1) $\forall x_1,\ldots,x_n(A \rightarrow B)$

where A is a conjunction of literals and B is a literal.

Let $A = A_1 \wedge \ldots \wedge A_n$, $n \geq 1$.

Then the adequate rule is:

$$\text{(rc1)} \quad \frac{K,\ B,\ H}{K,\ A_1,\ H,\ B \ |\ldots|\ K,\ A_n,\ H,\ B}$$

Principle II

A constraint is given in the form of a formula:

(c2) $\forall x_1,\ldots,x_n(A \rightarrow \exists z_1,\ldots,z_m B)$

where A is a conjunction of literals, variables $z_1,...,z_m$ do not occur in A, B is a conjunction of disjunctions of literals and variables occurring in B are among $x_1,....,x_n$, $z_1,...,z_m$.

Let $A=A_1\wedge...\wedge A_n$, $n\geq 1$, $B=B_1\wedge...\wedge B_m$, $m\geq 1$, and $B_i=B_{i,1}\vee...\vee B_{i,k(i)}$, $i=1,...m$, $k(i)\geq 1$.

Then the adequate rule (an m−fold branching rule) is:

(rc2) $\quad K, \neg A_1,...,\neg A_n, H$

$K, \neg B_{1,1},...,\neg B_{1,k(1)}, H, \neg A_1,...,\neg A_n$ |...| $K, \neg B_{m,1}...,\neg B_{m,k(m)}, H, \neg A_1,...,\neg A_n$

where any variable z_j, $j=1,...,m$, occurring in any $B_{s,k(s)}$, $s=1,...m$, is a restricted variable.

Principle III

A particular instance of II is when A is the empty conjunction of literals. The constraint has the form:

(c3) $\forall x_1,....,x_n \exists z_1,...,z_m B$

with B as above, and the adequate rule is:

(rc3) $\quad K$

$K, \neg B_{1,1},...,\neg B_{1,k(1)}, H$ |...| $K, \neg B_{m,1},...,\neg B_{m,k(m)}, H$

where $x_1,....,x_n$ are any variables and $z_1,...,z_m$ are restricted variables.

The formalisms that include the rules defined according to the principles II and III require a cut rule in order to get the completeness result. A discussion of when the cut rule may be avoided for relational proof theory can be found in (MacCaull 2000). In section 6 we follow the methodology outlined in Case 1 and present a relational proof theory for many−valued information systems.

Case 2

If a theory is given in a form of a class, say C, of algebras that are not necessarily algebras of relations, then the first step is to define a class ReC of relational structures for C such that C is adequately simulated in ReC in the following sense.

Simulation of an algebra in the associated relational structure:

There is a translation mapping t from the first order language L(C) of C into the first order language L(ReC) of ReC such that for any formula F of L(C), F is true in every algebra of C iff t(F) is true in every structure from ReC.

Then the second step is to develop a relational proof theory for the class ReC of structures.

In section 5 we present a relational proof system for the class of mv−algebras constructed with the method outlined in Case 2. Examples of ’relationalisation’ of algebraic structures and relational proof systems for such structures can be found in (MacCaull 1997, 1998, 1998a) and (Düntsch, I., Orlowska, E. and Hui Wang 2000).

Case 3

A theory is presented in the form of a class of logical systems. Semantic structures of the logics are usually either classes of algebras or Kripke−style structures. In both cases, the first step towards construction of a relational proof system is a development of a relational semantics for the logics. We proceed as in Case 2; that is, we assign relations to operations from the respective algebras or Kripke structures. Next, we define a relational formalism ReL. We then prove its adequacy for the logic L in the following sense.

Interpretability of a logic in the associated relational formalism:
There is a translation mapping t from the language of logic L into the language of ReL such that for any formula A of L, A is valid in L iff t(A) is valid in ReL.

In section 4 we follow the methodology outlined in Case 3 and present a relational proof theory for the class of SH_n logics, $n\geq 1$.

4. Relational SH_n formalism

SH_n logics, $n\geq 1$, were introduced in (Iturrioz 1983) in connection with a search for a lattice−based formulation of ukasiewicz logics $_n$. The formulas of these logics are built from propositional variables with the operations of disjunction ($\vee$), conjunction ($\wedge$), implication ($\rightarrow$), two negations ($\neg$, $-$), and the family $\{S_i: i=1,...,n-1\}$ of unary operations. Semantic structures for the logics are of the form:

$K=(W, \leq, \{s_i: i=1,...,n-1\}, g\}$

where W is a nonempty set, $\leq$ is a binary relation on W, s_i and g are functions on W, and the following conditions are satisfied:

(K1) $\leq$ is reflexive and transitive

(K2) $x\leq y$ implies $g(y) \leq g(x)$

(K3) $gs_i(x)=s_{n-i}g(x)$, for all $i=1,...,n-1$

(K4) $gg(x)=x$

(K5) $s_js_i(x)=s_j(x)$

(K6) $s_1(x) \leq x$

(K7) $x\leq s_{n-1}(x)$

(K8) $s_i(x) \leq s_j(x)$ for $i\leq j$

(K9) $x\leq y$ implies $s_i(x) \leq s_i(y)$ and $s_i(y) \leq s_i(x)$, for all $i=1,...,n-1$

(K10) $s_i(x)\leq x$ and $x\leq s_i(x)$ imply $s_i(x)=x$, for all $i=1,...,n-1$

(K11) $x\leq s_i(x)$ or $s_{i+1}(x)\leq x$, for all $i=1,...,n-1$.

A SH_n−model based on a structure K is a system $M=(K, m)$ such that m is a meaning function assigning subsets of W to propositional variables, and satisfies the so called atomic heredity condition:

(her) $x \leq y$ and $x \in m(p)$ imply $y \in m(p)$ for any propositional variable p.

We say that in a SH_n−model M, a state x satisfies a formula A, and we write M, x sat A, if the following conditions are satisfied:

M, x sat p iff $x \in m(p)$, for any propositional variable p.

M, x sat $A \vee B$ iff M,x sat A or M, x sat B

M, x sat $A \wedge B$ iff M, x sat A and M, x sat B

M, x sat $A \rightarrow B$ iff for all y, if $x \leq y$ and M, y sat A, then M, y sat B

M, x sat $\neg A$ iff for all y, if $x \leq y$ then not M, y sat A

M, x sat $S_i A$ iff M, $s_i(x)$ sat A

M, x sat $-A$ iff not M, g(x) sat A.

It follows that implication and negation $\neg$ are the intuitionistic operations, the negation $-$ is de Morgan negation, and the operations S_i are dual to the Post operations D_i.

A formula A is true in a SH_n−model iff M, x sat A for all $x \in W$, A is SH_n−valid iff it is true in all SH_n −models. It is known that logic SH_n is complete with respect to the given semantics (Iturrioz and Orlowska 1996).

For $n \leq 4$ the SH_n logics whose models satisfy the additional condition :

$x \leq g(x)$ or $g(x) \leq x$

are mutually interpretable with logics $_n$. For n>4 a further extension is needed. The respective class of logics is presented in (Cignoli 1982).

Relational structure $ReSH_n$, $n \geq 1$, for the logic SH_n is of the form:

$ReSH_n=(W, \leq, \{R_i: i=1,...,n-1\}, R_g, 1, 1')$

where W is a nonempty set, $\leq$ is a binary relation on W, $1=W \times W$, 1’ is the identity on W and the following conditions are satisfied:

(k1) $\leq$ is reflexive and transitive

(k2) $\leq \subseteq R_g;\geq;R_g^{-1}$ where $\geq = \leq^{-1}$

(k3) $R_g;R_i = R_{n-i};g$, for all $i=1,...,n-1$

(k4) $R_g;R_g = 1'$

(k5) $R_j;R_i = R_i$, for all $i,j=1,...,n-1$

(k6) $R_1; \leq = 1$

(k7) $R_{n-1};\geq = 1$

(k8) $1' \subseteq R_i;\leq;R_j^{-1}$, for $i \leq j$

(k9) $\leq \subseteq R_i; \leq;R_i^{-1}$ and $\leq \subseteq R_i;\geq;R_i^{-1}$, for all $i=1,...,n-1$

(k10) $1' \subseteq -(R_i;\leq \cap R_i;\geq) \cup R_i$, for all $i=1,...,n-1$

(k11) $1' \subseteq R_i;\geq \cup R_{i+1};\leq$, for all $i=1,...,n-2$

(k12) $R_i;R_i^{-1} \subseteq 1'$ and $R_i;1 = 1$

(k13) $R_g;R_g^{-1} \subseteq 1'$ and $R_g;1=1$.
Conditions (k1),...,(k11) are the relational counterparts to conditions (K1),..,(K11). Conditions (k12) and (k13) say that R_i and R_g are functional.

Each condition expressing the equality of two relations requires two rules, one for each of the implied inclusions. The rule corresponding to the inclusion $\subseteq$ associated to the condition (k3) is as follows:

(rk3$\subseteq$) $\quad$ K, $x-R_gz$, $z-R_iy$, H

K, $x-R_{n-i}t$, $t-R_gy$, H, $x-R_gz$, $z-R_iy$

where t is a restricted variable

The rule corresponding to (k6) is:

(rk6) $\quad$ K

K, $x-R_1z$, $z-\leq y$

where x, y are any variables and z is a restricted variable

The rule corresponding to the first condition in (k9) is:

(r9) $\quad$ K, $x-\leq y$, H

K, $x-R_iz$, $z-Rt$, $y-R_it$, H

where z, t are restricted variables

Axiomatic sequences of $ReSH_n$ are those of the basic relational formalism with $R=\leq, \geq, R_i, R_g$.

5. Relational mv−formalism

The class of mv−algebras was introduced in (Chang 1958). These algebras are a counterpart to the infinite−valued ukasiewicz logics. The formulation presented below is from (Hajek 1999).
An mv−algebra is a structure of the form: $(L, \cup, \cap, *, \rightarrow, 0, 1)$, where $(L, \cup, \cap, 0, 1)$ is a lattice with the smallest element 0 and the greatest element 1; $(L, *, 1)$ is a commutative semigroup, that is $*$ is a binary operation satisfying (mv1), (mv2), (mv3) below:

(mv1) $*$ is commutative

(mv2) $*$ is associative

(mv3) $1*x = x$

$\rightarrow$ is a residuum of $*$, that is:

(mv4) for all $x, y, z\in L$, $z\leq x\rightarrow y$ iff $x*z\leq y$, where $\leq$ is the lattice ordering

and, moreover

$$(mv5)\ x \cap y = x * (x \to y)$$

$$(mv6)\ (x \to y) \cup (y \to x) = 1$$

$$(mv7)\ x = ((x \to 0) \to 0).$$

Let MV be the class of mv−algebras. We define a relational structure ReMV as follows. First, for every operation of mv−algebras we define the respective relation:

$R_{\cup}(x,y,z)$ iff $x \le z$ and $y \le z$ and for all t, if $x \le t$ and $y \le t$ then $z \le t$

$R_{\cap}(x,y,z)$ iff $x \ge z$ and $y \ge x$ and for all t, if $x \ge t$ and $y \ge t$ then $z \ge t$, where $\ge\ =\ \le^{-1}$

$R_{*}(x,y,z)$ iff $x*y = z$

$R_{\to}(x,y,z)$ iff $x \to y = z$

The relations are assumed to satisfy the following conditions:

(1) For all x, y there is z such that $R_{*}(x,y,z)$

(2) If $R_{*}(x,y,z)$ and $R_{*}(x,y,t)$ then z1't

(3) If $R_{*}(x,y,z)$ then $R_{*}(y,x,z)$

(4) If $R_{*}(x,y,u)$ and $R_{*}(u,z,v)$ then $R_{*}(y,z,t)$ implies $R_{*}(x,t,v)$

(5) $R_{*}(1,x,x)$

(6a) If $R_{\to}(x,y,t)$ and $z \le t$ then $R_{*}(x,z,u)$ implies $u \le y$

(6b) If $R_{*}(x,z,u)$ and $u \le y$ then $R_{\to}(x,y,t)$ implies $z \le t$

(7a) If $R_{\cap}(x,y,z)$ then $R_{\to}(x,y,u)$ implies $R_{*}(x,u,z)$

(7b) If $R_{\to}(x,y,u)$ and $R_{*}(x,u,z)$ then $R_{\cap}(x,y,z)$

(8) If $R_{\to}(x,y,u)$ and $R_{\to}(y,x,v)$ then $R_{\cup}(u,v,1)$

(9) If $R_{\to}(x,0,u)$ and $R_{\to}(u,0,v)$ then x1'v

Conditions (1) and (2) say that relation R_{*} is a total function. Condition (3) reflects commutativity of $*$ and (4) its associativity. Observe that because of commutativity we need to assume only the counterpart to the inequality $x*(y*z) \le (x*y)*z$. Condition (5) corresponds to (mv3). Conditions (6a), (6b) and (7a), (7b) are the counterparts to (mv4) and (mv5), respectively. And finally, (8) and (9) reflect (mv6) and (mv7).

It is easy to see that the condition (1) can be expressed in the form (c2) from Principle II, and the conditions (2), (3), (4), (6a), (6b), (7a), (7b), (8), (9) are of the form (c1) from Principle I. By way of example we present below the rules for (1) and (6a).

(r1) K

K, x−Ry

where x is any variable and y is a restricted variable

(r6a) K, u≤y, H

$K, R_{\rightarrow}(x,y,t), K, u \leq y \mid K, z \leq t, H, u \leq y \mid K, R_{*}(x,z,u), H, u \leq y$

where x, z, t are any variables

Axiomatic sequences of ReMV are those of the basic relational formalism with $R = R_{\cup}, R_{\cap}, R_{*}, R_{\rightarrow}, \leq$, and any sequence containing the following expression corresponding to condition (5):

(a4) $R_{*}(1,x,x)$.

6. Relational structures for many−valued information systems

An information system with many−valued information is a structure of the form: (OB, AT, $\{VAL_a: a \in AT\}$) where OB is a nonempty set of objects, AT is a nonempty set of mappings a: $OB \rightarrow P(VAL_a)$ referred to as attributes and each VAL_a is the set of values of attribute a and $P(VAL_a)$ is its powerset. Systems of that form were introduced in (Lipski 1976) and named as the systems with incomplete information. The values of attributes represent properties of objects. For example, if OB is a set of persons and a is the attribute 'languages spoken', then if a(x)={English, French} for some $x \in OB$, then the a−properties of 'speaking English' and 'speaking French' might be attributed to x. However, such an information is not complete. Observe that there are four ways of interpreting the assignment a(x):

− conjunctively and exhaustively: x speaks English, French and no other languages;

− conjunctively and non−exhaustively: x speaks English, French and possibly some other languages;

− disjunctively and exhaustively: x speaks either English or French and no other languages;

− disjunctively and non−exhaustively: x speaks either English or French and possibly some other languages.

Hence, a complete representation of many−valued information requires a more refined concept of information system. For that purpose we define a *relational information system* (see also Düntsch et al. 1999) as a structure of the form (OB, AT, $\{VAL_a: a \in AT\}$, $\{I_a, B_a, H_a\}$, Δ)where OB, AT, VAL_a are as above; I_a, B_a, H_a are binary relations on the set $OB \times VAL_a$ such that:

(is1) $I_a \cap B_a = \emptyset$ and

(is2) $H_a = I_a \cup B_a$;

and Δ is a set of relational constraints.

Relations I_a and B_a are interpreted as follows:

$(x,v) \in I_a$ iff x certainly has the a−property v;

$(x,v) \in B_a$ iff x possibly has the a−property v.

Relations I_a, therefore enable us to represent conjunctive information about a−properties while relations B_a enable us to represent disjunctive information.

Relational constraints are represented in the form of binary relations determined by sets $I_a(x)=\{v \in VAL_a: (x,v) \in I_a\}$, $B_a(x)$, $H_a(x)$ (defined in the analogous way), for $x \in OB$ and $a \in AT$.

Let 1', $\subset$, $\supset$, D, PO be binary relations on a family of subsets of a set such that 1' is the identity, $\subset$, $\supset$ are proper inclusions, and D, PO are defined as follows:

$(X,Y) \in D$ iff $X \cap Y = \emptyset$ (disjointness relation)

$(X,Y) \in PO$ iff $X \cap Y \neq \emptyset$ and $(X,Y) \notin\ = \cup \subset \cup \supset$ (partial overlap relation).

The five relations above satisfy the following properties:

(is3) Their union equals 1 (the universal relation);

(is4) Any two of the relations are disjoint;

(is5) D is irreflexive and symmetric;

(is6) PO is irreflexive and symmetric;

(is7) $\subset$ and $\supset$ are irreflexive, transitive and $\subset\ = \supset^{-1}$;

(is8) 1' is reflexive, symmetric and transitive.

The atomic constraints in Δ have the form xTy where x, y represent objects and the relation T on the set OB has the following form:

$$(x,y) \in T \text{ iff } (R(x), S(y)) \in Q$$

with $R, S \in \{I_a(x), B_a(x), H_a(x): x \in OB, a \in AT\}$ and $Q \in \{1', \subset, \supset, D, PO\}$.

Hence, any relation T is of the form $R;Q;S^{-1}$, where ; is the composition of heterogeneous relations defined as follows. Let $P_1 \subseteq X \times Y$ and $P_2 \subseteq Y \times Z$, then $P_1;P_2 = \{(x,z) \in X \times Z$: there is $y \in Y$ such that $(x,y) \in P_1$ and $(y,z) \in P_2\}$. Any relation of the form T is referred to as an *information relation*.

The compound constraints are built from atomic constraints. They are of the form $T_1 \subseteq T_2$, where T_1 and T_2 are information relations.

It is easy to see that these constraints are instances of condition (c2) from section 3, and hence Principle II applies to them. Thus a formalism for relational information systems consists of the three parts. The first part consists of the basic relational formalism applied to the relations I_a, B_a, H_a, 1', $\subset$, $\supset$, D, PO. The second part consists of the rules and/or axiomatic sequences adequate for expressing the specific properties (is1),...,(is8) of the relations I_a, B_a, H_a, 1', $\subset$, $\supset$, D, PO. The third part consists of the rules adequate for the constraints from Δ. Below we present examples of rules from the second and the third part.

Let $R, S \in \{1', \subset, \supset, D, PO\}$, $R \neq S$. Then the rules adequate for the condition (is4) are of the form:

(ris4) K

K, xRy | K, xSy

where x, y are any variables

Now assume that the following constraint is a member of Δ. Let $T_1 = I_a;1';I_a^{-1}$ and $T_2 = I_a;D;B_a^{-1}$. Then the constraint $T_1 \subseteq T_2$ is represented by the following formula:

(δ) $\forall x,y,z,t(xI_az \wedge z1't \wedge yI_at \rightarrow \exists u,v(xI_au \wedge uDv \wedge yB_av))$.

Let L denote the sequence: $x{-}I_az$, $z{-}1't$, $y{-}I_at$.

According to Principle II the respective rule adequate for (δ) is:

(rδ) K, L, H

K, $x-I_a u$, H, L | K, $u-Dv$, H, L | K, $y-B_a v$, H, L

where u, v are restricted variables.

7. Conclusion

In this paper we presented three typical theories that deal with many−valued information: the class of SH_n logics, $n\geq 1$, the class of mv−algebras, and the class of many−valued information systems. For each of these theories we defined a corresponding relational theory. We outlined general principles of forming proof systems for relational theories. We showed how these principles lead to construction of deduction systems for the theories mentioned above.

The classical proof theory aims at constructing deduction systems without any reference to semantics of the languages of the theories in question. By contrast, an essential feature of the relational proof theory is that it attempts to encode the semantics in the proof rules. Our developments follow this methodology. The three theories considered in the paper are presented semantically and their relational counterparts provide a representation of the respective semantic structures.

References

Belnap, N., D. (1977) A useful four−valued logic. In: Dunn, J. M. and Epstein, G. (eds) Modern Uses of Multiple−Valued Logic, Reidel, Dordrecht, 8−37.

Buszkowski, W. and Orlowska, E. (1997) Indiscernibility−based formalization of dependencies in information systems. In: Orlowska,E. (ed) Incomplete Information: Rough Set Analysis. Physica Verlag, series Studies in Fuzziness and Soft Computing vol 13, 347−380.

Chang, C. C (1958) Algebraic analysis of many−valued logic. Transactions of the AMS 88, 467−490.

Cignoli, R. (1982) Proper n−valued ukasiewicz algebras as S−algebras of ukasiewicz n−valued propositional calculi. Studia Logica 41, 3−16.

Dobrowolska−Buffoli, B. (1998) Relational knowledge representation: Proof systems and decidability problems. PhD thesis, Institute of Computer Science, Polish Academy of Sciences, Warsaw.

Düntsch, I. and Orlowska, E. (1999) A proof system for contact relation algebras. Journal of Philosophical Logic, to appear.

Düntsch, I., Gediga, G. and Orlowska, E. (1999) Relational attribute systems. Submitted for publication.

Düntsch, I., Orlowska, E. and Hui Wang (2000) An algebraic and logical approach to the approximation of regions. In: Desharnais, J. (ed) Proceedings of the 5th International Seminar on Relational Methods in Computer Science, Université Laval, Quebec, Canada, 65−74.

Epstein, G. (1960) The lattice theory of Post algebras. Transactions of the American Mathematical Society 95, 300−317.

Frias, M. and Orlowska, E. (1995) A proof system for fork algebras and its applications to reasoning in logics based on intuitionism. Logique et Analyse 150−151−152, 239−284.

Hajek, P. (1999) Metamathematics of Fuzzy Logic. Series Trends in Logic, Studia Logica Library, Kluwer.

Iturrioz, I. (1983) Symmetric Heyting algebras with operators. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 29, 33−70.

Iturrioz, L. and Orlowska, E. (1996) A Kripke−style and relational semantics for logics based on ukasiewicz algebras. International Conference on the Work of Jan ukasiewicz: ukasiewicz in Dublin, Dublin 7−10 July, 1996.

Konikowska, B., Morgan, Ch. and Orlowska, E. (1998) Relational semantics for arbitrary finite−valued logics. In: Orlowska, E. and Szalas, A. (eds) Proceedings of the 4th International Seminar on Relational Methods in Logic, Algebra and Computer Science, RelMiCS'98, 138−143.

Konikowska, B., Morgan, Ch. and Orlowska, E. (1998a) A relational formalisation of arbitrary finite−valued logics. Logic Journal of IGPL 6 No 5, 1998, 755−774.

Lipski, W. (1976) Informational systems with incomplete information. In: Michaelson, S. and Milner, R. (eds) Third International Colloquium on Automata, Languages and Programming, Edinburgh University Press, 120−130.

ukasiewicz, J. (1920) O logice trójwartociowej. Ruch Filozoficzny 5, 170−171. English translation: On three−valued logic. In: J. ukasiewicz Selected Works. North Holland, Amsterdam, 1970, 87−88.

MacCaull, W. (1997) Relational proof theory for linear and other substructural logics. Logic Journal of the Interest Group of Pure and Applied Logic 5, 673−697.

MacCaull, W. (1998) Relational tableaux for tree models, language models and information networks. In: Orlowska, E. (ed) Logic at Work. Essays Dedicated to the Memory of Helena Rasiowa, Physica Verlag, Heidelberg, 354−382.

MacCaull, W. (1998a) Relational semantics and a relational proof system for full Lambek calculus. Journal of Symbolic Logic 63, No 2, 623−637.

MacCaull, W. (2000) Proof theory for generalized dependencies. Fundamenta Informaticae, to appear.

Orlowska, E. (1988) Relational interpretation of modal logics. In: Andreka, H., Monk, D. and Nemeti, I. (eds) Algebraic Logic. Colloquia Mathematica Societatis Janos Bolyai 54, North Holland, Amsterdam, 443–471.

Orlowska, E. (1996) Relational proof systems for modal logics. In: Wansing, H. (ed) Proof Theory of Modal Logics. Kluwer, 55−77.

Orlowska, E. (1997) Relational formalisation of nonclassical logics. In: Brink, C. Kahl, W. and Schmidt, G. (eds) Relational Methods in Computer Science, Springer, Wien − New York, 90−105.

Post, E. (1920) Introduction to a general theory of elementary propositions. Bulletin of the American Mathematical Society 26.

Post, E. (1921) Introduction to a general theory of elementary propositions. American Journal of Mathematics 43, 163−185.

Spencer, B. and MacCaull, W. (2000) ReVAT − Relational Validator via Analytic Tableaux. Technical Report, University of New Brunswick.