A multimodal logic for reasoning about complementarity[*]

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

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

December 14, 1999

Abstract

Two objects o_1, o_2 of an information system are said to be complementary with respect to attribute a if $a(o_1) = -a(o_2)$, where $a(o)$ is the set of values of attribute a assigned to o. They are said to be complementary with respect to a set of attributes A if they are complementary with respect to each attribute $a \in A$. A multi-modal logical language for reasoning about complementarity relations is presented, with modalities $[A]$ and $\langle A \rangle$ parameterised by subsets of a given set ATTR of attributes. Two complete deduction system for the language are presented: a Rasiowa–Sikorski style system using signed formulae, and an equivalent sequent calculus system in Gentzen style complete for theories.

1 Introduction

In the world around us, entities are perceived by the presence or absence of features, and the

$$\textsc{Object} \mapsto \textsc{Attribute}$$

relationship serves in many cases as a basis for knowledge representation. More formally, an *information system* is a tuple $\mathcal{I} = \langle \textsc{Obj}, \textsc{Attr}, \{V_a\}_{a \in \textsc{Attr}} \rangle$, such that

- OBJ is a nonempty set of *objects*,
- ATTR is a nonempty set of *attributes* or *features*, such that
  - Each $a \in \textsc{Attr}$ is a mapping from OBJ to 2^{V_a}, where V_a is a nonempty set of possible values of attribute a,

---

[*] The authors gratefully acknowledge partial support by the KBN/British Council Grant No WAR/992/151

which assigns to each object a set of values associated with a

(Pawlak, 1973, Lipski, 1976, Orłowska & Pawlak, 1987). If both OBJ and ATTR are finite, we can, in principle, depict the system as a data table, similar to a table of a relational database.

With each set A of attributes one can associate various relations on the object set, the most widely used being the relationship $ind(A)$ of *indiscernability*: here, two objects are related if they cannot be distinguished with the knowledge given by the attributes in A. In other words, for all $x, y \in$ OBJ,

$$\langle x, y\rangle \in ind(A) \text{ if and only if } a(x) = a(y) \text{ for all } a \in A. \tag{1.1}$$

Indeed, there are many types of such relations which can be sensibly derived from an information system, and the reader is invited to consult Orłowska (1995) for more details, and Düntsch et al. (1999) for a general form of relational representation.

While relations of similarity have been frequently studied and are well understood, the situation with regard to relations which describe the differences of objects is much less clear. One dissimilarity relation which has been studied in some detail is that of (strong) *complementarity*, where, for a nonempty set of attributes A,

$$\langle x, y\rangle \in comp(A) \text{ if and only if } a(x) = V_a \setminus a(y) \text{ for all } a \in A.$$

Demri & Orłowska (1998) show that $comp(A)$ is

C1. irreflexive,

C2. symmetric, and

C3. 3–transitive, i.e. for all $w, x, y, z \in$ OBJ,

$$\langle w, x\rangle, \langle x, y\rangle, \langle y, z\rangle \in comp(A) \text{ implies } \langle w, z\rangle \in comp(A).$$

Conversely, they show that for each relation R on OBJ with these properties there are an information system $\mathcal{I}$ and a set of attributes A such that $R = comp(A)$.

These relations are not independent, since

$$comp(\bigcup_{i\in I} A_i) = \bigcap_{i\in I} comp(A_i) \tag{1.2}$$

for any family $\{A_i : i \in I\}$ of nonempty subsets of ATTR, so, for example,

$$comp(A \cup B) = comp(A) \cap comp(B), \tag{1.3}$$

for all nonempty $A, B \subseteq$ ATTR.

Condition (1.2) says that two objects are complementary with respect to the attributes in $\bigcup_{i\in I} A_i$ iff they are complementary with respect to all attributes in each A_i for $i \in I$. In other words, the larger the nonempty set of attributes, the smaller the corresponding complementarity relation.

It was not without reason that above we limited our considerations to nonempty sets of attributes only. Indeed, it is not obvious how complementarity corresponding to an empty set of attributes is to be understood. One approach is based on remarking that in case of an empty set of attributes every object is assigned an empty set of attribute values, and the overall set of attribute values is also empty, so $\emptyset \setminus \emptyset = \emptyset$ implies that we could formally assume $comp(\emptyset) = \textsc{Obj} \times \textsc{Obj}$. However, this is counterintuitive, since the whole idea behind the complementarity relation is that the attribute values for the considered objects are totally different — and in case of an empty universe of values this cannot happen for any two attributes. Hence from the intuitive viewpoint is natural to assume an additional condition

$$comp(\emptyset) = \emptyset. \tag{1.4}$$

— which is what we shall do here.

The relations thus obtained are parameterised by the sets of attributes, and one speaks of the family $\langle \textsc{Obj}, comp(A)_{A \subseteq \textsc{Attr}} \rangle$ as a *family of complementarity relations parameterised by* $\textsc{Attr}$.

In this paper we will develop two complete formal systems of modal logic for reasoning about relative complementarity relations. The deduction systems we shall develop will be a Rasiowa-Sikorski style sequence formalism based on the use of signed formulae (Rasiowa & Sikorski, 1963), and a Gentzen-style sequent calculus complete for theories obtained from the former system. The approach which we adopt follows closely the one used by Konikowska (1997a,b) to develop a logic for reasoning about parameterised similarity relations. Different systems, based on sufficiency (as opposed to modal) operators and relative semantics have been presented by Düntsch & Orłowska (2000).

2 The language

The language $\mathcal{L}$ is parameterised by a fixed nonempty set $\textsc{Attr}$. The reason for fixing that set is that in a real-life information system the set of objects can change quite fast, whereas the set of attributes used for characterising these objects remains as a rule constant throughout the lifetime of a current system version.

The alphabet of $\mathcal{L}$ is the disjoint union of the following sets:

$\textsc{Cona} = \{\mathbf{a} : a \in \textsc{Attr}\}$	A set of constants representing individual attributes, one constant $\mathbf{a}$ for each single attribute a.
$\textsc{Varsa}$	A set of variables representing nonempty sets of attributes.
$\{\mathbf{0}\}$	A constant representing the empty set of attributes.
$\textsc{Varo}$	A set of variables representing individual objects.
$\textsc{Varso}$	A 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.
$\{\langle\,\rangle, [\,]\}$	Symbols for modalities.

There are two kinds of expressions:

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

The set $\textsc{Term}$ of terms is the smallest set satisfying the following conditions:

(i) $\textsc{Varsa} \cup \textsc{Cona} \cup \{\mathbf{0}\} \subseteq \textsc{Term}$,

(ii) If $A, B \in \textsc{Term}$, then $-A, A \cup B, A \cap B \in \textsc{Term}$.

The set $\textsc{Form}$ of formulae is the smallest set satisfying the following conditions:

(i). $\textsc{Varo} \cup \textsc{Varso} \subseteq \textsc{Form}$.

(ii). If $F, G \in \textsc{Form}$, then $\neg F, F \vee G, F \wedge G \in \textsc{Form}$.

(iii). If $A \in \textsc{Term}$ and $F \in \textsc{Form}$, then $[A]F, \langle A\rangle F \in \textsc{Form}$.

We shall also use the derived operator $\longrightarrow$ which serves as an abbreviation for

$$\neg F \vee G. \tag{2.1}$$

3 Semantics of $\mathcal{L}$

A *complementarity frame,* or just *frame,* is a pair

$$F = < \textsc{Obj}, \{comp(A)\}_{A \subseteq \textsc{Attr}} >, \tag{3.1}$$

where $\textsc{Obj}$ is a nonempty set of objects, $\textsc{Attr}$ a fixed nonempty set of attributes, and $\{comp(A)\}_{A\subseteq\textsc{Attr}}$ is a family of relations on $\textsc{Obj}$ which satisfy C1, C2, C3 on page 2, 1.4, (1.2) on page 2. For each $A \subseteq \textsc{Attr}$, we call $comp(A)$ the *complementarity relation corresponding to (the attributes in) A*. Except for (1.2), we do not put any restrictions on the assignment $A \mapsto comp(A)$. A *model of $\mathcal{L}$* is a pair $M = \langle F, v\rangle$, where $F = \langle \textsc{Obj}, \{comp(A)\}_{A\subseteq\textsc{Attr}}\rangle$ is a frame as in (3.1), and v is a (multi-sorted) valuation such that

- $v(A) \subseteq \textsc{Attr}$ for $A \in \textsc{VarsA}$,
- $v(O) \subseteq \textsc{Obj}$ for $O \in \textsc{VarsO}$,
- $v(x) \in \textsc{Obj}$ for $x \in \textsc{VarO}$.

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 **0** as $\emptyset$, each constant **a** $\in \textsc{ConA}$ as the corresponding attribute $a \in \textsc{Attr}$ (or, rather, as the singleton set containing a), the symbols $-, \cup, \cap$ as set–theoretical operations on sets of attributes, $\neg, \vee, \wedge$ as set–theoretical operations on sets of objects, and $\langle A\rangle, [A]$ as the possibility and necessity modalities corresponding to the accessibility relation $comp(\tau_M(A))$.

More precisely, the *interpretation of terms in M* is a function $\tau_M : \textsc{Term} \longrightarrow 2^{\textsc{Attr}}$ defined inductively as follows:

(i). $\tau_M(\mathbf{a}) = \{a\}$ for any $a \in \textsc{Attr}$, where **a** is the unique constant in $\textsc{ConA}$ representing a,

(ii). $\tau_M(\mathbf{0}) = \emptyset$,

(iii). $\tau_M(A) = v(A)$ for $A \in \textsc{VarsA}$,

(iv). For any $A, B \in \textsc{Term}$,

$$\begin{aligned}\tau_M(-A) &= \textsc{Attr} \setminus \tau_M(A),\\ \tau_M(A\cup B) &= \tau_M(A) \cup \tau_M(B),\\ \tau_M(A\cap B) &= \tau_M(A) \cap \tau_M(B).\end{aligned}$$

The interpretation of formulae is a function $\varphi_M : \textsc{Form} \longrightarrow 2^{\textsc{Obj}}$ such that for all $x \in \textsc{VarO}$, $O \in \textsc{VarsO}$, $F, G \in \textsc{Form}$, and $A \in \textsc{Term}$,

$$\begin{aligned}\varphi_M(x) &= \{v(x)\},\\ \varphi_M(O) &= v(O),\\ \varphi_M(\neg F) &= \textsc{Obj} \setminus \varphi_M(F),\\ \varphi_M(F\vee G) &= \varphi_M(F) \cup \varphi_M(G),\\ \varphi_M(F\wedge G) &= \varphi_M(F) \cap \varphi_M(G),\\ \varphi_M([A]F) &= \{o \in \textsc{Obj} : (\forall o' \in \textsc{Obj})((o,o') \in comp(\tau_M(A)) \rightarrow o' \in \varphi_M(A))\}\\ \varphi_M(\langle A\rangle F) &= \{o \in \textsc{Obj} : (\exists o' \in \varphi_M(A))(o,o') \in comp(\tau_M(A))\}.\end{aligned}$$

A formula $F \in \textsc{Form}$ is said to be *true in a model* M, written $\models_M F$, iff $\varphi_M(F) = \textsc{Obj}$, i.e. iff it evaluates to the whole object–universe of this model. A formula F is called *valid* iff $\models_M F$ for every model M.

It can be easily seen that, in general, neither the formula F nor the formula $\neg F$ need hold in a given model. Consequently, due to our notion of satisfaction, $\neg$ does not directly correspond to non-satisfaction, which should be kept in mind for things to come.

4 Signed formulae

In the development of our first deduction system, we shall use signed formulae, which are a well-known tool for expressing satisfaction and non-satisfaction of formulae. This type of "internalising" semantical notions has also been used successfully on the object level, see e.g. Blackburn (1998). The signs are dropped when we pass to the Gentzen deduction system, and the system we obtain is a sequent calculus for the original language. Syntactically, signed formulae are simply formulae in $\textsc{Form}$ preceded by one of the signed formula constructors $\mathbf{T}, \mathbf{N}$. More formally, the set $\textsc{Sform}$ of signed formulae is

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

where **T** stands for "true", or "satisfied", and **N** for "false", or "not satisfied".

The interpretation of signed formulae is a function $\sigma_M : \textsc{Sform} \longrightarrow \{\mathbf{tt}, \mathbf{ff}\}$, where **tt** denotes truth and **ff** falsity, defined by

$$\sigma_M(\mathbf{T}(F)) = \begin{cases} \mathbf{tt} & \text{iff} \models_M F, \\ \mathbf{ff} & \text{otherwise,} \end{cases} \qquad \sigma_M(\mathbf{N}(F)) = \begin{cases} \mathbf{tt} & \text{iff non} \models_M F, \\ \mathbf{ff} & \text{otherwise} \end{cases}$$

For a signed formula G, we say that G *is true in* M, and write $\models_M G$, iff $\sigma_M(G) = \mathbf{tt}$. We say that G is *valid* – and write $\models G$ – if and only if $\models_M G$ for any model M.

The deduction system we are going to develop will be based on formulae expressing the membership relation, namely,

$$x \in F \overset{\text{def}}{=} x \longrightarrow F, \tag{4.1}$$

Special instances of such formulae are $x \in y$ and

$$x\ cp(A)\ y \overset{\text{def}}{=} x \in \langle A \rangle y. \tag{4.2}$$

Here $x, y \in \textsc{Varo}, F \in \textsc{Form}, A \in \textsc{Term}$. Since

$$\models_M F \longrightarrow G \text{ iff } \varphi_M(F) \subseteq \varphi_M(G),$$

it can be easily seen that formulae of type (4.1) indeed represent the membership relation with $x \in x$ representing equality, and formulae of type (4.2) the complementarity relation. For the corresponding signed formulae we have

(4.3) $$\models_M \ \mathbf{T}(x \in F) \text{ iff } v(x) \in \varphi_M(F),$$

(4.4) $$\models_M \ \mathbf{N}(x \in F) \text{ iff } v(x) \notin \varphi_M(F)$$

(4.5) $$\models_M \ \mathbf{T}(x \; cp(A) \; y) \text{ iff } (v(x), v(y)) \in comp(\tau_M(A)),$$

(4.6) $$\models_M \ \mathbf{N}(x \; cp(A) \; y) \text{ iff } (v(x), v(y)) \notin comp(\tau_M(A)),$$

(4.7) $$\models_M \ \mathbf{T}(x \in y) \text{ iff } v(x) = v(y),$$

(4.8) $$\models_M \ \mathbf{N}(x \in y) \text{ iff } v(x) \neq v(y).$$

Indeed, for (4.3),

$$\varphi_M(x \in F) = \varphi_M(\neg x \vee F) = (\textsc{Obj} - \{v(x)\}) \cup \varphi_M(F),$$

whence $\models_M \ \mathbf{T}(x \in F)$ iff $\varphi_M(x \in F) = \textsc{Obj}$ iff $v(x) \in \varphi_M(F)$. Furthermore, (4.4) – (4.6) are just special cases of (4.3).

The formulae above provide the classical dichotomy of "satisfaction — non-satisfaction" not expressible directly in our original language, and are very useful for developing the deduction system. This is expressed in the following result from Konikowska (1997a) (see also Gabbay, 1981):

Lemma 4.1. *For any formula $F \in \textsc{Form}$:*

(i) $\mathbf{T}(F)$ *is valid iff* $\mathbf{T}(x \in F)$ *is valid, where $x \in \textsc{VarO}$ is any variable not occurring in F,*

(ii) $\mathbf{N}(F)$ *is valid whenever for any model M there is a variable $x \in \textsc{VarO}$ such that* $\mathbf{N}(x \in F)$ *is true in M.*

5 Components and transformation to normal form

In order to cope with the modalities corresponding to various sets of attributes, we use the "component" approach described in Konikowska (1997a,b) for the case of the similarity logic: 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

$$\textsc{ConA}(\Omega) = \{\mathbf{a} \in \textsc{ConA} : \mathbf{a} \text{ occurs in } \Omega\},$$

and

$$\textsc{VarSA}(\Omega) = \{A \in \textsc{VarSA} : A \text{ occurs in } \Omega\}$$

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

Suppose that

$$\textsc{Cona}(\Omega) = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}, \qquad \textsc{Varsa}(\Omega) = \{Q_1, \ldots, Q_m\},$$

where $n, m \geq 0$. The sequence Ω is said to be *nondegenerate* iff $m + n > 0$, i.e. iff it contains symbols from VARSA $\cup$ CONA[1].

Suppose that Ω is nondegenerate, and

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

for any term A. We denote

$$\begin{aligned} \textsc{Scomp}(\Omega) &= \{Q_1^{i_1} \cap \ldots \cap Q_m^{i_m} : i_1, \ldots, i_m \in \{+, -\}\}, \\ \textsc{Comp}(\Omega) &= \{\mathbf{a}_j \cap S : S \in \textsc{Scomp}(\Omega), 1 \leq j \leq n\} \cup \{-\mathbf{a}_1 \cap \ldots \cap -\mathbf{a}_n \cap S : S \in \textsc{Scomp}(\Omega)\} \end{aligned}$$

The elements of $\textsc{Comp}(\Omega)$ are called *components for* Ω, and those of $\textsc{Scomp}(\Omega)$ *subcomponents*. Note that the notion of component is defined for nondegenerate sequences only. By a *positive component* we mean any component of the form $\mathbf{a}_j \cap S$. Components will be always denoted by a suitably indexed C, and subcomponents by S. For nonpositive components we shall often use the shorthand notation of the form $-\mathbf{a} \cap S$, which should be read as $-\mathbf{a}_1 \cap \ldots \cap -\mathbf{a}_n \cap S$, where $\mathbf{a} = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$.

The components and subcomponents have the following important properties (Konikowska, 1997b):

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

1. *The sets $\{\tau_M(C)\}_{C \in \textsc{Comp}(\Omega)}$ form a disjoint cover of* ATTR, *i.e.*
   - *(a) $\tau_M(C) \cap \tau_M(C') = \emptyset$ for any $C, C' \in \textsc{Comp}(\Omega)$ such that $C \neq C'$;*
   - *(b) $\bigcup_{C \in \textsc{Comp}(\Omega)} \tau_M(C) =$ ATTR.*
2. *The sets $\{\tau_M(S)\}_{S \in \textsc{Scomp}(\Omega)}$ also form a disjoint cover of* ATTR.
3. *For any term A occurring in Ω, either*
   - *(i) A is semantically equivalent to* **0**, *or*
   - *(ii) there exists a unique subset $\{C_1, \ldots, C_k\}$ of $\textsc{Comp}(\Omega)$ 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 **0** or $-\mathbf{0}$.

---

[1] The reader should note that **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 **0** is considered as degenerate.

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 **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 with union semantically equivalent to A defined in Lemma 5.1.

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(3), 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 Rasiowa & Sikorski (1963), and will be henceforth called an *R–S system*. It consists of decomposition rules for sequences of signed formulae, and of axiomatic sequences to be defined below. Using 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. An R–S system is dual to a tableau system of the type used by Beth (1959) or Fitting (1983). One basic difference is that, in contrast to a refutation based tableau system, we try to show the validity of a formula or sequence of formulae. 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; a sequence of formulae then is valid iff it has a finite decomposition tree all of whose branches end in axiomatic sequences. The other difference is that in the tableau system an application of a rule to a vertex v takes into consideration the labels of all the ancestors of v, whereas in an R–S system the vertex explicitly inherits all the necessary information from its ancestor. In applying a rule to a vertex we need not consider anything but the label of the vertex.

A sequence $\Omega = G_1, G_2, \ldots, G_n$ of signed formulae is called *true in a model* M, written $\models_M \Omega$, iff $\models_M G_i$ for some $1 \leq i \leq n$. Ω is called *valid* iff $\models_M \Omega$ for every model M. A decomposition rule is either a pair Ω_1, Ω_2 or a triple $\Omega_1, \Omega_2, \Omega_3$ of sequences of formulae in SFORM, usually written as

$$\frac{\Omega_1}{\overline{\Omega_2}}, \quad \text{or} \quad \frac{\Omega_1}{\overline{\Omega_2 \mid \Omega_3}},$$

respectively. Ω_1 is called the *conclusion* of the rule, and Ω_2 (Ω_2, Ω_3) its *premise* (*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 a 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 (6.1) – (6.3) or pairs of formulae (6.4), (6.5) given below:

(6.1) $\mathbf{N}(x\ cp(\mathbf{0})\ y),$

(6.2) $\mathbf{N}(x\ cp(A)\ x),$

(6.3) $\mathbf{T}(x \in x),$

(6.4) $\mathbf{T}(F), \mathbf{N}(F),$

(6.5) $\mathbf{N}(x\ cp(\mathbf{a}_j \cap S)\ y), \mathbf{N}(x'\ cp(\mathbf{a}_j \cap S')\ y'),$

where S, S' are two different subcomponents in $\textsc{Scomp}(\Omega)$.

Lemma 6.1. *The signed formulae* (6.1) – (6.5) *are valid.*

Proof. The validity of (6.1) – (6.3) follows from C1, C2 and (4.3), (4.4), and, clearly, the pair (6.4) is also valid.

For the proof of (6.5), consider a 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, $\tau_M(\mathbf{a_j}) = \{v(\mathbf{a_j})\}$ is a singleton, and it follows that either $\tau_M(\mathbf{a_j}) \cap \tau_M(S) = \emptyset$ or $\tau_M(\mathbf{a_j}) \cap \tau_M(S') = \emptyset$. In other words, either $\tau_M(\mathbf{a_j} \cap S) = \emptyset$ or $\tau_M(\mathbf{a_j} \cap S') = \emptyset$. Since by 1.4 $comp(\emptyset) = \emptyset$, either $comp(\tau_M(\mathbf{a_j} \cap S)) = \emptyset$ or $comp(\tau_M(\mathbf{a_j} \cap S')) = \emptyset$. In the first case, we obviously have $(v(x), v(y)) \notin comp(\tau_M(\mathbf{a_j} \cap S))$, and in the second case $(v(x'), v(y')) \notin comp(\tau_M(\mathbf{a_j} \cap S'))$. Thus, by (4.4), one of the formulae in the pair must be true in M. ☐

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 $G \in \textsc{Sform}$ is said to be *indecomposable* iff it has one of the following forms:

(i) $\mathbf{T}(x \in O)$ or $\mathbf{N}(x \in O)$, where $x \in \textsc{Varo}, O \in \textsc{Varso} \cup \textsc{Varo}$,

(ii) $\mathbf{T}(x\ cp(C)\ y)$ or $\mathbf{N}(x\ cp(C)\ y)$, where $x, y \in$ VARO, and C is **0**, $-\mathbf{0}$, or

  (a) C is any component, if ATTR is infinite;

  (b) C is any positive component, if ATTR is finite.

Otherwise a formula is said to be *decomposable*.

Note that (ii) 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, **0** is also "indecomposable", and $-\mathbf{0}$ can be encountered only in the normal form of a degenerate formula, which has no components at all; thus, we cannot split $-\mathbf{0}$ into anything smaller. The condition (b) in (ii) follows from the fact that in case of a finite ATTR we can eliminate all negative components of the form $-\mathbf{a}\cap S$ by replacing them with a union of positive components of the form $\bigcup_{\mathbf{a}' \in \mathrm{CONA}-\mathbf{a}}(\mathbf{a}'\cap S)$.

A sequence Ω of signed formulae is said to be *indecomposable* iff all its elements are indecomposable.

The decomposition rules for the language of signed formulae is given in Tables 1 – 3[2].

In all DRS rules, $x, y, z \in$ VARO, $F, G \in$ FORM, Ω' and Ω" are sequences of signed formulae, with the prerequisite that Ω' is indecomposable, $A, B \in$ TERM, C is a component, and S a subcomponent. Since the decomposition process is applied to sequences in normal form only, the only terms encountered in the actually decomposed formulae will be unions of components; thus we need no rules for term constructors other than the union.

The rules (**T**) and (**N**), which are sound by Lemma 4.1, allow us to manage the awkward problem of "neither F nor $\neg F$ holds" by reducing the original formulae (in repeating steps in case of (**N**)) to dichotomous formulae of the form $\mathbf{T}(x \in F)$, $\mathbf{N}(x \in F)$. This reduction is analogous to that used in first-order logics in case of quantifiers: 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 $\mathbf{N}(F)$.

The rules for finite ATTR are justified by the fact that, in case of a finite ATTR, each negative component of the form $-\mathbf{a} \cap S$ can be replaced by a finite union of components of the form $\mathbf{a}' \cap S$ over $\mathbf{a}' \in$ CONA $\setminus$ $\mathbf{a}$.

From (4.3) – (4.5) and Lemma 4.1 we immediately obtain

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

The general idea of the deduction system is to break down the formulae occurring in a sequence of signed formulae into some elementary, indecomposable parts, whose validity determines the validity of the original sequence. In what follows, the five rules marked $(*)$ will be called *e*xpansion rules, and all others *replacement rules*. Roughly speaking, replacement rules replace the original formula they act upon by simpler formulae, whereas expansion rules add new formulae to the sequence, e.g. close it under some symmetry or transitivity law.

It can be easily seen that

---

[2] Tables are placed at the end of the paper.

- Replacement rules are applicable to decomposable sequences only.
- To any decomposable sequence of formulae we can apply at most one replacement rule; namely, the unique rule applicable to the leftmost decomposable formula in this sequence.
- Expansion rules can also be applied to indecomposable sequences.

Note that there may be several expansion rules applicable to a given sequence. To avoid ambiguity, we shall give a method of choosing an appropriate rule to be applied, as well as a variable in the quantifier–like rules.

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.

In order to make the decomposition tree unique (i.e. to avoid ambiguity), we need one more notion. Let Ω be a sequence of signed formulae; a decomposition rule R in DRS is called *correctly applicable to* Ω if both the following conditions are satisfied:

(i) R is a rule applicable to Ω such that its application augments Ω by some new formula,

(ii) There is no rule with this property that can be applied to a formula or a subsequence of formulae in Ω lying to the left of the formula or formulae, to which R can be applied[3].

Since there is at most one replacement rule applicable to any sequence of formulae, and (ii) uniquely defines the expansion rule which is correctly applicable to Ω, we obtain

Lemma 7.1. *At most one decomposition rule is correctly applicable to a any given sequence* Ω *of signed formulae.*

Hence, we can talk about *the unique rule R* correctly applicable to a given sequence Ω; this is of a fundamental importance for defining the notion a decomposition tree. To assure an unambiguous

---

[3] If $\ll$ is the relation of “lying to the left”, then we assume that for all $(F_1, \dots, F_k), (G_1, \dots, G_l)$, we have $(F_1, \dots, F_k) < (G_1, \dots, G_l)$, iff, for some $1 \leq m \leq min(k, l)$, $F_r = G_r$ for $r = 1, \dots, m-1$ and $F_m \ll G_m$, where the latter relation is understood in the natural way.

choice of variables in the decomposition process, we suppose for the rest of this paper that VARO is a well-ordered set with respect to some ordering $\leq$.

By a *decomposition tree* $DT(\Omega)$ for a *sequence* Ω of signed formulae we mean a maximal binary tree 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 correctly applicable to Σ;

  (b) Otherwise we expand the branch B beyond v by attaching to this node

    (b1) a single son labelled Σ_1, if the unique rule correctly applicable to Σ is of the form $\dfrac{\Sigma}{\Sigma_1}$,

    (b2) two sons labelled Σ_1 and Σ_2, respectively, if the unique rule correctly applicable to Σ is of the form $\dfrac{\Sigma}{\Sigma_1 \mid \Sigma_2}$.

  In a rule involving the choice of a new variable we choose the first variable with respect to the ordering $\leq$ which does not appear in Σ, see Rasiowa & Sikorski (1963) for the details.

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 the unique correctly applicable rule; the conditions on the choice of variables in quantifier-like rules assure the uniqueness of the extension. Hence, $DT(\Omega)$ is uniquely determined by $n(\Omega)$.

It is easy to see that $DT(\Omega)$ may be infinite — due to the ($\mathbf{N}$, $\mathbf{T} \in \langle A \rangle$) 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 sequence Ω of signed formulae is called *provable* in DRS, written as $\vdash_{DRS} \Omega$, iff its decomposition tree $DT(\Omega)$ is finite and all its terminal sequences are axiomatic.

From the equivalence character of the rules, it is evident that

---

[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.

Lemma 7.2. *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 7.3. *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:

$$\mathbf{T}(x \in O), \mathbf{N}(x \in O), \mathbf{T}(x\ cp(C)\ y), \mathbf{N}(x\ cp(C)\ y),$$

where $x, y \in$ VARO, $O \in$ VARSO $\cup$ VARO, and C is either **0** or $-\mathbf{0}$, or any component if ATTR is infinite, and any positive component if ATTR is finite, see the definition on page 10. Let

$$\text{CONA}(\Omega) = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}, \qquad \text{VARSA}(\Omega) = \{Q_1, \ldots, Q_m\},$$

and set $\text{CONA}(\Omega) = \mathbf{a}$. Then the terms C which can appear in Σ are as follows:

(i). $C = \mathbf{0}$.

(ii). $C = -\mathbf{0}$. 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 **0**s,

(iii). If $C \notin \{\mathbf{0}, -\mathbf{0}\}$ and ATTR is infinite, then C can be either of the form $\mathbf{a}_j \cap S$ or of the form $-\mathbf{a} \cap S$, where $S \in \text{SCOMP}(\Omega)$; if ATTR is finite, then C can only be of the form $\mathbf{a}_j \cap S$.

Recall that any subcomponent $S \in \text{SCOMP}(\Omega)$ is of the form

$$S = \{Q_1^{i_1} \cap \ldots \cap Q_m^{i_m} : i_1, \ldots, i_m \in \{+, -\}\},$$

where $A^+ = A,\ A^- = -A$ for any term A.

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

Let $\rho \subseteq$ VARO $\times$ VARO be a binary relation defined by

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

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

$$\rho^* = \rho \cup \{(x, x) : x \in \text{VARO}\}$$

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

$$F = \langle \textsc{Obj}, \{comp(A)\}_{A \subseteq \textsc{Attr}} \rangle$$

where

$$\textsc{Obj} = \textsc{Varo}/\rho^*$$

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

   (7.1) $\qquad v(x) = v(y)$ iff the formula $\mathbf{N}(x \in y)$ occurs in Σ

   for all $x, y \in \textsc{Varo}, x \neq y$.

2. For all $O \in \textsc{Varso}$,

   $$v(O) = \{v(x) : \mathbf{N}(x \in O) \text{ occurs in } \Sigma\}.$$

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

   $$\textsc{Cona}(\Omega) = \mathbf{a} = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}, \qquad \textsc{Varsa}(\Omega) = \{Q_1, \ldots, Q_m\}.$$

   We consider two cases:

   (a) $\textsc{Attr}$ is infinite:

   In this case $A' = \textsc{Attr} \setminus \{a_1, \ldots, a_n\}$ is also infinite. As the set of subcomponents $\textsc{Scomp}(\Omega)$ is finite, there exists an injection $\Phi : \textsc{Scomp}(\Omega) \hookrightarrow A'$ with

   $$\Phi(S) \neq a_j \text{ for all } S \in \textsc{Scomp}(\Omega) \text{ and all } j,\ 1 \leq j \leq n.$$

   By clause (6.5) on page 10, any sequence containing both $\mathbf{N}(x\, cp(\mathbf{a}_j \cap S))$ and $\mathbf{N}(x'\, cp(\mathbf{a}_j \cap S')\ y')$, where $S \neq S'$, is axiomatic. Since, by our assumption, Σ is not axiomatic, it follows that for any $1 \leq j \leq n$, there is at most one $S \in \textsc{Scomp}(\Omega)$ such that for some $x, y \in \textsc{Varo}$ the formula $\mathbf{N}(x\ cp(\mathbf{a}_j \cap S)\ y)$ occurs in Σ. If such an S exists for a given j, we denote it by S_j and say that j is *positive in* Σ. We put $v(Q) = \emptyset$ for any $Q \in \textsc{Varsa} \setminus \{Q_1, \ldots, Q_m\}$ and

   $$\begin{aligned} v(Q_k) &= \{\Phi(S) : S \in \textsc{Scomp}(\Omega) \text{ and } Q_k^+ \text{ occurs in } S\} \\ &\cup \{a_j : j \text{ positive in } \Sigma \text{ and } Q_k^+ \text{ occurs in } S_j\} \end{aligned}$$

(b) ATTR is finite:

In this case the only components that can occur in Σ are of the form $\mathbf{a}_j \cap S$, where $S \in \text{SCOMP}(\Omega)$. The definition of v restricted to VARSA is a simplification of the one given above; we put $v(Q) = \emptyset$ for any $Q \in \text{VARSA} - \{Q_1, \dots, Q_m\}$ and

$$v(Q_k) = \{a_j : j \text{ positive in } \Sigma \text{ and } Q_k^+ \text{ occurs in } S_j\}$$

Recall that by convention $\mathbf{a} = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$, and $-\mathbf{a} = -\mathbf{a}_1 \cap \dots \cap -\mathbf{a}_n$. One can show that for the interpretation of terms τ_H induced by the valuation v

(i) $\tau_H(\mathbf{a}_j \cap S) = \begin{cases} \{a_j\} & \text{if } j \text{ is positive in } \Sigma \text{ and } S = S_j \\ \emptyset & \text{otherwise.} \end{cases}$

(ii) If ATTR is infinite, then $\tau_H(-\mathbf{a} \cap S) = \{\Phi(S)\} \neq \{a_j\}$ for any $1 \leq j \leq n$.

Thus, τ_H evaluates all components occurring in Σ to singletons or $\emptyset$. This is the key property which helps us to define the family of complementarity relations $\{comp(A)\}_{A \subseteq \text{ATTR}}$ on OBJ × OBJ, which is the last step needed to complete the definition of the model. Due to property C3, complementarity relations corresponding to individual attributes in ATTR generate the whole family $\{comp(A)\}_{A \subseteq \text{ATTR}}$; they can also be defined independently of each other, which solves the basic technical problem connected with modalities parameterised by arbitrary sets of attributes.

We begin with defining the family of complementarity relations $\{comp(a)\}_{a \in \text{ATTR}}$ 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 $\mathbf{T}(x\ cp(C)\ y)$ or $\mathbf{N}(x\ cp(C)\ y)$ occurring in Σ, we have either $C = \mathbf{0}$ or $C = -\mathbf{0}$. We put

$$comp(a) = \{(v(x), v(y)) : \text{ the formula } \mathbf{N}(x\ cp(-\mathbf{0})\ y) \text{ is in } \Sigma\}$$

for each $a \in \text{ATTR}$.

CASE 2. Σ does not contain $-\mathbf{0}$; then, for any formula of the form $\mathbf{T}(x\ cp(C)\ y)$ or $\mathbf{N}(x\ cp(C)\ y)$ occurring in Σ, the term C is either a component or $\mathbf{0}$.

We have to consider two subcases:

(i) ATTR is infinite. Then, for any $a \in \text{ATTR}$, we define

$$comp(a) = \begin{cases} \{(v(x), v(y)) : \mathbf{N}(x\ cp(\mathbf{a}_j \cap S_j)\ y) \text{ is in } \Sigma\} & \text{if } a = a_j, \text{ where } 1 \leq j \leq n \\ & \text{and } j \text{ is positive in } \Sigma, \\ \{(v(x), v(y)) : \mathbf{N}(x\ cp(-\mathbf{a} \cap S)\ y) \text{ is in } \Sigma\} & \text{if } a = \Phi(S), \text{where } S \in \text{SCOMP}(\Omega), \\ \emptyset & \text{otherwise.} \end{cases}$$

The definition is correct, because Φ is one-to-one and $\Phi(S) \neq a_j$ for $j = 1, \dots, n$.

(ii) ATTR is finite. The definition is again a simplification of that for infinite ATTR: we put

$$comp(a) = \begin{cases} \{(v(x), v(y)) : \mathbf{N}(x\ cp(\mathbf{a}_j \cap S_j)\ y) \text{ is in } \Sigma\} & \text{if } a = a_j\text{, where } 1 \leq j \leq n \\ & \text{and } j \text{ is positive in } \Sigma, \\ \emptyset & \text{otherwise.} \end{cases}$$

Having defined the singleton-based relations, we set

$$comp(\emptyset) = \emptyset, \tag{7.2}$$

and

$$comp(A) = \bigcap_{a \in A} comp(a) \tag{7.3}$$

for all $\emptyset \neq A \subseteq$ ATTR.

We shall show that the family $\{comp(A)\}_{A \subseteq \text{ATTR}}$ defined above is a family of complementarity relations which satisfies conditions C1 – C3 and (1.4). The degenerate Case 1 is clear, and thus, we will only show Case 2.

Conditions C1 and C3 follow immediately from the definitions (7.2) and (7.3). Thus, all that is left to show is C2. Since the intersection of complementarity relations is again a complementarity relation, and by (7.3), it is enough to consider the case when A is a singleton, say, $A = \{a\}$.

For irreflexivity, assume that $([z], [z]) \in comp(a)$ for some $z \in$ VARO. Then, there exist $x, y \in$ VARO and $A \in$ TERM such that $v(x) = v(y) = [z]$ and $\mathbf{N}(x\ cp(A)\ y) \in \Sigma$. But $v(x) = v(y)$ implies $\mathbf{N}(x \in y) \in \Sigma$ by (7.1), whence by rule $(tran \in cp)$ we obtain $\mathbf{N}(x\ cp(A)\ x) \in \Sigma$. (6.2) now implies that Σ is axiomatic, a contradiction. It follows, that $comp(a)$ is irreflexive.

The fact that $comp(a)$ is 3-transitive is proved in a similar way, based on rule $(3 - trancp)$.

Finally, symmetry follows directly from rule $(sym\ cp)$: Since Σ is closed under this rule, $\mathbf{N}(x\ \ cp(A) y) \in \Sigma$ implies $\mathbf{N}(y\ cp(A)\ x) \in \Sigma$ for any term A.

Thus, $F = \langle \text{OBJ}, \{comp(A)\}_{A \subseteq \text{ATTR}} \rangle$ is a complementarity frame as defined in (3.1), and $H = \langle F, v \rangle$ is model of our language.

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

$$\mathbf{T}(x \in y), \mathbf{N}(x \in y), \mathbf{T}(x \in O), \mathbf{N}(x \in O), \mathbf{T}(x\ cp(C)\ y), \mathbf{N}(x\ cp(C)\ y),$$

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

For example, if $\mathbf{N}(x \in y)$ is in Σ, then, by definition, we have $v(x) = v(y)$, whence by (4.8) on page 7, $\not\models_{M_H}\ \mathbf{N}(x \in y)$.

When proving that the $\mathbf{T}$–type formulae are not true, we also make use of the fact that Σ is not axiomatic. For example, if $\mathbf{T}(x \in y)$ is in Σ, then $x \neq y$, because of (6.3), and, in addition, $\mathbf{N}(x \in y)$ cannot occur in Σ by (6.4). This yields $v(v) \neq v(y)$, whence $\not\models_H\ \mathbf{T}(x \in y)$ by (4.7).

The proofs for the cp–type formulae follow from the equalities giving $v(C)$ for any component C that can occur in Σ, as well as from the definition of the complementarity relations $comp(a)$. The details of analogous proof for similarity logic can be found in Konikowska (1997a).

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 7.4. *Every valid sequence of signed formulae* $\Omega \in$ 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 7.3, 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 Rasiowa & Sikorski (1963), p. 302, to suit the structure of our language. The details are analogous to those in Konikowska (1997a), 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 the 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 neither of the three single formulae (6.1) – (6.3) nor any of the two pairs of formulae (6.4), (6.5) on page 10, whose presence in a sequence make it axiomatic.

Thus, reasoning exactly in the same way as in the proof of Lemma 7.3, we can build a model H such that $\not\models_H G$ 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 Konikowska (1997a). $\square$

8 A sequent calculus for complementarity

In this section, we will present a sequent calculus arising from the complete R-S proof system for sequences of signed formulae we have just described. The method of transforming an R-S system into a sequent calculus was described in Konikowska (1997a), which the reader is invited to consult for any details omitted below.

The sequents will only involve ordinary, non-signed formulae of the language $\mathcal{L}$ as described in Section 2; sets of such formulae will be denoted by Γ, Δ with suitable indices. A model M of $\mathcal{L}$ is called a *model of* Γ iff every formula in Γ is satisfied in M in the semantics given in Section 3.

By a *sequent* we mean a pair (Γ, Δ) of finite sets $\Gamma, \Delta \subseteq \textsc{Form}$, usually written in the form $\Gamma \vdash \Delta$. In the sequent notation $\Gamma, F \vdash \Delta, G$, commas denote set-theoretical union, and individual formulae $F, G \in \textsc{Form}$ are identified with the respective singletons.

The sequent $\Gamma \vdash \Delta$ is called *valid*, written $\models (\Gamma \vdash \Delta)$, iff

(8.1) $$\models_M \Gamma \text{ implies } (\exists F \in \Delta) \models_M F.$$

Thus, $\Gamma \vdash \Delta$ is valid if a model M which satisfies each formula in Γ also satisfies at least one formula in Δ.

For any finite set of formulae $Ax \subseteq \textsc{Form}$, and any $F \in \textsc{Form}$, we say that F *is a semantic consequence of Ax*, and write $Ax \models F$, iff F holds in every model of Ax. The set Ax may be treated as a set of axioms of some specific theory T; then $Ax \models F$ iff F is a semantic consequence of the theory T. Evidently, $Ax \models F$ iff the sequent $Ax \vdash F$ is valid, which implies that the syntactic entailment $\vdash$ in a valid sequent may be viewed as a formal counterpart of the semantic consequence relation. In other words, a complete axiomatisation of the sequent calculus will automatically provide a deduction system complete for theories.

The sequent calculus (SC) which we shall develop consists of *axioms*, having the form of single valid sequents, and *inference rules*, leading from valid sequents to valid sequents. A deduction rule has one of the two forms

$$\frac{\mathbf{S}_1}{\mathbf{S}} \quad \text{or} \quad \frac{\mathbf{S}_1 \qquad \mathbf{S}_2}{\mathbf{S}}.$$

The sequent $\mathbf{S}$ will be called *the conclusion* of the rule, and the sequent(s) $\mathbf{S}_1$ ($\mathbf{S}_1, \mathbf{S}_2$) — its *premise(s)*. A rule is called *sound* iff its conclusion is valid whenever all its premises are valid.

It turns out that there is a simple connection between validity of sequents and validity of sequences of signed formulae. For any finite set $\Gamma = \{F_1, F_2, \ldots, F_k\} \subseteq \textsc{Form}$, let $\mathbf{T}(\Gamma)$ and $\mathbf{N}(\Gamma)$ be sequences of signed formulae obtained by preceding all elements of Γ with the operators **T** and **N**, respectively, so that

$$\mathbf{T}(\Gamma) = \mathbf{T}(F_1), \mathbf{T}(F_2), \ldots, \mathbf{T}(F_n), \quad \mathbf{N}(\Gamma) = \mathbf{N}(F_1), \mathbf{N}(F_2), \ldots, \mathbf{N}(F_n)$$

We now have

Lemma 8.1. *A sequent* $\Gamma \vdash \Delta$ *is valid whenever the sequence* $\mathbf{N}(\Gamma), \mathbf{T}(\Delta)$ *of signed formulae is valid.*

The above fact suggests a straightforward method of developing a sequent calculus for our original formal language based on the DRS deduction system for sequences of signed formulae; we simply have to derive from the decomposition rules of DRS the corresponding inference rules of the sequent calculus, leading from valid sequents to valid sequents.

For any sequence Ω of signed formulae, let us denote

$$\Omega^+ = \{F \in \textsc{Form} : \mathbf{T}(F) \text{ is in } \Omega\}, \quad \Omega^- = \{F \in \textsc{Form} : \mathbf{N}(F) \text{ is in } \Omega\}.$$

Lemma 8.1 implies the following basic result:

Lemma 8.2.

- *For every axiomatic sequence* Ω *of DRS,* $\Gamma, \Omega^- \vdash \Delta, \Omega^+$ *is a valid axiom of the sequent calculus.*
- *For every rule* $\dfrac{\Omega', \Pi, \Omega''}{\Omega', \Sigma, \Omega''}$ *in DRS,* $\dfrac{\Gamma, \Sigma^- \vdash \Delta, \Sigma^+}{\Gamma, \Pi^- \vdash \Delta, \Pi^+}$ *is a valid inference rule of the sequent calculus.*
- *For every rule* $\dfrac{\Omega', \Pi, \Omega''}{\Omega', \Sigma_1, \Omega'' \mid \Omega', \Sigma_2, \Omega''}$ *in DRS,* $\dfrac{\Gamma, \Sigma_1^- \vdash \Delta, \Sigma_1^+ \qquad \Gamma, \Sigma_2^- \vdash \Delta, \Sigma_2^+}{\Gamma, \Pi^- \vdash \Delta, \Pi^+}$ *is a valid rule of the sequent calculus.*

The resulting axioms and inference rules of the sequent calculus (SC) are given in Tables 4 – 7 alongside the original axiomatic sequences and rules of DRS from which they were derived. Note that, in addition to the rules derived from DRS, we have also included *thinning rules* in the sequent calculus, which allow us simplify some of the derived rules. In their presence, we can avoid repeating certain formulae appearing in the premise(s) of the rule in its conclusion; this concerns especially the expansion rules.

That is exactly why some of the sequent calculus rules given in Tables 5-6 do not correspond to the general "recipe" for generating such rules given in Lemma 8.2. For example, the sequent rule for introducing $\langle A \rangle$ on the right hand side should formally be

$$\frac{\Gamma \vdash \Delta, y \in F, x \in \langle A \rangle F \qquad \Gamma \vdash \Delta, x \; cp(A) \; y, x \in \langle A \rangle F}{\Gamma \vdash \Delta, x \in \langle A \rangle F}$$

but thanks to the weakening rules we could replace it by the simpler rule

$$\frac{\Gamma \vdash \Delta, y \in F \qquad \Gamma \vdash \Delta, x \; cp(A) \; y}{\Gamma \vdash \Delta, x \in \langle A \rangle F}$$

In all axioms and rules given in Tables 4 – 7, we suppose that $x, y, z \in$ VARO, $F, G \in$ FORM, the Ω's are sequences of signed formulae, Γ, Δ are finite subsets of FORM, $A, B \in$ TERM, $\mathbf{a} = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\} \subseteq$ CONA, $-\mathbf{a} = -\mathbf{a}_1 \cap \ldots \cap -\mathbf{a}_n$, C is a component, and S a subcomponent. The components and subcomponents for a sequent Γ, Δ are defined as components and subcomponents for the sequence of signed formulae $\mathbf{N}(\Gamma), \mathbf{T}(\Delta)$ equivalent to that sequent.

In the rules marked by the symbols listed below, we also make the following stipulations:

> * – F is not of the form $z \in G$ for any $z \in$ VARO and any $G \in FORM$, x is a a new variable in VARO, i.e. x occurs neither above the double line in the DRS rule nor below the single line in the SC rule.
>
> ** – F is not of the form $z \in G$ for any $z \in$ VARO and any $G \in FORM$, y is an arbitrary variable in VARO
>
> *** – z is a new variable, i.e. occurs neither above the double line in the DRS rule nor below the single line in the SC rule.
>
> • – y is an arbitrary variable in VARO.

Before defining the notion of provability in our sequent calculus, we recall that the decomposition rules of DRS were tailored to sequences of signed formulae in normal form, and the decomposition tree of such a sequence Ω started with the normal form $n(\Omega)$ of the sequence. Thus, our derived inference rules will be applicable to sequents in normal form, which is defined in analogy to the normal form of sequences of signed formulae; hence, like in the DRS system, the deduction mechanism of SC is applied to the normal form of a sequent instead of the original sequent.

More formally, the *normal form* of a sequent $\mathbf{S}$ of the form $\Gamma \vdash \Delta$ is the sequent $n(\mathbf{S})$ of the form $\mathbf{n}(\Gamma) \vdash \mathbf{n}(\Delta)$, where $\mathbf{n}(\Gamma), \mathbf{n}(\Delta)$ are obtained from Γ, Δ by replacing every term A occurring in these sets by its normal form $n(A)$, defined with respect to the sequence $\mathbf{N}(\Gamma), \mathbf{T}(\Delta)$ in the way described in Section 6.

We say that a sequent $\mathbf{S}$ is *provable*, and write $\vdash_{SC} \ S$, iff $n(\mathbf{S})$ can be derived from the axioms of the sequent calculus given in Table 4 by a finite number of applications of the inference rules given in Tables 5 – 7. As above, such a definition is justified by the fact that $n(\mathbf{S})$ can be obtained from $\mathbf{S}$ by a simple algorithm, and $\mathbf{S}$ is valid iff $n(\mathbf{S})$ is.

Clearly, all SC rules are sound, and all axioms are valid; this follows from Lemmas 8.1 and 8.2. Hence,

Theorem 8.3. *Any provable sequent is valid, i.e.* $\vdash_{SC} \ (\Gamma \vdash \Delta)$ *implies* $\models \ (\Gamma \vdash \Delta)$.

Proof. Suppose the sequent $\mathbf{S}$ is provable; then its normal form, $n(\mathbf{S})$, can be derived in a finite number of steps from the axioms of SC. The axioms are valid sequents, and all the derivation steps consist in applying the inference rules of SC which lead from valid sequents to valid sequents; it follows that $n(\mathbf{S})$ is valid. Since $n(\mathbf{S})$ is semantically equivalent to $\mathbf{S}$, the latter sequent is valid as well. $\square$

Theorem 8.4. *Any valid sequent is provable, i.e.* $\models\ (\Gamma \vdash \Delta)$ *implies* $\vdash_{SC}\ (\Gamma \vdash \Delta)$.

Proof. If a sequent $\Gamma \vdash \Delta$ is valid, then the corresponding sequence $\mathbf{N}(\Gamma), \mathbf{T}(\Delta)$ is also valid. By the completeness theorem for the DRS system, this means that this sequence has a finite decomposition tree DT with only axiomatic sequences at its leaves. Based on this tree, it is straightforward to construct a proof of the normal form of the original sequent in the SC calculus. corresponding An outline is as follows: We start from the terminal nodes; these are labeled by axiomatic sequences, so the corresponding sequents can be derived from the axioms of the sequent calculus by means of the thinning rules. Now we go upwards in DT, replacing each downward application of a decomposition rule by an upward application of the corresponding sequent calculus rule. For a detailed proof in case of an analogous similarity logic, the reader is referred to Konikowska (1997a). ☐

Since syntactic entailment $\vdash$ in a sequent corresponds to the semantic consequence relation, Theorem 8.4 means that the deduction system we have developed in this section is complete for theories:

Theorem 8.5. *If* $Ax = \{F_1, F_2, \ldots, F_n\}$ *is a finite subset of* FORM, *then, for any* $F \in$ FORM, $Ax \models F$ *iff the sequent* $Ax \vdash F$ *is provable in SC.*

9 Summary

Each information systems gives rise to various types of information relations on the object set. In his paper, we have studied relations $comp(A)$ of complementarity, parameterised by the set of subsets of the attribute set in such a way that for any family $\mathcal{A}$ of nonempty subsets of ATTR,

$$comp\left(\bigcup \mathcal{A}\right) = \bigcap_{A \in \mathcal{A}} comp(A).$$

We have presented multimodal logics for these families of complementarity relations along with two sound and complete deduction systems. The first, a Rasiowa–Sikorski style calculus, involved the use of variables to model both the membership relation and modalities, and representation of the modality parameter A as a union of components. This enabled us to replace formulae containing modalities of the form $[A], \langle A \rangle$ by formulae with modalities $[C], \langle C \rangle$ parameterised by atomic, disjoint sets. Thus, we could reduce reasoning about formulae with arbitrary modalities to reasoning about simple signed formulae of the form $\mathbf{T}(x\ cp(C)\ y), \mathbf{N}(x\ cp(C)\ y)$, where C is a component and x, y are individual variables, avoiding the usual problems arising in case of multiple accessibility relations. It should be noted that the DRS modality rules cannot be expressed as Hilbert–style axioms within propositional modal logic: In the DRS rules, the use of variables to model quantification implicit in modalities relies heavily on the fact that these rules model semantic consequence on the validity level, whereas Hilbert axioms model consequence on the truth level.

The second proof system was a Gentzen sequent calculus which does not use signed formulas. It was obtained observing that a sequent $\Gamma \vdash \Delta$ is valid whenever the sequence $\mathbf{N}(\Gamma), \mathbf{T}(\Delta)$ of signed formulae is valid. This led to a straightforward translation of the RS calculus DRS to the sequent calculus SC, and soundness and completeness were obtained in analogy.

Acknowledgement

We should like to thank the anonymous referee for thoughtful and constructive remarks.

References

Beth, E. (1959). The Foundations of Mathematics. Amsterdam: North Holland.

Blackburn, P. (1998). Internalizing labeled deduction. Tech. Rep. 102, Computerlinguistik, Universität des Saarlandes.

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

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

Düntsch, I. & Orłowska, E. (2000). Logics of complementarity in information systems. *Mathematical Logic Quarterly*, **46**. To appear.

Fitting, M. C. (1983). Proof Methods for Modal and Intutionistic Logics. Dordrecht: Reidel.

Gabbay, D. (1981). An irreflexivity lemma with applications to axiomatizations of conditions on tense frames. In U. Mönnich (Ed.), *Aspects of Philosophical Logic*, 67–89. Dordrecht: D. Reidel Publishing Co.

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

Konikowska, B. (1997b). A logic for reasoning about similarity. In E. Orłowska (Ed.), *Incomplete Information – Rough Set Analysis*, 462–491. Heidelberg: Physica – Verlag.

Lipski, W. (1976). Informational systems with incomplete information. In S. Michaelson & R. Milner (Eds.), *Third International Colloquium on Automata, Languages and Programming*, 120–130, University of Edinburgh. Edinburgh University Press.

Orłowska, E. (1995). Information algebras. In *Proceedings of AMAST 95*, vol. 639 of *Lecture Notes in Computer Science*. Springer–Verlag.

Orłowska, E. & Pawlak, Z. (1987). Representation of nondeterministic information. *Theoretical Computer Science*, **29**, 27–39.

Pawlak, Z. (1973). Mathematical foundations of information retrieval. ICS Research Report 101, Polish Academy of Sciences.

Rasiowa, H. & Sikorski, R. (1963). The Mathematics of Metamathematics, vol. 41 of *Polska Akademia Nauk. Monografie matematyczne*. Warsaw: PWN.

Table 1: DRS: the decomposition rules for signed formulae I

($\mathbf{T}$)

$$\begin{array}{c} \Omega', \mathbf{T}(F), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in F), \Omega'' \end{array},$$

where F is not of the form $z \in G$ for any $z \in$ VARO, $G \in FORM$, $x \in$ VARO, and x does not occur above the double line,

($\mathbf{N}$)

$$\begin{array}{c} \Omega', \mathbf{N}(F), \Omega'' \\ \hline\hline \Omega', \mathbf{N}(y \in F), \Omega'', \mathbf{N}(F) \end{array}$$

where F is not of the form $z \in G$ for any $z \in$ VARO, $G \in FORM$, and y is any variable in VARO,

($\mathbf{T} \in \neg \mathbf{F}$)

$$\begin{array}{c} \Omega', \mathbf{T}(x \in \neg F), \Omega'' \\ \hline\hline \Omega', \mathbf{N}(x \in F), \Omega'' \end{array}$$

($\mathbf{N} \in \neg \mathbf{F}$)

$$\begin{array}{c} \Omega', \mathbf{N}(x \in \neg F), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in F), \Omega'' \end{array}$$

($\mathbf{T} \in \wedge$)

$$\begin{array}{c} \Omega', \mathbf{T}(x \in F \wedge G), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in F), \Omega'' \mid \Omega', \mathbf{T}(x \in G), \Omega'' \end{array}$$

($\mathbf{N} \in \wedge$)

$$\begin{array}{c} \Omega', \mathbf{N}(x \in F \wedge G), \Omega'' \\ \hline\hline \Omega', \mathbf{N}(x \in F), \mathbf{N}(x \in G), \Omega'' \end{array}$$

($\mathbf{T} \in \vee$)

$$\begin{array}{c} \Omega', \mathbf{T}(x \in F \vee G), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in F), \mathbf{T}(x \in G), \Omega'' \end{array}$$

($\mathbf{N} \in \vee$)

$$\begin{array}{c} \Omega', \mathbf{N}(x \in F \vee G), \Omega'' \\ \hline\hline \Omega', \mathbf{N}(x \in F), \Omega'' \mid \Omega', \mathbf{N}(x \in G), \Omega'' \end{array}$$

Table 2: DRS: the decomposition rules for signed formulae II

$(\mathbf{T} \in \langle \mathbf{A} \rangle)$

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

where y is an arbitrary variable in VARO,

$(\mathbf{N} \in \langle \mathbf{A} \rangle)$

$$\frac{\Omega', \mathbf{N}(x \in \langle A \rangle F), \Omega''}{\Omega', \mathbf{T}(x \in [A] \neg F), \Omega''}$$

$(\mathbf{T} \in [\mathbf{A}])$

$$\frac{\Omega', \mathbf{T}(x \in [A] F), \Omega''}{\Omega', \mathbf{N}(x\ cp(A)\ z), \mathbf{T}(z \in F), \Omega''},$$

where $z \in$ VARO, and z does not occur above the double line,

$(\mathbf{N} \in [\mathbf{A}])$

$$\frac{\Omega', \mathbf{N}(x \in [A] F), \Omega''}{\Omega', \mathbf{T}(x \in \langle A \rangle \neg F), \Omega''}$$

$(\mathbf{T}\ \mathbf{cp}(\mathbf{A} \cup \mathbf{B}))$

$$\frac{\Omega', \mathbf{T}(x\ cp(A \cup B)\ y), \Omega''}{\Omega', \mathbf{T}(x\ cp(A)\ y), \Omega'' \mid \Omega', \mathbf{T}(x\ cp(B)\ y), \Omega''}$$

$(\mathbf{N}\ \mathbf{cp}(\mathbf{A} \cup \mathbf{B}))$

$$\frac{\Omega', \mathbf{N}(x\ cp(A \cup B)\ y), \Omega''}{\Omega', \mathbf{N}(x\ cp(A)\ y), \mathbf{N}(x\ cp(B)\ y), \Omega''}$$

(sym $\in$)

$$\frac{\Omega', \mathbf{N}(x \in y), \Omega''}{\Omega', \Omega'', \mathbf{N}(x \in y), \mathbf{N}(y \in x)} \qquad (*)$$

(tran $\in$)

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

(sym cp)

$$\frac{\Omega', \mathbf{N}(x\ cp(C)\ y), \Omega''}{\Omega', \Omega'', \mathbf{N}(x\ cp(C)\ y), \mathbf{N}(y\ cp(C)\ x)} \qquad (*)$$

(tran $\in$ cp)

$$\frac{\Omega', \mathbf{N}(x \in y), \Omega'', \mathbf{N}(y\ cp(C)\ z), \Omega'''}{\Omega', \Omega'', \Omega''', \mathbf{N}(x \in y), \mathbf{N}(y\ cp(C)\ z), \mathbf{N}(x\ cp(C)\ z)} \qquad (*)$$

(3-tran cp)

$$\frac{\Omega', \mathbf{N}(xcp(C)y), \Omega'', \mathbf{N}(y\ cp(C)\ z), \Omega''', \mathbf{N}(zcp(C)t), \Omega''''}{\Omega', \Omega'', \Omega''', \Omega'''', \mathbf{N}(xcp(C)y), \mathbf{N}(ycp(C)z), \mathbf{N}(zcp(C)t), \mathbf{N}(xcp(C)t),} (*)$$

Table 3: Additional DRS rules for finite ATTR

$$(\mathbf{T}\ \text{cp}\ -\mathbf{a})\qquad \frac{\Omega', \mathbf{T}(x\ cp(-\mathbf{a}\cap S)\ y), \Omega''}{\Omega', \mathbf{T}(x\ cp(\bigcup_{\mathbf{a}'\in\textsc{Cona}_{-\mathbf{a}}}\ \mathbf{a}'\cap S)\ y), \Omega''}$$

$$(\mathbf{N}\ \text{cp}\ -\mathbf{a})\qquad \frac{\Omega', \mathbf{N}(x\ cp(-\mathbf{a}\cap S)\ y), \Omega''}{\Omega', \mathbf{N}(x\ cp(\bigcup_{\mathbf{a}'\in\textsc{Cona}_{-\mathbf{a}}}\ \mathbf{a}'\cap S)\ y), \Omega''}$$

Table 4: Sequent axioms

Axiomatic subsequence: DRS	Sequent axiom: SC
$\mathbf{T}(F), \mathbf{N}(F)$	$\Delta, F \vdash \Gamma, F$
$\mathbf{T}(x \in x)$	$\Delta \vdash \Gamma, x \in x$
$\mathbf{N}(x\ cp(\mathbf{0})\ y)$	$\Delta, x\ cp(\mathbf{0})\ y \vdash \Gamma$
$\mathbf{N}(x\ cp(A)\ x)$	$\Delta, x\ cp(A)\ x \vdash \Gamma$
$\mathbf{N}(x\ cp(\mathbf{a}_j \cap \mathbf{S})\ y), \mathbf{N}(x'\ cp(\mathbf{a}_j \cap \mathbf{S}')\ y'),$ where $\mathbf{S} \neq \mathbf{S}'$ are subcomponents	$\Delta, x\ cp(\mathbf{a}_j \cap \mathbf{S})\ y, x'\ cp(\mathbf{a}_j \cap \mathbf{S}')\ y', \vdash \Gamma$ where $\mathbf{S} \neq \mathbf{S}'$ are subcomponents

Table 5: SC rules I

Decomposition rules: DRS	Sequent calculus rules: SC
$\begin{array}{c}\Omega', \mathbf{T}(F), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in F), \Omega''\end{array} \quad *$	$\dfrac{\Gamma \vdash \Delta, x \in F}{\Gamma \vdash \Delta, F} \quad *$
$\begin{array}{c}\Omega', \mathbf{N}(F), \Omega'' \\ \hline\hline \Omega', \mathbf{N}(y \in F), \Omega'', \mathbf{N}(F)\end{array} \quad **$	$\dfrac{\Gamma, y \in F \vdash \Delta}{\Gamma, F \vdash \Delta} \quad **$
$\begin{array}{c}\Omega', \mathbf{T}(x \in \neg F), \Omega'' \\ \hline\hline \Omega', \mathbf{N}(x \in F), \Omega''\end{array}$	$\dfrac{\Gamma, x \in F \vdash \Delta}{\Gamma \vdash \Delta, x \in \neg F}$
$\begin{array}{c}\Omega', \mathbf{N}(x \in \neg F), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in F), \Omega''\end{array}$	$\dfrac{\Gamma \vdash \Delta, x \in F}{\Gamma, x \in \neg F \vdash \Delta}$
$\begin{array}{c}\Omega', \mathbf{T}(x \in F \wedge G), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in F), \Omega'' \mid \Omega', \mathbf{T}(x \in G), \Omega''\end{array}$	$\dfrac{\Gamma \vdash \Delta, x \in F \qquad \Gamma \vdash \Delta, x \in G}{\Gamma \vdash \Delta, x \in F \wedge G}$
$\begin{array}{c}\Omega', \mathbf{N}(x \in F \wedge G), \Omega'' \\ \hline\hline \Omega', \mathbf{N}(x \in F), \mathbf{N}(x \in G), \Omega''\end{array}$	$\dfrac{\Gamma, x \in F, x \in G \vdash \Delta}{\Gamma, x \in F \wedge G \vdash \Delta}$
$\begin{array}{c}\Omega', \mathbf{T}(x \in F \vee G), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in F), \mathbf{T}(x \in G), \Omega''\end{array}$	$\dfrac{\Gamma \vdash \Delta, x \in F, x \in G}{\Gamma \vdash \Delta, x \in F \vee G}$
$\begin{array}{c}\Omega', \mathbf{N}(x \in F \vee G), \Omega'' \\ \hline\hline \Omega', \mathbf{N}(x \in F), \Omega'' \mid \Omega', \mathbf{N}(x \in G), \Omega''\end{array}$	$\dfrac{\Gamma, x \in F \vdash \Delta \qquad \Gamma, x \in G \vdash \Delta}{\Gamma, x \in F \vee G \vdash \Delta}$
$\begin{array}{c}\Omega', \mathbf{T}(x \in \langle A \rangle F), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(y \in F), \Omega'', \mathbf{T}(x \in \langle A \rangle F) \mid \Omega', \mathbf{T}(x\ cp(A) y), \Omega'', \mathbf{T}(x \in \langle A \rangle F)\end{array}$	$\dfrac{\Gamma \vdash \Delta, y \in F \qquad \Gamma \vdash \Delta, x\ cp(A)\ y}{\Gamma \vdash \Delta, x \in \langle A \rangle F}$
$\begin{array}{c}\Omega', \mathbf{N}(x \in \langle A \rangle F), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in [A]\neg F), \Omega''\end{array}$	$\dfrac{\Gamma \vdash \Delta, x \in [A]\neg F}{\Gamma, x \in \langle A \rangle F \vdash \Delta}$
$\begin{array}{c}\Omega', \mathbf{T}(x \in [A]F), \Omega'' \\ \hline\hline \Omega', \mathbf{N}(x\ cp(A)\ z), \mathbf{T}(z \in F), \Omega''\end{array} \quad ***$	$\dfrac{\Gamma, x\ cp(A)\ z \vdash \Delta, z \in F}{\Gamma \vdash \Delta, x \in [A]F} \quad ***$

Table 6: SC rules II

Decomposition rules: DRS	Sequent calculus rules: SC
$\begin{array}{c} \Omega', \mathbf{N}(x \in [A]F), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in \langle A \rangle \neg F), \Omega'' \end{array}$	$\dfrac{\Gamma \vdash \Delta, x \in \langle A \rangle \neg F}{\Gamma, x \in [A]F \vdash \Delta}$
$\begin{array}{c} \Omega', \mathbf{T}(x\ cp(A \cup B)\ y), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x\ cp(A)\ y), \Omega'' \mid \Omega', \mathbf{T}(x\ cp(B)\ y), \Omega'' \end{array}$	$\dfrac{\Gamma \vdash \Delta, x\ cp(A)\ y \qquad \Gamma \vdash \Delta, x\ cp(B)\ y}{\Gamma \vdash \Delta, x\ cp(A \cup B)\ y}$
$\begin{array}{c} \Omega', \mathbf{N}(x\ cp(A \cup B)\ y), \Omega'' \\ \hline\hline \Omega', \mathbf{N}(x\ cp(A)\ y), \mathbf{N}(x\ cp(B)\ y), \Omega'' \end{array}$	$\dfrac{\Gamma, x\ cp(A)\ y, x\ cp(B)\ y \vdash \Delta}{\Gamma, x\ cp(A \cup B)\ y \vdash \Delta}$
$\begin{array}{c} \Omega', \mathbf{N}(x \in y), \Omega'' \\ \hline\hline \Omega', \Omega'', \mathbf{N}(x \in y), \mathbf{N}(y \in x) \end{array}$	$\dfrac{\Gamma, y \in x \vdash \Delta}{\Gamma, x \in y \vdash \Delta}$
$\begin{array}{c} \Omega', \mathbf{N}(x \in y), \Omega'', \mathbf{N}(y \in F), \Omega''' \\ \hline\hline \Omega', \Omega'', \Omega''', \mathbf{N}(x \in y), \mathbf{N}(y \in F), \mathbf{N}(x \in F) \end{array}$	$\dfrac{\Gamma, x \in F \vdash \Delta}{\Gamma, x \in y, y \in F \vdash \Delta}$ ●
$\begin{array}{c} \Omega', \mathbf{N}(x\ cp(C)\ y), \Omega'' \\ \hline\hline \Omega', \Omega'', \mathbf{N}(x\ cp(C)\ y), \mathbf{N}(y\ cp(C)\ x) \end{array}$	$\dfrac{\Gamma, y\ cp(C)\ x \vdash \Delta}{\Gamma, x\ cp(C)\ y \vdash \Delta}$
$\begin{array}{c} \Omega', \mathbf{N}(x \in y), \Omega'', \mathbf{N}(y\ cp(C)\ z), \Omega''' \\ \hline\hline \Omega', \Omega'', \Omega''', \mathbf{N}(x \in y), \mathbf{N}(y\ cp(C)\ z), \mathbf{N}(x\ cp(C)\ z) \end{array}$	$\dfrac{\Gamma x\ cp(C)\ z \vdash \Delta}{\Gamma, x \in y, y\ cp(C)\ z \vdash \Delta}$ ●
$\begin{array}{c} \Omega', \mathbf{N}(x\ cp(C)\ y), \Omega'', \mathbf{N}(y\ cp(C)\ z), \Omega''', \mathbf{N}(z\ cp(C)\ t), \Omega'''' \\ \hline\hline \Omega', \Omega'', \Omega''', \mathbf{N}(x\ cp(C)\ y), \mathbf{N}(y\ cp(C)\ z), \mathbf{N}(z\ cp(C)\ t), \mathbf{N}(x\ cp(C)\ t) \end{array}$	$\dfrac{\Gamma, x\ cp(C)\ t \vdash \Delta}{\Gamma, x\ cp(C)\ y, y\ cp(C)\ z, z\ cp(C)\ t \vdash \Delta}$ ●

Table 7: SC rules III

Rules for finite ATTR: DRS	Rules for finite ATTR: SC
$\dfrac{\Omega', \mathbf{T}(x\ cp(-\mathbf{a} \cap \mathbf{S})\ y), \Omega''}{\Omega', \mathbf{T}(x\ cp(\bigcup_{\mathbf{a}' \in \textsc{Cona}-\mathbf{a}}\ \mathbf{a}' \cap S)\ y), \Omega''}$	$\dfrac{\Gamma \vdash \Delta, x\ cp(\bigcup_{\mathbf{a}' \in \textsc{Cona}-\mathbf{a}}\ \mathbf{a}' \cap S)\ y}{\Gamma \vdash \Delta, x\ cp(-\mathbf{a} \cap S)\ y}$
$\dfrac{\Omega', \mathbf{N}(x\ cp(-\mathbf{a} \cap S)\ y), \Omega''}{\Omega', \mathbf{N}(x\ cp(\bigcup_{\mathbf{a}' \in \textsc{Cona}-\mathbf{a}}\ \mathbf{a}' \cap S)\ y), \Omega''}$	$\dfrac{\Gamma, x\ cp(\bigcup_{\mathbf{a}' \in \textsc{Cona}-\mathbf{a}}\ \mathbf{a}' \cap S)\ y \vdash \Delta}{\Gamma, x\ cp(-\mathbf{a} \cap S)\ y \vdash \Delta}$
	Thinning rules
	$\dfrac{\Gamma \vdash \Delta}{\Gamma, F \vdash \Delta}, \quad \dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, F}$