Brock University

Department of Computer Science

A Multi-Modal Logic for Disagreement and Exhaustiveness

Ivo Düntsch and Beata Konikowska
Technical Report # CS-07-01
January 2007

Brock University
Department of Computer Science
St. Catharines, Ontario
Canada L2S 3A1
www.cosc.brocku.ca

A multi–modal logic for disagreement and exhaustiveness*

Ivo Düntsch
Department of Computer Science
Brock University
St. Catharines, Ontario, Canada, L2S 3A1,
duentsch@brocku.ca

Institute of Computer Science,
Polish Academy of Sciences
Ordona 21, 01–237 Warsaw, Poland
beatak@ipipan.waw.pl

January 2, 2007

Abstract

The paper explores two basic types of relations betwen objects of a Pawlak-style information system generated by the values of some attribute of those objects: disagreement (disjoint sets of values) and exhaustiveness (sets of values adding up to the whole universe of the attribute). Out of these two fundamental types of relations, most other types of relations on objects of an information system considered in the literature can be derived – as, for example, indiscernibility, similarity and complementarity. The algebraic properties of disagreement and indiscernibility relations are explored, and a representation theorem for each of these two types of relations is proved.

The notions of disagreement and exhaustiveness relations for a single attribute are extended to relations generated by arbitrary sets of attributes, yielding two families of relations parametrized by sets of attributes. They are used as accessibility relations to define a multi–modal logic with modalities corresponding to the lower and upper approximation of a set in Pawlak's rough set theory. Finally, a complete Rasiowa-Sikorski deduction system that logic is developed.

*Ivo Düntsch gratefully acknowledges support by the Natural Sciences and Engineering Research Council of Canada. Both authors gratefully acknowledge partial support by the KBN/British Council Grant No WAR/992/174.

1 Introduction

Knowledge representation in the form of an

$$\text{Object} \mapsto \text{Attribute values}$$

assignment is perhaps the oldest and most widely used method of structuring information, reflecting the intention- extension paradigm of cognition. This assignment gave rise to the formal notion of an *i*nformation system introduced by Z. Pawlak in the early 1980s [11]. His general model of an information system proved to be a seminal notion, capturing the most essential features of any information framework by the simplest mathematical means possible. It gave rise both to an extensive body of research, including the famous rough set theory also developed by Pawlak [12], and to many practical applications in areas like production control systems, expert systems, decision systems and the like.

Our paper is one more attempt to draw on the power of the above paradigm, aimed at characterization of some basic relations between elements of an information system in the Pawlak's style. Accordingly, let us start with introducing the fundamental notion of the information system underlying this paper.

An *information system* is a triple $\mathscr{I} = \langle U, \textsc{Attr}, \{V_a\}_{a\in\textsc{Attr}}\rangle$ such that

1. U and each $V_a, a \in \textsc{Attr}$, are finite non-empty sets.
2. $\textsc{Attr}$ is a finite set of mappings $a : U \to 2^{V_a}$.

We interpret U as a set of objects (subjects, agents, etc.), and $\textsc{Attr}$ as a set of attribute functions, each of which assigns to each $x \in U$ a set of values from V_a. There are various ways in which $a(x)$ can be interpreted: One possibility is to say that x possesses all properties listed in $a(x)$, another, that x possesses (only) some of these properties; following from this, there are several ways of interpretation such as "x may have any of the properties in $a(x)$, but we do not know (or do not care) which", and related statements. For the purposes of this paper, the semantic interpretation of $a(x)$ is not relevant.

Given $a(x)$ and $a(y)$, there are various ways in which these sets can be related with respect to the (result of) the set operations in 2^{V_a}, for example

$$a(x) = a(y), a(x) \subseteq a(y),\ a(x) \supseteq a(y),\ a(x) \cap a(y) = \emptyset,\ a(x) \cup a(y) = V_a,$$

and Boolean combinations of these and similar relations. These relations, in turn, can be used to define binary relations on U in the following way: Suppose that $a \in \textsc{Attr}$, and T is a binary relation on subsets of V_a. T defines a binary relation $R_{T,a}$ on U by setting

$$xR_{T,a}y \iff a(x)Ta(y). \tag{1.1}$$

More generally, suppose that $f : U \to W$ is a mapping, and that T is a binary relation on W. We define the *pre-image relation of* T *with respect to* f by

$$x \operatorname{pre}(T,f)y \iff f(x)Tf(y). \tag{1.2}$$

Clearly, $\operatorname{pre}(T,f)$ is the largest relation R on U such that $xRy \iff f(x)Tf(y)$, and it shares many properties with T:

Lemma 1.1. *[1] Suppose that ϕ is a formula in a first order language (without equality) with one binary relation symbol, and that $f : A \to B$ is surjective. Then, $\langle A,\ pre(S,f)\rangle \models \phi \iff \langle B,S\rangle \models \phi$.*

In particular, if f is surjective and P is one of the properties "reflexive", "symmetric", "transitive", then $\operatorname{pre}(T,f)$ has property P if and only if T has.

Besides the case where T is the equality relation, which is well understood, the *complementarity relation* induced by

$$a(x) = -a(y)$$

has received some attention. Complementarity relations and their logics have been studied in detail in [3, 5, 6]. Abstractly, we say that C is a *complementarity relation* if it has the properties

$$C \cap 1' = \emptyset, \text{ i.e. } C \text{ is irreflexive,} \tag{1.3}$$

$$C = C^{\smile}, \text{ i.e. } C \text{ is symmetric,} \tag{1.4}$$

$$C \mathbin{;} C \mathbin{;} C \subseteq C, \text{ i.e. } C \text{ is 3-transitive.} \tag{1.5}$$

Here, $1'$ is the identity relation on U, $C^{\smile}$ is the relational converse of C, and $;$ denotes relational composition; these are formally defined in Section 2.

Real world examples of complementarity relations seem to be hard to come by, and thus, one looks for less restrictive conditions. One natural idea is to split the two conditions which define complementarity, namely

$$a(x) = -a(y) \iff a(x) \cap a(y) = \emptyset \text{ and } a(x) \cup a(y) = V_a,$$

and look at them separately. This leads to the following two definitions:

Suppose that $a \in \textsc{Attr}$; the *relation of disagreement* D_a, determined by a, is defined by

$$xD_a y \iff a(x) \cap a(y) = \emptyset. \tag{1.6}$$

An instance of a disagreement relation is the following scenario: Suppose that U is a group of medical experts, and that each $a \in \textsc{Attr}$ represents some disease. The value $a(x)$ indicates the set of symptoms which expert x assigns to disease a. If $a(x) \cap a(y) = \emptyset$, then the experts x and y totally disagree on the question which symptoms belong to the disease.

Similarly, we define the *relation of exhaustiveness* E_a by

$$xE_ay \iff a(x) \cup a(y) = V_a. \tag{1.7}$$

If the information system $\mathscr{I}$ in which we work is important, we will write $D_a^{\mathscr{I}}$ and $E_a^{\mathscr{I}}$.

D_a and E_a are, respectively, the relations of right and left orthogonality of [10]. An important motivation for considering them is that most information relations appearing in the literature can be obtained from those two basic relations using the Boolean set operators and Tarski's relational operators.

The relation of complementarity C_a can now be recovered by

$$C_a = D_a \cap E_a.$$

Observe that D_a is not necessarily irreflexive. To make D_a irreflexive we would have to require the attribute functions to have non-empty values sets. However, this is overly restrictive: If we stay with the experts example above, one of them may not have any opinion, or choose not to answer a particular question for other reasons.

There are various ways to define relations arising from a set of attributes. Given a non-empty set $A \subseteq \textsc{Attr}$, we can consider

$$\begin{aligned} xR^s_{T,A}y &\iff a(x)Ta(y) \text{ for all } a \in A, \\ xR^w_{T,A}y &\iff a(x)Ta(y) \text{ for some } a \in A. \end{aligned}$$

We do not allow the case $A = \emptyset$, since we do not want to make an ontological commitment for the sake of mathematical expediency. $R^s_{T,A}$ is called the *strong version of* $\{R_{T,a} : a \in A\}$, and $R^w_{T,A}$ the *weak version*; these are only what one might call "extreme" cases of possible combinations of the relations $R_{T,a}$.

In this paper, we shall study the relational properties and representability of disagreement and exhaustiveness relations, and provide (multi-)modal logics for handling information systems with these information relations.

2 Relational properties

The set of all binary relations on U is denoted by $Rel(U)$. If $R, S \in Rel(U)$, then

$$xR \,;\, Sy \iff (\exists z \in U)xRzSy.$$

is the *relational composition* of R and S, and

$$R^{\smile} = \{\langle y,x\rangle : xRy\}$$

is the *converse of* R. Furthermore, we define

$$\begin{aligned}
1' &= \{\langle x,x\rangle : x \in U\}, && \text{identity relation,}\\
0' &= \{\langle x,y\rangle : x,y \in U, x \neq y\}, && \text{diversity relation,}\\
\mathbf{1} &= \{\langle x,y\rangle : x,y \in U\}, && \text{universal relation,}\\
R(x) &= \{y \in U : xRy\}, && \text{range of } x \text{ in } R,\\
\operatorname{ran} R &= \bigcup\{R(x) : x \in U\} && \text{range of } R,\\
\operatorname{dom} R &= \{x : (\exists y)xRy\}, && \text{domain of } R.
\end{aligned}$$

It is easy to see that D_a and E_a are symmetric. Indeed, they have the same first order properties: Given $\mathscr{I} = \langle U, \textsc{Attr}, \{V_a\}_{a\in\textsc{Attr}}\rangle$ and $a \in \textsc{Attr}$, we define

$$\mathscr{I}' = \langle U, \textsc{Attr}', \{V_b\}_{b\in\textsc{Attr}'}\rangle$$

by

$$\textsc{Attr}' = \{b_a : a \in \textsc{Attr}\}, \qquad V_{b_a} = V_a, \qquad b_a(x) = V_{b_a} \setminus a(x),$$

for all $a \in \textsc{Attr}, x \in U$. Then, for all $x,y \in U$,

$$xD_a y \iff xE_{b_a} y. \tag{2.1}$$

The question arises, whether every symmetric relation can be represented as a relation of disagreement for some information system $\mathscr{I}$, and whether every relation having the properties (1.3) – (1.5) can be represented as some complementarity relation C_a in the following sense:

If $R \in Rel(U)$, we say that

1. R is *set representable with respect to disagreement*, if there are some set W and a mapping $a : U \to 2^W$ such that

$$xRy \iff a(x) \cap a(y) = \emptyset$$

   for all $x,y \in U$.

2. R is *set representable with respect to exhaustiveness* if there are some set W and a mapping $a : U \to 2^W$ such that

$$xRy \iff a(x) \cup a(y) = W$$

   for all $x,y \in U$.

3. R is *set representable with respect to complementarity*. if there are some set W and a mapping $a : U \to 2^W$ such that

$$xRy \iff a(x) \cup a(y) = W,\ a(x) \cap a(y) = \emptyset$$

   for all $x,y \in U$.

It has been shown in [4] by a general method that each of these relational types has a set representation. Below, we shall give their concrete representations as pre–image relations.We first make an observation regarding the "reflexive parts" of disagreement and exhaustiveness relations: If D_a is a disagreement relation, then

$$xD_ax \iff a(x) = \emptyset \iff (\forall y)xD_ay. \tag{2.2}$$

Similarly, if E_a is an exhaustiveness relation, then

$$xE_ax \iff a(x) = V_a \iff (\forall y)xE_ay. \tag{2.3}$$

In relational terms, (2.2) and (2.3) become

$$(1' \cap R) \; ; \; (-R) = \mathbf{0}. \tag{2.4}$$

with $D_a = R$ or $E_a = R$.

Theorem 2.1. *Suppose that $R \in Rel(U)$ is symmetric and satisfies* (2.4)*. Then, R is set representable with respect to disagreement.*

Proof: Let $W = \{\{x,y\} : x,y \in U\}$, and define $a : U \to 2^W$ by

$$x \overset{a}{\mapsto} \{\{x,y\} : x(-R)y\}.$$

We show that

$$a(x) \cap a(y) = \emptyset \iff xRy.$$

"$\Rightarrow$": Let $x(-R)y$. Then, $\{x,y\} \in a(x)$, and the symmetry of R (and hence, of $-R$) implies that $\{x,y\} \in a(y)$ as well. It follows that $a(x) \cap a(y) \neq \emptyset$.

"$\Leftarrow$": Suppose that xRy. We distinguish two cases:

1. $x \neq y$: Assume that $\{s,t\} \in a(x) \cap a(y)$. Since $x \neq y$, we suppose without loss of generality that $x = s,\ x(-R)t$ and $y = t,\ y(-R)s$. It follows that $x(-R)y$ contrary to our assumption.

2. $x = y$: Assume that $a(x) \neq \emptyset$. Then, xDx implies that there is some $z \in U$ such that $x \neq z$ and $x(-R)z$. Since xRx, this contradicts (2.4). $\square$

Note that (2.4) was only required in the case when R was not irreflexive, and that the finiteness of U was not needed. From (2.1) we obtain

Corollary 2.2. *Suppose that $R \in Rel(U)$ is symmetric and satisfies* (2.4)*. Then, R is set representable with respect to exhaustiveness.*

Our next aim is to give a concrete representation of a complementarity relation C. First, we shall consider a special case: Recall that a permutation of order 2 is a (bijective) mapping f for which $f \; ; \; f = 1'$. In this case, f can be written as a product of cycles of (s,t) of length 2 and cycles (s,s) of length 1. A *fixpoint* of f is some $x \in U$ for which $f(x) = x$, i.e. the cycle containing x has length 1.

Lemma 2.3. *Suppose that C is a permutation on U of order 2 without fixpoints. Then, $\langle U, C\rangle$ has a complementarity set representation.*

Proof. Suppose that $|U| = n$ and

$$C = (s_1, t_1)(s_2, t_2) \dots (s_n, t_n)$$

in cycle form; note that the condition on C implies that n is an even number. For each $1 \leq i \leq n$, let $\mathscr{P}_i$ be a partition of U with classes C_1^i, C_2^i such that all the $\mathscr{P}_i$'s are different. If we set $a(s_i) = C_1^i,\ a(t_i) = C_2^i$, then $xCy \iff a(x) = -a(y)$. □

Theorem 2.4. *If $C \in Rel(U)$ satisfies (1.3)–(1.5), then C has a set representation with respect to complementarity.*

Proof. Since C is symmetric, we have $\operatorname{ran} C = \operatorname{dom} C$ as well, and therefore we can assume without loss of generality that $\operatorname{dom} C = U$. Define a binary relation θ on U by

$$x\theta y \iff C(x) = C(y).$$

Clearly, θ is an equivalence relation on U. For each $x \in U$ we let x' be its equivalence class, and set $U' = \{x' : x \in U\}$; furthermore, $\pi : U \twoheadrightarrow U'$ is the quotient mapping. Next, we define a relation C' on U' by

$$x'C'y' \iff xCy.$$

Since $x'C'y' \iff \pi^{-1}(x') \times \pi^{-1}(y') \subseteq C$, C' is well defined, and it is just the pre–image relation of C with respect to π. Hence, C' satisfies (1.3)–(1.5) by Lemma 1.1.

Our next aim is to show that C' is a permutation of order 2 without fixpoints. Since $\operatorname{dom} C = U$, we have $\operatorname{dom} C' = U'$, so that C' is total. Suppose that $x'C'y'$ and $x'C'z'$. If $s \in y'$, then sCy by symmetry of C, and thus, $sCyCxCz$, again using symmetry. The 3–transitivity of C now implies sCz, and thus, $s \in z'$. Similarly, $z' \subseteq y'$, and therefore, C' is a function. The same argument shows that C' is one–to–one with order 2, and it follows from the irreflexivity of C' that it has no fixpoints.

Let $a' : U' \to 2^V$ be a representation of $\langle U', C'\rangle$ which exists by Lemma 2.3, and define $a : U \to 2^V$ by

$$a(x) = a'(x').$$

Then, $xCy \iff a(x) = -a(y)$: If xCy, then $x'C'y'$, and it follows that $a(x) = a'(x') = -a'(y') = -a(y)$.

Conversely, suppose that $a(x) = -a(y)$, and that $s \in x',\ t \in y'$. We have to show that sCt: Since $s \in x',\ t \in y'$, we have

$$a'(s') = a'(x') = -a'(y') = -a'(t'),$$

and therefore, $s'C't'$. Since C is the pre–image relation of C' it follows that sCt. □

Suppose that D, E are symmetric relations on U, and that $C = D \cap E$. Besides (2.4) and (2.3), necessary conditions for joint representability, i.e. $D = D_a$ and $E = E_a$ for some attribute a, are

$$(\forall x,y,z,u)[xDyEzDu \Rightarrow xDu] \qquad \text{i.e. } D \,;\, E \,;\, D \subseteq D, \tag{2.5}$$

$$(\forall x,y,z,u)[xEyDzEu \Rightarrow xEu] \qquad \text{i.e. } E \,;\, D \,;\, E \subseteq E, \tag{2.6}$$

$$(\forall x,y)[xDy \text{ and } yEy \Rightarrow xDx] \qquad \text{i.e. } (D \,;\, (E \cap 1') \,;\, \mathbf{1}) \cap 1' \subseteq D \cap 1', \tag{2.7}$$

$$(\forall x,y)[xEy \text{ and } yDy \Rightarrow xEx] \qquad \text{i.e. } (E \,;\, (D \cap 1') \,;\, \mathbf{1}) \cap 1' \subseteq E \cap 1', \tag{2.8}$$

$$(\forall x)[\neg(xDx \text{ and } xEx)] \qquad \text{i.e. } D \cap E \cap 1' = \emptyset, \tag{2.9}$$

$$(\forall x,y,z)[yCxCz \Rightarrow D(y) = D(z) \text{ and } E(y) = E(z)] \qquad \text{i.e. } C \,;\, C \subseteq -(D \,;\, -D) \cup (-D \,;\, D) \cup (-E \,;\, E) \cup (E \,;\, -E)) \tag{2.10}$$

The issue whether these conditions are sufficient for joint representability remains an open question.

3 A logic for disagreement and exhaustiveness

3.1 The language

The language $\mathscr{L}$ is parametrized by a fixed nonempty set ATTR, which stands for the set of all attributes. The alphabet of $\mathscr{L}$ is the disjoint union of the following sets:

$\{\mathbf{0}\}$	where $\mathbf{0}$ represents an empty set of attributes.
CONA $= \{\mathbf{a} : a \in$ ATTR$\}$	set of constants representing individual attributes, one constant $\mathbf{a}$ for each single attribute a.
VARSA	set of variables representing sets of attributes.
VARO	set of variables representing individual objects.
VARSO	set of variables representing sets of objects.
$\{-, \cup, \cap\}$	symbols for set-theoretic operations on sets of attributes.
$\{\neg, \vee, \wedge\}$	symbols for set-theoretic operations on sets of objects.
$\{\langle\ \rangle_D, [\ ]_D, \langle\ \rangle_E, [\ ]_E\}$	symbols for modalities.

The language contains two types of expressions:

1. Terms, representing sets of attributes.
2. Formulae, representing sets of objects .

The set TERM of terms is the least set containing VARSA, CONA, $\mathbf{0}$ and closed under $-, \cup, \cap$.

The set FORM of formulae is the least set containing, VARO, VARSO, closed under $\neg, \vee, \wedge$, and such that if $A \in$ TERM and $F \in$ FORM, then $[A]_D F, \langle A\rangle_D F, [A]_E F, \langle A\rangle_E F \in$ FORM

We shall also use the derived operator $\longrightarrow$ defined by:

$$F \longrightarrow G \overset{df}{\equiv} \neg F \vee G. \tag{3.1}$$

3.2 Semantics of $\mathscr{L}$

Let us fix a nonempty, finite set ATTR, intuitively our set of attributes[1], and denote

$$2^{+\text{ATTR}} = 2^{\text{ATTR}} \setminus \{\emptyset\}$$

A *D-E frame*, or just *frame*, is a triple

$$\mathscr{F} = < U, \{D_A\}_{A \in 2^{+\text{ATTR}}}, \{E_A\}_{A \in 2^{+\text{ATTR}}} >, \tag{3.2}$$

where U is a nonempty universe of objects, and for all $x, y, z \in U,\ a \in$ ATTR, $A \in \mathscr{P}^+$,

$$D_A \subseteq U \times U \text{ is symmetric}, \qquad E_A \subseteq U \times U \text{ is symmetric}, \tag{3.3}$$

$$D_A = \bigcap_{a \in A} D_a, \qquad E_A = \bigcap_{a \in A} E_a, \tag{3.4}$$

$$xD_a yE_a zD_a u \Rightarrow xD_a u, \qquad xE_a yD_a zE_a u \Rightarrow xE_a u, \tag{3.5}$$

$$xD_a x \Rightarrow (\forall y) xD_a y, \qquad xE_a x \Rightarrow (\forall y) xE_a y, \tag{3.6}$$

and, furthermore,

$$\neg xD_a x \text{ or } \neg xE_a x, \tag{3.7}$$

$$xD_a y,\ xD_a z,\ xE_a y,\ xE_a z \Rightarrow (\forall u)(y\, D_a\, u \Leftrightarrow z\, D_a\, u \ \text{ and } y\, E_a\, u \Leftrightarrow z\, E_a\, u). \tag{3.8}$$

One can easily check that the above conditions correspond to (2.2) – (2.10); observe that (2.5) and (2.6) follow from (3.3) and (3.5)

For each $A \in 2^{+\text{ATTR}}$, we call D_A, E_A the *disagreement and exhaustiveness relations corresponding to (the attributes in) A, respectively*. Except for the conditions set forth above, we do not put any restrictions on the assignment $A \mapsto D_A, E_A$.

A *model of* $\mathscr{L}$ is a pair

$$M = \langle \mathscr{F}, v \rangle,$$

where $\mathscr{F} = < U, \{E_A\}_{A \in 2^{+\text{ATTR}}}, \{D_A\}_{A \in 2^{+\text{ATTR}}} >$ is a frame as in 3.2, and v is a (multi-sorted) valuation such that

- $v(A) \subseteq$ ATTR for $A \in$ VARSA,

[1]For simplicity, we use ATTR both for the syntax and the semantics.

- $v(O) \subseteq U$ for $O \in$ VARSO,
- $v(x) \in U$ for $x \in$ VARO.

The valuation v is extended in a natural way to an interpretation τ_M of terms and an interpretation φ_M of formulae by interpreting $\mathbf{0}$ as $\emptyset$, each constant $\mathbf{a} \in$ CONA as the corresponding attribute $a \in$ ATTR (or, rather, as the singleton set containing a), the symbols $-, \cup, \cap$ and $\neg, \vee, \wedge$ as set-theoretical operations on sets of attributes and sets of objects, respectively, and $\langle A \rangle_Z, [A]_Z$ as the possibility and necessity modalities corresponding to the accessibility relation $Z_{\tau_M(A)}$, where $Z \in \{E, D\}$. The last clause, however, is fraught with some difficulties. Namely, if $\tau_M(A)$ happens to be empty, then $D_{\tau_M(A)}, E_{\tau_M(A)}$ are not defined, since our model contains disagreement and exhaustiveness relations parametrized by non-empty sets of attributes only. Hence in such a case we cannot ascribe any defined value to any modal formula based on these relations, which necessarily makes our logic a partial one.

Let $\perp$ denote a special, 'undefined' value of a formula. Then, speaking more precisely, the *interpretation of terms in M* is a function $\tau_M : \text{TERM} \longrightarrow 2^{\text{ATTR}}$ defined inductively as follows:

1. $\tau_M(\mathbf{a}) = \{a\}$ for any $a \in$ ATTR, where $\mathbf{a}$ is the unique constant in CONA representing a,
2. $\tau_M(\mathbf{0}) = \emptyset$, $\quad \tau_M(A) = v(A)$ for $A \in$ VARSA,
3. For any $A, B \in$ TERM,

   $\tau_M(-A) = \text{ATTR} \setminus \tau_M(A), \qquad \tau_M(A \cup B) = \tau_M(A) \cup \tau_M(B), \qquad \tau_M(A \cap B) = \tau_M(A) \cap \tau_M(B)$

The interpretation of formulae is a function $\varphi_M : \text{FORM} \longrightarrow 2^U \cup \{\perp\}$ such that for all $x \in$ VARO, $O \in$ VARSO, $F, G \in$ FORM, and $A \in$ TERM,

$$
\begin{aligned}
\varphi_M(x) &= \{v(x)\}, \\
\varphi_M(\neg F) &= \begin{cases} \perp & \text{if } \varphi_M(F) = \perp, \\ U \setminus \varphi_M(F) & \text{otherwise,} \end{cases} \\
\varphi_M(F \vee G) &= \begin{cases} U & \text{if either } \varphi_M(F) = U \text{ or } \varphi_M(G) = U, \\ \varphi_M(F) \cup \varphi_M(G) & \text{if } \varphi_M(F) \neq \perp \neq \varphi_M(G), \\ \perp & \text{otherwise} \end{cases} \\
\varphi_M(F \wedge G) &= \begin{cases} \emptyset & \text{if either } \varphi_M(F) = \emptyset \text{ or } \varphi_M(G) = \emptyset, \\ \varphi_M(F) \cap \varphi_M(G) & \text{if } \varphi_M(F) \neq \perp \neq \varphi_M(G), \\ \perp & \text{otherwise} \end{cases} \\
\varphi_M([A]_Z F) &= \begin{cases} \perp & \text{if either } \tau_M(A) = \emptyset \text{ or } \varphi_M(F) = \perp \\ \{o \in U : (\forall o' \in U)((o, o') \in Z_{\tau_M(A))} \to o' \in \varphi_M(F))\} & \text{otherwise} \end{cases} \\
\varphi_M(\langle A \rangle_Z F) &= \begin{cases} \perp & \text{if either } \tau_M(A) = \emptyset \text{ or } \varphi_M(F) = \perp \\ \{o \in U : (\exists o' \in \varphi_M(F))(o, o') \in Z_{\tau_M(A))}\} & \text{otherwise.} \end{cases}
\end{aligned}
$$

where $Z \in \{D, E\}$.

Note that the semantics of possibility and necessity modalities corresponds to the notions of upper and lower approximations in Pawlak's rough set theory [12].

A formula $F \in \textsc{Form}$ is said to be:

- *true in a model M*, written $M \models F$, iff $\varphi_M(F) = U$, i.e. iff F evaluates to the whole object–universe of that model;
- *false in a model M* if $\varphi_M(F) \subset U$ but $\varphi_M(F) \neq U$,
- *undefined in a model M* if $\varphi_M(F) = \perp$.

A formula F is called *valid* iff $M \models F$ for every model M.

4 Signed formulae

For any $x, y \in \textsc{VarO}$, $F \in \textsc{Form}$, $A \in \textsc{Term}$ and $Z \in \{D, E\}$, we shall write:

$$\begin{array}{llll} x \in F & \text{for } \neg x \vee F & x \notin F & \text{for } x \in \neg F \\ x = y & \text{for } x \in y & x \neq y & \text{for } x \in \neg y \\ x Z_A\, y & \text{for } x \in \langle A \rangle_Z\, y & x \overline{Z_A}\, y & \text{for } x \notin \langle A \rangle_Z y \end{array} \tag{4.1}$$

It can be easily checked that the above notation corresponds to the semantics of these formulae, which will play a crucial role in the deduction system we shall develop below.

The deduction system will operate on signed formulae [2], belonging to the set

$$\textsc{SForm} = \{\mathbf{T}(F) : F \in \textsc{Form}\} \cup \{\mathbf{N}(F) : F \in \textsc{Form}\}$$

where $\mathbf{T}$ denotes the *is-satisfied* operator and $\mathbf{N}$ denotes the *is-not-satisfied* operator. Their semantics in a model M is given by $\sigma_M : \textsc{SForm} \Rightarrow \{\mathbf{tt}, \mathbf{ff}\}$, defined in the usual way by:

$$\sigma_M(\mathbf{T}(F)) = \begin{cases} \mathbf{tt} & \text{if } M \models F, \\ \mathbf{ff} & \text{otherwise} \end{cases} \qquad \sigma_M(\mathbf{N}(F)) = \begin{cases} \mathbf{tt} & \text{if } M \not\models F, \\ \mathbf{ff} & \text{otherwise} \end{cases}$$

It should be noted that since the original logic is three-valued, with the third, 'undefined' value being $\perp$, $\mathbf{N}(x \in F)$ is in general not equivalent to $\mathbf{T}(x \in \neg F)$.

Two more important facts we should also keep in mind are:

$$(\varphi_M(F) = \perp) \quad \Rightarrow \quad [\sigma_M(\mathbf{T}(x \in F)) = \sigma_M(\mathbf{T}(x \in \neg F)) = \mathbf{ff}], \tag{4.2}$$

$$(\tau_M(A) = \emptyset) \quad \Rightarrow \quad [\sigma_M(\mathbf{T}(x Z_A\, x)) = \sigma_M(\mathbf{T}(\neg x Z_A\, x)) = \mathbf{ff}], \tag{4.3}$$

for any $x \in \textsc{VarO}$.

Consequently, defining the signed formulae $\mathscr{D}_F, \mathscr{U}_F, \mathscr{E}_A, \mathscr{N}_A$ by

$$
\begin{array}{ll}
\mathscr{D}_F \stackrel{df}{=} \mathbf{T}(x \in [F \vee \neg F]) & \mathscr{U}_F \stackrel{df}{=} \mathbf{N}(x \in [F \vee \neg F]) \\
\mathscr{N}_A \stackrel{df}{=} \mathbf{T}(x \in [\langle A \rangle_Z x \vee \neg \langle A \rangle_Z x]) & \mathscr{E}_A \stackrel{df}{=} \mathbf{N}(x \in [\langle A \rangle_Z x \vee \neg \langle A \rangle_Z x])
\end{array}
\tag{4.4}
$$

where x is an arbitrarily chosen variable in VARO, and Z an arbitrarily chosen symbol in $\{D, E\}$, we have

$$
\begin{array}{lllllll}
\sigma_M(\mathscr{D}_F) = \mathbf{tt} & \Leftrightarrow & \varphi_M(F) \neq \bot & & \sigma_M(\mathscr{U}_F) = \mathbf{tt} & \Leftrightarrow & \varphi_M(F) = \bot \\
\sigma_M(\mathscr{N}_A) = \mathbf{tt} & \Leftrightarrow & \tau_M(A) \neq \emptyset & & \sigma_M(\mathscr{E}_A) = \mathbf{tt} & \Leftrightarrow & \tau_M(A) = \emptyset
\end{array}
\tag{4.5}
$$

The formulae introduced above will be very useful in dealing with undefinedness of formulae in the deduction system to be developed here. For this purpose, special decomposition rules for the $\mathscr{D}$, $\mathscr{U}$, $\mathscr{N}$, $\mathscr{E}$ operators, consistent with their semantics following from the above definitions, will be introduced. In consequence, the above operators, though definable in our language in the way shown above, will in fact be treated as independent constructors of the language in the decomposition process.

5 Components and normal form

In order to develop a complete deduction system for our polymodal logic, we must cope with the modalities corresponding to various sets of attributes. To this end, we use the "component" approach described in [7, 8] and [5] for the cases of the similarity logic and complementarity logics, respectively. Namely, we replace the terms appearing in signed formulae by unions of certain special terms called "components" which evaluate to a disjoint cover of ATTR in any model.

For an arbitrary finite sequence Ω of signed formulae, let

$$
\begin{array}{ll}
\text{CONA}(\Omega) & = \{\mathbf{a} \in \text{CONA} : \mathbf{a} \text{ occurs in } \Omega\}, \text{ and} \\
\text{VARSA}(\Omega) & = \{A \in \text{VARSA} : A \text{ occurs in } \Omega\}
\end{array}
$$

be the set of all constants in CONA and all variables in VARSA that appear in the terms of Ω.

Suppose that

$$
\begin{array}{ll}
\text{CONA}(\Omega) & = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}, \\
\text{VARSA}(\Omega) & = \{Q_1, \ldots, Q_m\},
\end{array}
$$

where $n, m \geq 0$. The sequence Ω is said to be *nondegenerate* iff $m + n > 0$, i.e. iff it contains symbols from $\text{VARSA} \cup \text{CONA}$[2].

Suppose that Ω is nondegenerate, and

$$A^+ = A, \quad A^- = -A$$

[2] The reader should note that $\mathbf{0}$ belongs neither to CONA nor to VARSA, whence any sequence whose terms contain no symbol for an attribute or sets of attributes other than $\mathbf{0}$ is considered as degenerate.

for any term A. We denote

$$\begin{array}{lcl} \text{SCOMP}(\Omega) & = & \{Q_1^{i_1} \cap \ldots \cap Q_m^{i_m} : i_1, \ldots, i_m \in \{+,-\}\}, \\ \text{COMP}(\Omega) & = & \{\mathbf{a} \cap S : \mathbf{a} \in \text{CONA}(\Omega), S \in \text{SCOMP}(\Omega)\} \end{array}$$

The elements of $\text{COMP}(\Omega)$ are called *components for* Ω, and those of $\text{SCOMP}(\Omega)$ *subcomponents*. Note that the notion of component is define for nondegenerate sequences only. Components will be always denoted by a suitably indexed C, and subcomponents by S.

The components and subcomponents have the following important properties, see [8] for a proof:

Lemma 5.1. *For any nondegenerate sequence Ω of signed formulae and any model M,*

1. *The sets $\{\tau_M(C)\}_{C \in \text{COMP}(\Omega)}$ form a disjoint cover of* ATTR*, and so do the sets $\{\tau_M(S)\}_{S \in \text{SCOMP}(\Omega)}$.*
2. *For any term A occurring in Ω, either A is semantically equivalent to $\mathbf{0}$ or there exists a unique subset $\{C_1, \ldots, C_k\}$ of $\text{COMP}(\Omega)$ such that the term $C_1 \cup \ldots \cup C_k$ is semantically equivalent to A in all models.*

It is easy to see that every term A appearing in a degenerate sequence Ω is semantically equivalent to either $\mathbf{0}$ or $-\mathbf{0}$.

Thus, for any term A in any sequence Ω, by the *normal form $n(A)$ of A* (with respect to Ω) we shall understand either the sequence $\mathbf{0}$ or $-\mathbf{0}$, if A is semantically equivalent to one of the above, or else

$$n(A) \stackrel{\text{def}}{=} C_1 \cup \ldots \cup C_k$$

where $\{C_1, \ldots, C_k\}$ is a set of components whose union is semantically equivalent to A as provided by Lemma 5.1(2).

The sequence obtained from any sequence Ω (degenerate or not) by replacing every term in Ω by its normal form with respect to Ω is denoted by $n(\Omega)$, and called *the normal form of* Ω. The algorithm for obtaining $n(\Omega)$ is just a simple modification of the standard algorithm of transformation into complete disjunctive normal form.

By Lemma 5.1(2), the interpretation of $n(\Omega)$ in any model coincides with the interpretation of Ω. Hence, our deduction system will only be concerned with sequences in normal form.

6 DRS: a deduction system for signed formulae in Rasiowa – Sikorski style

The type of deduction system we are going to develop was introduced by H. Rasiowa and R. Sikorski in [13], and is often called an *R–S system*[3]. It operates on sequences of

[3]For more details on R-S systems and their applications, see [9]

(signed) formulae, and consists of decomposition rules for such sequences, and of axiomatic sequences to be defined below. In a more common terminology, the decomposition rules are the “inference rules” of the system, whereas the axiomatic sequences represent its “axioms”, or rather axiom schemes. Using decomposition rules, we construct a decomposition tree with vertices labelled by sequences of formulae whose branches terminate “correctly” only if we encounter a simple, axiomatic sequence of formulae – like $\mathbf{T}(F), \mathbf{N}(F)$, which is guaranteed to be valid. The decomposition tree is called a proof if it is finite and all its branches terminate correctly in the above sense.

A sequence $\Omega = G_1, G_2, \ldots, G_n$ of signed formulae is *satisfied in a model M*, written $M \models \Omega$, iff $M \models G_i$ for some $1 \leq i \leq n$. Ω is *valid* iff $M \models \Omega$ for every model M. A decomposition rule is an n+1-tuple $\Omega_0, \Omega_1, \ldots, \Omega_n, n \geq 1$, of sequences of formulae in SFORM, usually written as

$$\frac{\Omega_0}{\Omega_1 \mid \Omega_2 \mid \ldots \mid \Omega_n}$$

respectively. Ω_0 is called the *conclusion* of the rule, and $\Omega_1, \ldots, \Omega_n$ its *premises*. A rule is said to be *sound* provided its conclusion is valid iff all its premises are valid.

Thus, a decomposition rule is sound iff it leads from valid sequences to valid sequences both “downwards” and “upwards”. Such a "two-way" notion of soundness, crucial for R-S systems, is stronger than the usual one, and amounts to invertibility of rules in the common terminology. To underline this fact, we separate the premises from the conclusion in the decomposition rules by a double line instead of the usual single one.

It should be stressed that in our deduction system, the decomposition of a sequence of formulae will be preceded by transforming it to normal form; in other words, all decomposition rules will be applied to sequences in normal form only.

A sequence Ω of formulae is called *axiomatic* if it contains one of the formulae or sequences of formulae given in the table below:

(i)	$\mathscr{D}_{O'}$	(ii)	$\mathscr{E}_{\mathbf{0}}$	(iii)	$\mathscr{N}_{-\mathbf{0}}$
(iv)	$\mathscr{D}_F, \mathscr{U}_F$	(v)	$\mathscr{N}_A, \mathscr{E}_A$	(vi)	$\mathbf{T}(x = x)$
(vii)	$\mathscr{E}_{\mathbf{a} \cap S}, \mathscr{E}_{\mathbf{a} \cap S'}$	$(viii)$	$\mathbf{T}(x \overline{D_A} x), \mathbf{T}(x \overline{E_A} x), \mathscr{E}_A$		
(ix)	$\mathbf{T}(x \overline{Z_A} x), \mathbf{T}(x Z_A y), \mathscr{E}_A$	(x)	$\mathbf{T}(x \overline{Z_A} y), \mathbf{T}(x Z_A y), \mathscr{E}_A$		
(xi)	$\mathbf{T}(x \in O), \mathbf{T}(x \notin O)$	(xii)	$\mathbf{T}(x = y), \mathbf{T}(x \neq y)$		

(6.1)

where $x, y \in$ VARO, $O \in$ VARSO, $O' \in$ VARO $\cup$ VARSO, $\mathbf{a} \in$ CONA, $S, S' \in$ SCOMP, $S \neq S', Z \in \{D, E\}$.

Lemma 6.1. *The signed formulae and sequences of signed formulae (i) – (xii) are valid.*

Proof. The validity of (i) follows from (4.5) and fact that $\sigma_M(O) \subset U$ for $O \in$ VARSO. Since $\tau_M(\mathbf{0}) = \emptyset$ and $\tau_M(-\mathbf{0}) = \text{ATTR} \neq \emptyset$, then (ii) and (iii) are again valid by (4.5). Furthermore, one can easily see that (4.5) also implies the validity of (iv) and (v), while the validity of (vi) is a simple consequence of the fact that by (4.1) we have $\sigma_M(x = x) = \sigma_M(\neg x \vee x) = (U \setminus \{v(x)\}) \cup \{v(x)\} = U$.

For the proof of (vii), consider an arbitrary model M. Since S, S' are two different subcomponents, then, by Lemma 5.1, the sets $\tau_M(S)$ and $\tau_M(S')$ are disjoint. Now, $\tau_M(\mathbf{a}) = \{v(\mathbf{a})\}$ is a singleton, and it follows that either $\tau_M(\mathbf{a}) \cap \tau_M(S) = \emptyset$ or $\tau_M(\mathbf{a_j}) \cap \tau_M(S') = \emptyset$. In other words, either $\tau_M(\mathbf{a} \cap S) = \emptyset$ or $\tau_M(\mathbf{a} \cap S') = \emptyset$. Thus, by (4.5), one of the formulae in (vii) must be satisfied in M.

As to (viii), it is surely satisfied in a model M if $M \models \mathscr{E}_A$. If $M \not\models \mathscr{E}_A$, then $\tau_M(A) \neq \emptyset$ by (4.5), and hence$\langle A \rangle_Z x \neq \perp$ for $Z \in \{D, E\}$. Thus, in view of (4.1), either $M \models \mathbf{T}(z \in \langle A \rangle_Z x)$ or $M \models \mathbf{T}(z \notin \langle A \rangle_Z x)$. As by (3.7) we have either $M \not\models \mathbf{T}(x\, D_A\, x)$ or $M \not\models \mathbf{T}(x\, E_A\, x)$, then obviously either $\mathbf{T}(x\, \overline{D_A}\, x)$ or $\mathbf{T}(x\, \overline{E_A}\, x)$ holds in M.

For (ix), note again that the sequence is satisfied if $M \models \mathscr{E}_A$. Otherwise $\tau_M(A) \neq \emptyset$, whence $\langle A \rangle_Z\, x$ is defined, and $M \not\models \mathbf{T}(x\, \overline{Z_A}\, x)$ implies $M \models \mathbf{T}(x\, Z_A\, x)$. By (3.6), this yields $M \models \mathbf{T}(x\, Z_A\, y)$ for any y, whence (vii) holds in M.

In case of (x), the sequence is again satisfied if $M \models \mathscr{E}_A$. If the latter is not true, $\tau_M(A) \neq \emptyset$ and $\langle A \rangle_Z\, y$ is defined. Hence either $\mathbf{T}(x\, \overline{Z_A}\, y)$ or $\mathbf{T}(x\, Z_A\, y)$ must be satisfied.

In turn, the validity of (xi) follows from the fact that, for any $O \in$ VARSO, $\sigma_M(O) \subset U$ for any model $M = < F, v >$, whence either $v(x) \in \sigma_M(O)$ and $M \models \mathbf{T}(x \in O)$, or $v(x) \in U \setminus \sigma_M(O)$ and $M \models \mathbf{T}(x \notin O)$. Finally, (xii) is valid because either $v(x) = v(y)$ or $v(x) \neq v(y)$. □

As any sequence of formulae containing a valid formula is valid, we have

Corollary 6.2. *Every axiomatic sequence is valid.*

To define the decomposition rules, we need the notion of an indecomposable signed formula or a sequence of such formulae. Intuitively, this an elementary formula (sequence) which cannot be broken into any simpler formulae using the decomposition rules.

A formula $G \in$ SFORM is said to be *indecomposable* iff it has one of the following forms:

$$(i) \quad \mathbf{T}(x \in O), \mathbf{T}(x \notin O) \qquad (ii) \quad \mathbf{T}(x = y), \mathbf{T}(x \neq y)$$

$$(iii) \quad \mathbf{T}(x\, Z_C\, y), \mathbf{T}(x\, \overline{Z_C}\, y) \qquad (iv) \quad \mathscr{E}_C, \mathscr{N}_C, \mathscr{D}_O, \mathscr{U}_O$$

where $x, y \in$ VARO, $O \in$ VARO $\cup$ VARSO, $Z \in \{D, E\}$, $C \in [\{\mathbf{0}, -\mathbf{0}\} \cup \text{COMP}]$.

Otherwise a formula is said to be *decomposable*. Note that (iii) is justified by the fact that in decomposing sequences into normal form we split the unions of components into individual components which are considered as "atomic"; clearly, $\mathbf{0}$ is also "indecomposable". A sequence Ω of signed formulae is called *indecomposable* iff all its elements are indecomposable.

The general idea of an R-S deduction system is to break down the formulae occurring in a sequence of (signed) formulae into sequences of elementary, indecomposable formulae, whose validity is equivalent to the validity of the original sequence. The decomposition rules present in the system can be divided into *replacement rules* and *expansion rules*. Roughly speaking, replacement rules replace the original formula they act upon with simpler formulae (or at least extract such formulae from it), whereas expansion rules only add new formulae to the sequence under decomposition, e.g. to close it under some symmetry or transitivity law. In consequence:

- Replacement rules are only applicable to decomposable sequences.
- Expansion rules can also be applied to indecomposable sequences.

DRS: Decomposition rules for signed formulae

Replacement rules:

(**T**) $$\frac{\Omega', \mathbf{T}(F), \Omega''}{\Omega', \mathbf{T}(x \in F), \Omega''}$$ (**N**) $$\frac{\Omega', \mathbf{N}(F), \Omega''}{\Omega', \mathbf{N}(y \in F), \Omega'', \mathbf{N}(F)}$$

where F is not of the form $z \in G$ for $z \in \textsc{Varo}, G \in FORM$,
x does not occur above the double line, and y is any variable in $\textsc{Varo}$,

($\mathbf{N} \in$) $$\frac{\Omega', \mathbf{N}(x \in F), \Omega''}{\Omega', \mathscr{U}_F, \mathbf{T}(x \notin F), \Omega''}$$ ($\mathbf{N} \notin$) $$\frac{\Omega', \mathbf{N}(x \notin F), \Omega''}{\Omega', \mathscr{U}_F, \mathbf{T}(x \in F), \Omega''}$$

($\mathbf{T} \notin \neg$) $$\frac{\Omega', \mathbf{T}(x \notin \neg F), \Omega''}{\Omega', \mathbf{T}(x \in F), \Omega''}$$ ($\mathbf{T} \in \wedge$) $$\frac{\Omega', \mathbf{T}(x \in F \wedge G), \Omega''}{\Omega', \mathbf{T}(x \in F), \Omega'' \mid \Omega', \mathbf{T}(x \in G), \Omega''}$$

($\mathbf{T} \notin \wedge$) $$\frac{\Omega', \mathbf{T}(x \notin (F \wedge G)), \Omega''}{\Omega', \mathbf{T}(\neg F), \mathbf{T}(\neg G), \mathscr{D}_F, \Omega'' \mid \Omega', \mathbf{T}(\neg F), \mathbf{T}(\neg G), \mathscr{D}_G, \Omega'' \mid \Omega', \mathbf{T}(x \notin F), \mathbf{T}(x \notin G), \Omega''}$$

($\mathbf{T} \in \vee$) $$\frac{\Omega', \mathbf{T}(x \in F \vee G), \Omega''}{\Omega', \mathbf{T}(F), \mathbf{T}(G), \mathscr{D}_F, \Omega'' \mid \Omega', \mathbf{T}(F), \mathbf{T}(G), \mathscr{D}_G, \Omega'' \mid \Omega', \mathbf{T}(x \in F), \mathbf{T}(x \in G), \Omega''}$$

($\mathbf{T} \notin \vee$) $$\frac{\Omega', \mathbf{T}(x \notin (F \vee G)), \Omega''}{\Omega', \mathbf{T}(x \notin F), \Omega'' \mid \Omega', \mathbf{T}(x \notin G), \Omega''}$$

($\mathbf{T} \in \langle A \rangle$) $$\frac{\Omega', \mathbf{T}(x \in \langle A \rangle_Z F), \Omega''}{\Omega', \mathbf{T}(y \in F), \Omega'', \mathbf{T}(x \in \langle A \rangle_Z F) \mid \Omega', \mathbf{T}(x Z_A y), \Omega'', \mathbf{T}(x \in \langle A \rangle_Z F)}$$

where y is an arbitrary variable in $\textsc{Varo}$

($\mathbf{T} \notin \langle A \rangle$) $$\frac{\Omega', \mathbf{T}(x \notin \langle A \rangle_Z F), \Omega''}{\Omega', \mathbf{T}(x \overline{Z_A} z), \mathbf{T}(z \notin F), \Omega''}$$ ($\mathbf{T} \in [A]$) $$\frac{\Omega', \mathbf{T}(x \in [A]_Z F), \Omega''}{\Omega', \mathbf{T}(x \overline{Z_A} z), \mathbf{T}(z \in F), \Omega''}$$

where $z \in \textsc{Varo}$, and z does not occur above the double line,

($\mathbf{T} \notin [A]$) $$\frac{\Omega', \mathbf{T}(x \notin [A]_Z F), \Omega''}{\Omega', \mathbf{T}(y \notin F), \Omega'', \mathbf{T}(x \notin [A]_Z F) \mid \Omega', \mathbf{T}(x Z_A y), \Omega'', \mathbf{T}(x \notin [A]_Z F)}$$

where y is an arbitrary variable in $\textsc{Varo}$,

($\mathscr{D} \neg$) $$\frac{\Omega', \mathscr{D}_{\neg F}, \Omega''}{\Omega', \mathscr{D}_F, \Omega''}$$ ($\mathscr{U} \neg$) $$\frac{\Omega', \mathscr{U}_{\neg F}, \Omega''}{\Omega', \mathscr{U}_F, \Omega''}$$

($\mathscr{D} \vee$) $$\frac{\Omega', \mathscr{D}_{F \vee G}, \Omega''}{\Omega', \mathbf{T}(F), \mathbf{T}(G), \mathscr{D}_F, \Omega'' \mid \Omega', \mathbf{T}(F), \mathbf{T}(G), \mathscr{D}_G, \Omega''}$$

($\mathscr{D} \wedge$) $$\frac{\Omega', \mathscr{D}_{F \wedge G}, \Omega''}{\Omega', \mathbf{T}(\neg F), \mathbf{T}(\neg G), \mathscr{D}_F, \Omega'' \mid \Omega', \mathbf{T}(\neg F), \mathbf{T}(\neg G), \mathscr{D}_G, \Omega''}$$

$$(\mathscr{U}\vee)\qquad \frac{\Omega', \mathscr{U}_{F\vee G}, \Omega''}{\Omega', \mathbf{N}(F), \Omega'' \mid \Omega', \mathbf{N}(G), \Omega'' \mid \Omega', \mathscr{U}_F, \mathscr{U}_G, \Omega''}$$

$$(\mathscr{U}\wedge)\qquad \frac{\Omega', \mathscr{U}_{F\wedge G}, \Omega''}{\Omega', \mathbf{N}(\neg F), \Omega'' \mid \Omega', \mathbf{N}(\neg G), \Omega'' \mid \Omega', \mathscr{U}_F, \mathscr{U}_G, \Omega''}$$

$$(\mathscr{D} \mid \cdot \mid)\qquad \frac{\Omega', \mathscr{D}_{|A|_Z F}, \Omega''}{\Omega', \mathscr{D}_F, \Omega'' \mid \Omega', \mathscr{N}_A, \Omega''} \qquad\qquad (\mathscr{U} \mid \cdot \mid)\qquad \frac{\Omega', \mathscr{U}_{|A|_Z F}, \Omega''}{\Omega', \mathscr{U}_F, \mathscr{E}_A, \Omega''}$$

$$(\mathbf{T}\, Z_{A\cup B})\qquad \frac{\Omega', \mathbf{T}(x Z_{A\cup B}\, y), \Omega''}{\Omega', \mathbf{T}(x Z_A\, y), \Omega'' \mid \Omega', \mathbf{T}(x Z_B\, y), \Omega''}$$

$$(\mathbf{T}\, \overline{Z_{A\cup B}})\qquad \frac{\Omega', \mathbf{T}(x \overline{Z_{A\cup B}}\, y), \Omega''}{\Omega', \mathbf{T}(x \overline{Z_A}\, y), \mathbf{T}(x \overline{Z_B}\, y), \Omega''}$$

Expansion rules:

$$(\text{sym } =)\qquad \frac{\Omega', \mathbf{T}(x \neq y), \Omega''}{\Omega', \mathbf{T}(x \neq y), \mathbf{T}(y \neq x), \Omega''} \qquad\qquad (\text{sub} =)\qquad \frac{\Omega', \mathbf{T}(x \neq y), \alpha(x), \Omega''}{\Omega', \mathbf{T}(x \neq y), \alpha(x), \alpha(y), \Omega''}$$

$$(\text{sym Z})\qquad \frac{\Omega', \mathbf{T}(x \overline{Z_C}\, y), \Omega''}{\Omega', \mathbf{T}(x \overline{Z_C}\, y), \mathbf{T}(y \overline{Z_C}\, x), \Omega''} \qquad\qquad (xZx)\qquad \frac{\Omega', \mathbf{T}(x \overline{Z_C}\, x), \Omega''}{\Omega', \mathbf{T}(x \overline{Z_C}\, x), \mathbf{T}(x \overline{Z_C}\, y), \Omega''}$$

$$(\text{tran } =)\qquad \frac{\Omega', \mathbf{N}(x = y), \Omega'', \mathbf{N}(y \in F), \Omega'''}{\Omega', \Omega'', \Omega''', \mathbf{N}(x = y), \mathbf{N}(y \in F), \mathbf{N}(x \in F)}$$

$$(EDE)\qquad \frac{\Omega', \mathbf{T}(x \overline{E_C}\, y), \mathbf{T}(y \overline{D_C}\, z), \mathbf{T}(z \overline{E_C}\, u), \Omega''}{\Omega', \mathbf{T}(x \overline{E_C}\, y), \mathbf{T}(y \overline{D_C}\, z), \mathbf{T}(z \overline{E_C}\, u), \mathbf{T}(x \overline{E_C}\, u), \Omega''}$$

$$(DED)\qquad \frac{\Omega', \mathbf{T}(x \overline{D_C}\, y), \mathbf{T}(y \overline{E_C}\, z), \mathbf{T}(z \overline{D_C}\, u), \Omega''}{\Omega', \mathbf{T}(x \overline{D_C}\, y), \mathbf{T}(y \overline{E_C}\, z), \mathbf{T}(z \overline{D_C}\, u), \mathbf{T}(x \overline{D_C}\, u), \Omega''}$$

$$(=_C)\qquad \frac{\Omega', \mathbf{T}(x \overline{D_C}\, y), \mathbf{T}(x \overline{D_C}\, z), \mathbf{T}(x \overline{E_C}\, y), \mathbf{T}(x \overline{E_C}\, z), \mathbf{T}(y \overline{Z_C}\, u), \Omega''}{\Omega', \mathbf{T}(x \overline{D_C}\, y), \mathbf{T}(x \overline{D_C}\, z), \mathbf{T}(x \overline{E_C}\, y), \mathbf{T}(x \overline{E_C}\, z), \mathbf{T}(y \overline{Z_C}\, u), \mathbf{T}(z \overline{Z_C}\, u), \Omega''}$$

In all DRS rules, $x, y, z, u \in \textsc{Varo}, F, G \in \textsc{Form}, Z \in \{D, E\}$, Ω' and Ω" are sequences of signed formulae, with the prerequisite that Ω' is indecomposable, $A, B \in \textsc{Term}$, C is a component, and α is a signed formula. Since the decomposition process is applied to sequences in normal form only, the only terms encountered in the actually decomposed formulae will be $\mathbf{0}, -\mathbf{0}$ and unions of components; thus we need no rules for term constructors other than the union. Finally, $|\cdot|$ denotes any of the modalities $[\cdot], \langle\cdot\rangle$.

The rules (**T**) and (**N**) allow us to base our deduction system on membership formulae of the form $\mathbf{T}(x \in F)$, $\mathbf{N}(x \in F)$. They are analogous to those used for quantifiers in first-order logic, as the semantic conditions for $\mathbf{T}(F), \mathbf{N}(F)$ to be true involve implicit quantification over all objects, universal in case of $\mathbf{T}(F)$, and existential in case of

$\mathbf{N}(F)$. In turn, rules ($\mathbf{N}\in$) and ($\mathbf{N}\notin$) allow us to base the system on the ($\mathbf{T}$) type of formulae only, with the $\mathscr{N}$ constructor hidden away in the $\mathscr{U}, \mathscr{E}$ predicates expressing undefinedness of formulae and emptiness of attribute sets, respectively.

One can easily check, using the definition of semantics and reasoning similarly as we did in case of the axiomatic sequences, that the rules given above are indeed sound in the strong, two-way sense defined above. Thus we have:

Lemma 6.3. *The decomposition rules in the (DRS) system are sound.*

7 Decomposition trees for sequences of formulae and completeness of the DRS system

In this Section we shall describe the mechanism of using the DRS system introduced in the preceding section to prove the validity of sequences of signed formulae, and show the completeness of the system.

The validity proofs in DRS consist in constructing decomposition trees for sequences of formulae, using the decomposition rules in the way described below. We shall prove that a sequence is valid iff its decomposition tree is finite and all its branches end in axiomatic sequences only; this amounts to the completeness of the system.

By a *decomposition tree* for a *sequence* Ω of signed formulae we mean any maximal binary tree $DT(\Omega)$ with vertices labelled by sequences of signed formulae defined inductively as follows:

(i) The root of $DT(\Omega)$ is labelled by $n(\Omega)$, i.e. the normal form of Ω.

(ii) Let v, labelled by Σ, be an end node of a branch B of the tree constructed up to now. Then,

  (a) We terminate the branch B at node v if either:

    (a1) Σ is an axiomatic sequence, or

    (a2) Σ is indecomposable and no expansion rule is applicable to Σ;

  (b) Otherwise we expand the branch B beyond v by attaching to this node i descendants labelled with $\Sigma_1, \ldots, \Sigma_i$, if $\dfrac{\Sigma}{\Sigma_1 \mid \ldots \mid \Sigma_i}$ is a rule in *DRS* applicable to Σ.

The decomposition tree starts with a single node, labelled by the normal form of Ω. The initial node is then expanded into a tree by means of the rules in DRS. In case (a1) there is no sense to extend branch B any further, since we already have an axiomatic sequence at its end. In case (a2) no replacement rule is applicable to Σ, and we cannot augment this sequence by applying an expansion rule, whence branch B cannot be extended beyond node v. Otherwise the branch is expanded by means of some applicable rule.

It is easy to see that $DT(\Omega)$ may be infinite owing to the rules $(\mathbf{N}), (\mathbf{T} \in \langle A \rangle), (\mathbf{N} \notin \langle A \rangle), (\mathbf{T} \notin [A]), (\mathbf{N} \in [A])$ rules. Furthermore, a node of $DT(\Omega)$ is terminal (a leaf), iff its label Σ is either a axiomatic sequence, or an indecomposable sequence closed under all the expansion rules[4].

In the sequel, we will refer to sequences of signed formulae labelling the terminal nodes of $DT(\Omega)$ as the *terminal sequences* of Ω.

The notion of provability in our system is as follows:

- A decomposition tree $DT(\Omega)$ is said to be *proof* if it is finite and all its terminal sequences are axiomatic
- A sequence Ω of signed formulae is called *provable* in DRS, written as $DRS \vdash \Omega$, iff it has a decomposition tree $DT(\Omega)$ which is a proof.

From the equivalent (invertible) character of the rules, it is evident that

Lemma 7.1. *The system DRS is sound, i.e. every provable sequence Ω of signed formulae is valid.*

The cornerstone of the converse result – the completeness theorem — is the following crucial

Lemma 7.2. *For any sequence Ω of signed formulae, each valid terminal sequence Σ in $DT(\Omega)$ is axiomatic.*

Proof. Let Ω and Σ satisfy the assumptions of the Lemma, and assume that Σ is not axiomatic. Since Σ is a terminal sequence of $DT(\Omega)$, it follows that Σ is indecomposable and closed under all expansion rules. Hence, by the definition of an indecomposable sequence, each element of Σ must have one of the following forms:

$$\mathbf{T}(x \in O), \mathbf{T}(x \notin O), \mathbf{T}(x = y), \mathbf{T}(x \neq y), \mathbf{T}(x Z_C y), \mathbf{T}(x \overline{Z_C} y), \mathscr{E}_C, \mathscr{N}_C, \mathscr{U}_O$$

where $x, y \in \text{VARO}$, $O \in \text{VARO} \cup \text{VARSO}, Z \in \{D, E\}, C \in (\{\mathbf{0}, -\mathbf{0}\} \cup \text{COMP})$, see the definition on page 15. Note that by (6.1) Σ cannot contain $\mathscr{D}_O$, since it is not axiomatic. Assume $\text{CONA}(\Omega)$ is the set of all attribute constants that occur in Ω, while

$$\text{VARSA}(\Omega) = \{Q_1, \ldots, Q_m\}$$

is the set of all $Q \in \text{VARSA}$ that occur in Ω. Then the terms C which can appear in Σ are as follows:

1. $C = \mathbf{0}$.

[4] We say that Σ is closed under an expansion rule R if either R is not applicable to Σ or its application cannot add any new formulae to that sequence.

2. $C = -\mathbf{0}$. Since the normal form of any term A with respect to a non–degenerate sequence Ω is different from $-\mathbf{0}$ by Lemma 5.1, this case is only possible if Ω is degenerate, i.e. if all the terms in Ω are Boolean combinations of $\mathbf{0}$s,

3. $C = \mathbf{a} \cap S$. where $a \in \textsc{Cona}(\Omega)$ and $S \in \textsc{Scomp}(\Omega)$ is a subcomponent of the form
$$S = \{Q_1^{i_1} \cap \ldots \cap Q_m^{i_m} : i_1, \ldots, i_m \in \{+,-\}\},$$
with $A^+ = A$, $A^- = -A$ for any term A.

We are going to construct a counterexample to Σ, i.e. a model in which Σ is not true.

Let $\rho \subseteq \textsc{Varo} \times \textsc{Varo}$ be a binary relation defined by

$$\rho(x,y) \text{ iff the formula } \mathbf{T}(x \neq y) \text{ occurs in the sequence } \Sigma.$$

Then ρ is both symmetric and transitive, because Σ is closed under the $(sym =)$ and $(tran =)$ rules. Hence the relation

$$\rho^* = \rho \cup \{(x,x) : x \in \textsc{Varo}\}$$

is an equivalence relation on $\textsc{Varo}$. As our counter-example we take a modified Herbrand-type model of the form

$$H = \langle F, v \rangle,$$

with
$$F = \langle U, \{D_A\}_{A \in 2^+\textsc{Attr}}, \{E_A\}_{A \in 2^+\textsc{Attr}} \rangle$$
where
$$U = \textsc{Varo}/\rho^*$$
is the set of all equivalence classes of ρ^*.

The multi-sorted valuation v is defined as follows:

1. For all $x \in \textsc{Varo}$,
$$v(x) = [x]_{\rho^*},$$
where $[x]_{\rho^*}$ is the equivalence class of ρ^* which contains x. By definition of ρ, we have
$$v(x) = v(y) \text{ iff the formula } \mathbf{T}(x \neq y) \text{ occurs in } \Sigma \tag{7.1}$$
for all $x, y \in \textsc{Varo}, x \neq y$.

2. For all $O \in \textsc{Varso}$,
$$v(O) = \{v(x) : \mathbf{T}(x \notin O) \text{ occurs in } \Sigma\}.$$

3. To define the valuation of the variables ranging over sets of attributes,recall that

$$\textsc{VarsA}(\Omega) = \{Q_1, \ldots, Q_m\}.$$

By clause (vii) of the definition of an axiomatic sequence, any sequence containing both $\mathscr{E}_{\mathbf{a}\cap S}$ and $\mathscr{E}_{\mathbf{a}\cap S'}$, where $S, S' \in \textsc{SComp}, S \neq S'$, is axiomatic. Since, by our assumption, Σ is not axiomatic, it follows that for any $\mathbf{a} \in \textsc{ConA}(\Omega)$, there is at most one $S \in \textsc{SComp}(\Omega)$ such that the formula $\mathscr{E}_{\mathbf{a}\cap S}$ occurs in Σ. If such an S exists for a given $\mathbf{a}$, we denote it by $S_{\mathbf{a}}$ and say that $\mathbf{a}$ is *positive in* Σ. We put $v(Q) = \emptyset$ for any $Q \in \textsc{VarsA} \setminus \{Q_1, \ldots, Q_m\}$ and

$$v(Q_k) \quad = \{\mathbf{a} : \mathbf{a} \text{ positive in } \Sigma \text{ and } Q_k^+ \text{ occurs in } S_{\mathbf{a}}\}$$

One can show that for the interpretation of terms τ_H induced by the valuation v

$$\tau_H(\mathbf{a} \cap S) = \begin{cases} \{\mathbf{a}\}, & \text{if } \mathbf{a} \text{ is positive in } \Sigma \text{ and } S = S_{\mathbf{a}}, \\ \emptyset, & \text{otherwise.} \end{cases} \tag{7.2}$$

Thus, τ_H evaluates all components occurring in Σ to either singletons or $\emptyset$. This is the key property which helps us to define the families $\{D_A\}_{A\in 2^+\textsc{Attr}}, \{E_A\}_{A\in 2^+\textsc{Attr}}$ of disagreement and exhaustiveness relations on $U \times U$, which is the last step needed to complete the definition of the model. Due to property (3.4), disagreement and exhaustiveness relations corresponding to individual attributes in $\textsc{Attr}$ generate the whole family $\{D_A\}_{A\in 2^+\textsc{Attr}}, \{E_A\}_{A\in 2^+\textsc{Attr}}$; they can also be defined independently of each other, which solves the basic technical problem connected with modalities parametrized by arbitrary sets of attributes.

We begin with defining the families $\{Z_a\}_{a\in\textsc{Attr}}, Z \in \{D, E\}$, of disagreement and exhaustiveness relations with respect to individual attributes. There are two cases:

Case 1: Σ contains the term $-\mathbf{0}$; this can happen only if Ω is degenerate. Hence in this case, for any formula of the form either $\mathbf{T}(x\, Z_C\, y)$ or $\mathbf{T}(x\, \overline{Z_C}\, y)$ occurring in Σ we have either $C = \mathbf{0}$ or $C = -\mathbf{0}$. We put

$$Z_a = \{(v(x), v(y)) \mid \mathbf{T}(x\, \overline{Z_{-\mathbf{0}}}\, y) \text{ is in } \Sigma\}$$

for each $a \in \textsc{Attr}, Z \in \{D, E\}$.

Case 2. Σ does not contain $-\mathbf{0}$; then, for any formula of the form $\mathbf{T}(x Z_C\, y)$ or $\mathbf{T}(x\overline{Z_C}\, y)$ occurring in Σ, the term C is either a component or $\mathbf{0}$. In the latter case, $C = \mathbf{a} \cap S$, where $\mathbf{a} \in \textsc{ConA}(\Omega)$. We put:

$$Z_a = \begin{cases} \{(v(x), v(y)) \mid \mathbf{T}(x\, \overline{Z_{\mathbf{a}\cap S_{\mathbf{a}}}}\, y) \text{ is in } \Sigma\} & \text{if } \mathbf{a} \text{ is positive in } \Sigma \text{ and } S = S_{\mathbf{a}}, \\ \emptyset & \text{otherwise.} \end{cases} \tag{7.3}$$

Having defined the singleton-based relations, we set

$$Z_A = \bigcap_{a\in A} Z_a \tag{7.4}$$

for any $Z \in \{D,E\}$ and all $\emptyset \neq A \subseteq \textsc{Attr}$.

We shall show that the families $\{D_A\}_{A\in 2^+\textsc{Attr}}, \{E_A\}_{A\in 2^+\textsc{Attr}}$ defined above are families of disagreement and exhaustiveness relations which satisfy conditions (3.3) – (3.8). The degenerate Case 1 is clear, and thus, we will only show Case 2.

Condition (3.4) follows immediately from definition (7.4) above. Before proving the other conditions, let us first remark that, for any $\mathbf{a}$ positive in Σ and any $x,y \in U$, we have

$$[x]\, Z_a\, [y] \quad \text{iff} \quad \mathbf{T}(x\,\overline{Z_{C_\mathbf{a}}}\,y) \text{ is in } \Sigma \tag{7.5}$$

where $C_\mathbf{a} = \mathbf{a} \cap S_\mathbf{a}$.

Indeed: the backward implication follows directly from the definition of Z_a in (7.3). Conversely, if $[x]\, Z_a\, [y]$, then $\mathbf{a}$ is positive in Σ, and by (7.3) there exist $x', y' \in \textsc{VarO}$ such that $v(x') = [x], v(y') = [y]$ and $\mathbf{T}(x'\,\overline{Z_{C_\mathbf{a}}}\,y')$ is in Σ, with $C_\mathbf{a}$ as above. Then by the definition of v we have either $x' = x$ or $\mathbf{T}(x' \neq x)$ is in Σ, and either $y' = y$ or $\mathbf{T}(y' \neq y)$ is in Σ. Since Σ is closed under rule (sub =), this implies (7.5) holds.

It can be easily seen that (7.5) implies the symmetry condition (3.3), since $\mathbf{T}(x\,\overline{Z_{C_\mathbf{a}}}\,y) \in \Sigma$ yields $\mathbf{T}(y\,\overline{Z_{C_\mathbf{a}}}\,x) \in \Sigma$ in view of Σ being closed under rule $(sym\ =)$.

$$\mathbf{T}(x\,\overline{D_{C_\mathbf{a}}}\,y), \mathbf{T}(y\,\overline{E_{C_\mathbf{a}}}\,z), \mathbf{T}(z\,\overline{D_{C_\mathbf{a}}}\,u)$$

are all in Σ. Using rule (DED), we conclude that formula $\mathbf{T}(x\,\overline{D_{C_\mathbf{a}}}\,u)$ is also in Σ, whence $[x]\, D_a\, [u]$ by (7.5). The second implication in condition (3.5) is proved in a quite analogous way, but using rule (EDE)

It is easy to see that condition (3.6) follows from (7.5) and rule (xZx), since by the latter rule $\mathbf{T}(x\,\overline{Z_{C_\mathbf{a}}}\,y) \in \Sigma$ whenever $\mathbf{T}(x\,\overline{Z_{C_\mathbf{a}}}\,x) \in \Sigma$.

As to condition (3.7), suppose both $[x]\, D_a\, [x]$ and $[x]\, E_a\, [x]$. Then by (7.5) both $\mathbf{T}(x\,\overline{D_{C_\mathbf{a}}}\,x)$ and $\mathbf{T}(x\,\overline{E_{C_\mathbf{a}}}\,x)$ are in Σ. Since Σ is not axiomatic, then by (viii) of the definition of an axiomatic sequence Σ cannot contain $\mathscr{E}_{C_\mathbf{a}}$. However, this implies $\mathbf{a}$ is not positive in Σ, whence $Z_a = \emptyset$, which is a contradiction. Thus H satisfies (3.7).

Finally, to show condition (3.8), assume $[x]\, D_a\, [y], [x]\, D_a\, [z], [x]\, E_a\, [y], [x]\, E_a\, [z]$ all hold. Then by (7.5) the formulae $\mathbf{T}(x\,\overline{D_{C_\mathbf{a}}}\,y), \mathbf{T}(x\,\overline{D_{C_\mathbf{a}}}\,z), \mathbf{T}(x\,\overline{E_{C_\mathbf{a}}}\,y), \mathbf{T}(x\,\overline{E_{C_\mathbf{a}}}\,z)$ are in Σ. If now $[y]\, Z_a\, [u]$ holds, where $Z \in \{D,E\}$, then $\mathbf{T}(y\,\overline{Z_{C_\mathbf{a}}}\,u)$ is in Σ by (7.5), whence by rule $(=_\mathbf{C})$ we get that $\mathbf{T}(z\,\overline{Z_{C_\mathbf{a}}}\,u)$ is in Σ too, which yields $[z]\, Z_a\, [u]$ by (7.5). The converse implications in both the equivalences in (3.8) are proved by symmetry.

Thus, $\mathscr{F} = \langle U, \{D_A\}_{A\in 2^+\textsc{Attr}}, \{E_A\}_{A\in 2^+\textsc{Attr}} \rangle$ is a disagreement and exhaustiveness frame as defined in (3.2), and $H = \langle F, v \rangle$ is model of our language.

The proof that $H \not\models \Sigma$ is by considering each type of signed formula that can occur in Σ, i.e.

$$\mathbf{T}(x \in O), \mathbf{T}(x \notin O), \mathbf{T}(x = y), \mathbf{T}(x \neq y), \mathbf{T}(x Z_C\, y), \mathbf{T}(x\,\overline{Z_C}\,y), \mathscr{E}_C, \mathscr{N}_C, \mathscr{U}_O$$

where $x, y \in \text{VARO}, O \in \text{VARO} \cup \text{VARSO}$, C is either a component or $C \in \{\mathbf{0}, -\mathbf{0}\}$, and proving that it cannot be true in H.

We shall prove these cases one by one, assuming C is a component; the special case when $C \in \{\mathbf{0}, -\mathbf{0}\}$ can be handled in a similar way.

Let us begin with the $\mathscr{E}_C, \mathscr{N}_C$ and $\mathscr{U}_O$ type of formulae. As $\sigma_H(O) \subset U$ for $O \in \text{VARO} \cup \text{VARSO}$, then $H \not\models \mathscr{U}_O$. If $\mathscr{E}_C$ is in Σ for some $C \in \text{COMP}$, then $C = \mathbf{a} \cap S$ for some $\mathbf{a} \in \text{CONA}(\Omega), S \in \text{SCOMP}$. What is more, in this case $\mathbf{a}$ is positive in Σ and $S = S_{\mathbf{a}}$. Thus by (7.2) we have $\tau_H(C) = \{\mathbf{a}\} \neq \emptyset$, whence $H \not\models \mathscr{E}_C$. In turn, if $\mathscr{N}_C$ is in Σ, then $\mathscr{E}_C$ cannot be in Σ, since Σ is not axiomatic. Hence $C = \mathbf{a} \cap S$, where either $\mathbf{a}$ is not positive in Σ or $\mathbf{a}$ is positive in Σ but $S \neq S_{\mathbf{a}}$. Thus $\tau_H(C) = \emptyset$ by (7.2) and $H \not\models \mathscr{N}_C$

Now let us pass to the $\mathbf{T}\notin$-type of formulae. If $\mathbf{T}(x \neq y)$ is in Σ, then, by the definition of v, we have $v(x) = v(y)$, whence $H \not\models \mathbf{T}(x \neq y)$. Similarly, if $\mathbf{T}(x \notin O)$ is in Σ, then $[x] \in v(O)$ by definition of v, whence $H \not\models \mathbf{T}(x \notin O)$. Finally, if $\mathbf{T}(x\, \overline{Z_C}\, y)$ is in Σ, then again $C = \mathbf{a} \cap S$ for some $\mathbf{a} \in \text{CONA}(\Omega), S \in \text{SCOMP}$, and we have two possibilities. If either $\mathbf{a}$ is not positive in Σ or $S \neq S_{\mathbf{a}}$, then $\tau_H(C) = \emptyset$ and $H \not\models \mathbf{T}(x\, \overline{Z_C}\, y)$. In the opposite case, we get $[x]\, Z_a\, [y]$ by (7.5), which again implies $H \not\models \mathbf{T}(x\, \overline{Z_C}\, y)$.

The final step is to consider the $\mathbf{T}$–type formulae. The main argument used here is that Σ, which is not axiomatic, cannot contain any of the signed formulae or sequences of signed formulae listed in the definition of an axiomatic sequence.

If $\mathbf{T}(x = y)$ is in Σ, then $x \neq y$ in view of the clause (vi) in the definition of an axiomatic sequence in (6.1). In addition, $\mathbf{T}(x \neq y)$ cannot occur in Σ by (xii) of the same definition. This yields $v(v) \neq v(y)$, whence $H \not\models \mathbf{T}(x = y)$. If $\mathbf{T}(x \in O)$ is in Σ, then again $\mathbf{T}(x \notin 0)$ cannot be in Σ by clause (xi) in (6.1) . Hence, by the definition of v, $v(x) \in v(O)$ only if $\mathbf{T}(x' \notin 0)$ is in Σ for some x' such that $\mathbf{T}(x \neq x')$ is in Σ. However, the latter implies $\mathbf{T}(x \notin 0)$ is in Σ by rule (**sub** $=$), which is a contradiction. Thus $v(x) \notin v(O)$, and hence $H \not\models \mathbf{T}(x \in O)$.

Finally, if $\mathbf{T}(x\, Z_C\, y)$ is in Σ, then we have two cases analogous to those for $\mathbf{T}(x\, \overline{Z_C}\, y)$ considered above. The first is when $C = \mathbf{a} \cap S$, where either $\mathbf{a}$ is not positive in Σ or $S \neq S_{\mathbf{a}}$; then $\tau_H(C) = \emptyset$ and $H \not\models \mathbf{T}(x\, Z_C\, y)$. The second is when $C = \mathbf{a} \cap S_{\mathbf{a}}$, where $\mathbf{a}$ is positive in Σ. In this case $\mathscr{E}_C$ is in Σ. In view of $\mathbf{T}(x\, Z_C\, y)$ being in Σ, and clause (x) in (6.1), this implies $\mathbf{T}(x\, \overline{Z_C}\, y)$ cannot be in Σ, whence $H \not\models \mathbf{T}(x\, Z_C\, y)$ by (7.5).

In this way, we arrive at a contradiction, whence a valid terminal sequence Σ must be axiomatic. $\square$

Now we can state the completeness theorem:

Theorem 7.3. *Every valid sequence of signed formulae $\Omega \in \text{SFORM}$ is provable, i.e. it has a finite decomposition tree whose terminal sequences are all axiomatic.*

Proof. Suppose Ω is valid, and recall that the root of $DT(\Omega)$ is labelled by $n(\Omega)$, the normal form of Ω. If $DT(\Omega)$ is finite, then from the (two-way) soundness of the decomposition rules in DRS it follows that $n(\Omega)$ is valid iff all the terminal sequences of

$DT(\Omega)$ are valid. However, by Lemma 7.2, the later holds iff all of these are axiomatic. Thus if Ω is valid and $DT(\Omega)$ is finite, then also $n(\Omega)$ is valid, and hence all the terminal sequences of $DT(\Omega)$ are axiomatic.

To complete the proof we have to prove that if $DT(\Omega)$ is infinite, then Ω cannot be valid. We achieve this by modifying the standard proof used in [13], p. 302, to suit the structure of our language. The details are analogous to those in [7], so we shall give only the basic outline of the method.

Suppose $DT(\Omega)$ is infinite. As $DT(\Omega)$ is a binary tree, it follows from König's Lemma that it has an infinite branch B starting at the root. Let us denote by Δ the set of all indecomposable formulae which occur in the sequences labelling the nodes of B. Assume that Σ is an axiomatic subsequence of Δ. Since any subsequence of Δ consists of indecomposable formulae only, and each vertex in $DT(\Omega)$ inherits all indecomposable formulae from its ancestors, there is some node v of B such that Σ is a subsequence of its label Ω_v. It follows that Ω_v is axiomatic as well, and therefore, B terminates at v. This contradicts the fact that B is infinite. Hence, Δ can contain none of the formulae or sequences of formulae (i)-(viii) from the definition of an axiomatic sequence.

Thus, reasoning exactly in the same way as in the proof of Lemma 7.2, we can build a model H such that $H \not\models G$ for every $G \in \Delta$. Indeed, the only assumptions used in building a counter-example H for a sequence Σ were that Σ was indecomposable, and that it did not contain any axiomatic sequence. Both those assumptions hold for Δ as well; the only difference is that Δ is infinite, but this is irrelevant for the construction of H.

By definition of a decomposition tree, the top node of B which coincides with the root of $DT(\Omega)$ is labelled by $n(\Omega)$. Hence, arguing by induction on the complexity of a formula, the fact that no indecomposable formula in the labels of B is true in H implies that $n(\Omega)$ is not true in H. Thus, Ω cannot be valid. Details of the argument can be found in [7].

References

[1] Andréka, H. and Németi, I. (2001). Personal communication.

[2] Beth, E. (1959). *The Foundations of Mathematics*. North–Holland, Amsterdam.

[3] Demri, S. and Orłowska, E. (1998a). Complementarity relations: reduction of decision rules and informational representability. In Polkowski, L. and Skowron, A., editors, *Rough sets in knowledge discovery, Vol. 1*, pages 99–106. Physica–Verlag, Heidelberg.

[4] Demri, S. and Orłowska, E. (1998b). Informational representability of models for information logics. In Orłowska, E., editor, *Logic at Work*, pages 383–409. Physica – Verlag, Heidelberg.

[5] Düntsch, I. and Konikowska, B. (2000). A multimodal logic for reasoning about complementarity. *J. Appl. Non-Classical Logics*, 10(3):273–301.

[6] Düntsch, I. and Orłowska, E. (2000). Logics of complementarity in information systems. *Mathematical Logic Quarterly*, 46:267–288.

[7] Konikowska, B. (1997). A logic for reasoning about similarity. *Studia Logica*, 58:185–226.

[8] Konikowska, B. (1998). A logic for reasoning about similarity. In Orłowska, E., editor, *Incomplete Information – Rough Set Analysis*, pages 462–491. Physica – Verlag, Heidelberg.

[9] Konikowska, B. (2002). Rasiowa–Sikorski deduction systems in computer science applications. *Theoretical Computer Science*, 286:323–366.

[10] Orłowska, E. (1995). Information algebras. In Alagar, V. S. and Nivat, M., editors, *Algebraic Methodology and Software Technology, 4th International Conference, AMAST '95, Montreal, Canada, Proceedings*, volume 639 of *Lecture Notes in Computer Science*, pages 50–65. Springer–Verlag.

[11] Pawlak, Z. (1981). Information systems, theoretical foundations. *Information Systems*, 6:205–218.

[12] Pawlak, Z. (1982). Rough sets. *Internat. J. Comput. Inform. Sci.*, 11:341–356.

[13] Rasiowa, H. and Sikorski, R. (1963). *The Mathematics of Metamathematics*, volume 41 of *Polska Akademia Nauk. Monografie matematyczne*. PWN, Warsaw.