Beata Konikowska

A Logic for Reasoning about Relative Similarity

Abstract. A similarity relation is a reflexive and symmetric binary relation between objects. Similarity is relative: it depends on the set of properties of objects used in determining their similarity or dissimilarity. A multi-modal logical language for reasoning about relative similarities is presented. The modalities correspond semantically to the upper and lower approximations of a set of objects by similarity relations corresponding to all subsets of a given set of properties of objects. A complete deduction system for the language is presented.

Key words: rough sets, entities, properties, similarity, lower and upper approximations, multi-modal logics, completeness

1. Introduction

The notion of similarity is widely employed in everyday life, as well as in science in general, and computer science in particular; we talk about similarity of two people, countries, cars, mathematical ideas, and so on. We have to do with similarity in case of image recognition, expert systems, or knowledge bases. In general, we deal with similarity in any situation involving measurements which can be performed with a limited accuracy only, because in such a situation we necessarily identify similar objects, i.e. the objects whose attributes yield the same values of measurements.

In mathematical terms, similarity is a binary relation which is reflexive and symmetric, but in general not transitive; hence it is not an equivalence relation. A good example is similarity of people with respect to age: two people are considered to be of similar age if the difference of their ages is at most 5 years. Such a relation is an example of a non-transitive 'threshold' similarity, in which two objects are considered as similar if the difference of their appropiate parameters does not exceed a given threshold value.

A moment's reflection will tell us that similarity is a relative notion: indeed, two objects may be similar in one aspect, but quite dissimilar in another aspect. Thus similarity is parametrised by a set of properties (attributes) of objects on which we base our comparison.

In this paper we will develop a complete formal system for reasoning about the notion of relative similarity understood in the above way, i.e.

*The paper was sponsored by the Committee of Scientific Research GrantNo 2 P301 007 04

as a reflexive and symmetric relation parametrised by a set of properties with respect to which the considered objects are judged as either similar or dissimilar. In fact, we shall talk about a whole family of similarity relations corresponding to various subsets of a given set of properties of objects. The hitherto existing formal systems (see [10], [11]) were aimed at modelling non-relative similarity only. The approach to similarity presented here stems from the rough set methodology (see [8]).

The language we discuss here is styled on the language introduced in [7] for reasoning about indiscernibility (i.e. similarity with the additional property of transitivity). However, no complete deduction system for the latter language has been proposed. The similarity language was first introduced by the author in [4], yet the deduction system presented there was complete for tautologies only, not for theories. Since from the practical point of view completeness for tautologies is not useful enough, in the present paper we develop a deduction system which is complete for theories. As the natural candidate we choose a sequent calculus system in Gentzen style, because such a system axiomatizes the consequence relation itself, and its completeness automatically implies completeness for finitely axiomatizable theories.

2. The universe and operations of lower and upper approximation

The semantic framework we work with is a *universe* U defined as follows:

$$U = \langle \textsc{Ent}, \textsc{Prop}, \{sim(P)\}_{P \subseteq \textsc{Prop}} \rangle \tag{1}$$

where $\textsc{Ent}$ is a set of *entities*, $\textsc{Prop}$ — a non-empty set of *properties*, and $\{sim(P)\}_{P \subseteq \textsc{Prop}}$ is a family of binary relations on $\textsc{Ent}$, called *similarity relations* and satisfying the following conditions:

(C1) $sim(\emptyset) = \textsc{Ent} \times \textsc{Ent}$;

(C2) each $sim(P)$ is reflexive and symmetric;

(C3) $sim(P \cup Q) = sim(P) \cap sim(Q)$ for any $P, Q \subseteq \textsc{Prop}$.

For any $P \subseteq \textsc{Prop}$, $sim(P)$ is referred to as the *similarity relation corresponding to the properties in* P, i.e. similarity with respect to the properties in P. It should be noted that we do not impose any structure whatsoever on the set $\textsc{Prop}$; in particular, we do not assume that elements of $\textsc{Prop}$ are to be identified with subsets of $\textsc{Ent}$: in our model, properties are simply *parameters* for similarity relations, and no other information about them is necessary.

Conditions (C1)–(C3) require some commment. Condition (C2) expresses the general properties of similarity relations discussed in the introduction. Condition (C1) says that any two entities are considered as similar with respect to an empty set of properties. Condition (C3) says that any two entities are similar with respect to the properties in $P \cup Q$ iff they are similar with respect to both the properties in P and the properties in Q. In other words, the bigger the set of properties, the smaller the similarity relation corresponding to this set.

EXAMPLE 2.1. A simple example of such an universe U is a triple

$$U = \langle \text{ENT}, \text{PROP}, \{sim(P)\}_{P \subseteq \text{PROP}} \rangle,$$

where ENT is the set of people living in a given town, and the set of properties is $\text{PROP} = \{age, fin, fam, res\}$, with fin standing for "financial status", fam — for "family status", and res — for "residence". The similarity relations $sim(p)$ for individual properties $p \in \text{PROP}$ are defined by

$$\begin{array}{lll} x\ sim(age)\ y & \text{iff} & |age(x) - age(y)| \leq 5 \text{ years}, \\ x\ sim(fin)\ y & \text{iff} & |income(x) - income(y)| \leq \$1000 \\ x\ sim(fam)\ y & \text{iff} & x, y \text{ are either both single or both married} \\ & & \text{and } |\#children(x) - \#children(y)| \leq 1, \\ x\ sim(res)\ y & \text{iff} & x, y \text{ live within 2 miles of each other,} \end{array}$$

where $\#children$ denotes the number of children. The relations $sim(P)$ for other subsets $P \subseteq \text{PROP}$ are given by

$$sim(\emptyset) = \text{ENT} \times \text{ENT}, \qquad sim(P) = \bigcup_{p \in P} sim(p) \text{ for } P \subseteq \text{PROP}, P \neq \emptyset$$

It is easy to see that U defined as above satifies conditions (C1)–(C3) of our definition.

Within this framework, we consider the operations of the *upper* and *lower approximation* of a set of entities E with respect to the similarity relation $sim(P)$ defined as follows:

$$\begin{array}{lll} \overline{\mathbf{sim}}(P)E & = & \{e \in \text{ENT} : (\exists e')((e, e') \in sim(P) \text{ and } e' \in E)\}, \\ \underline{\mathbf{sim}}(P)E & = & \{e \in \text{ENT} : (\forall e')((e, e') \in sim(P) \text{ implies } e' \in E)\} \end{array} \qquad (2)$$

respectively. Thus $\underline{sim}(P)E$ consists of all the entities which are not similar with respect to the properties in P to any entity outside E, and $\overline{sim}(P)E$

— of all the entities which are similar to some entity in E. The lower approximation of E could be said to be the interior of E with respect to the similarity $sim(P)$, and the upper approximation — the closure of E with respect to $sim(P)$. However, since a similarity relation need not be transitive, the above operations do not strictly qualify for such names, because they need not be idempotent. Obviously, we have

$$\underline{\mathbf{sim}}(P)E \subseteq E \subseteq \overline{\mathbf{sim}}(P)E$$

Both the above operations are very important in the rough set theory (see e.g. [8]), because they reflect an important feature of the situation we often encounter in practice, e.g. when we have to classify objects according to necessarily inexact measurements. Then we cannot hope to find the set E itself: instead of it, we can only consider either its upper or lower approximation (the latter being identical with the complement of the upper approximation of the complement of E, see below). This is because we cannot distinguish similar objects which yield the same measurements. Taking an element in $\underline{\mathbf{sim}}(P)E$, we are sure it will belong to E itself. However, if we limit our search to $\underline{\mathbf{sim}}(P)E$, we will disregard the 'border' elements of E, which are outside its inner core represented by the lower approximation. On the other hand, taking an element in $\overline{\mathbf{sim}}(P)E$ we may hope that it belongs to E (though we can never be sure of that, if the element does not belong to $\underline{\mathbf{sim}}(P)E$). Proceeding in this way we will not miss any elements in E, but we will probably consider by mistake some elements which are not in E.

A detailed discussion of lower and upper approximations can be found in [8, 7]. For our purposes, it suffices to quote the following basic properties of these operations:

$$\begin{array}{lcl}
\underline{\mathbf{sim}}(P)E & = & \textsc{Ent} - \overline{\mathbf{sim}}(P)(\textsc{Ent} - E) \\
\overline{\mathbf{sim}}(P \cup Q)E & \subseteq & \overline{\mathbf{sim}}(P)E \cap \overline{\mathbf{sim}}(Q)E \\
\overline{\mathbf{sim}}(P \cup Q)\{e\} & = & \overline{\mathbf{sim}}(P)\{e\} \cap \overline{\mathbf{sim}}(Q)\{e\} \\
\overline{\mathbf{sim}}(P)(E \cup F) & = & \overline{\mathbf{sim}}(P)E \cup \overline{\mathbf{sim}}(P)F
\end{array}$$

3. Syntax of the language

Having defined the semantic framework, we can now define a formal language for reasoning about similarity. The language in question will be parametrised by the set PROP. In other words, given a set PROP, we develop a language to talk about all universes of the form (1) with just this set of properties, an arbitrary set of entities ENT, and arbitrary similarity structure $\{sim(P)\}_{P \subseteq \textsc{Prop}}$. Thus we fix a set of properties PROP (possibly infinite).

The expressions of our language are built of symbols from the following sets:

$\textsc{Conp} = \{\mathbf{p} : p \in \textsc{Prop}\}$ — a set of unique, fixed constants representing individual properties, one constant $\mathbf{p}$ for each single property p; $\mathbf{p}$ is intended to represent p in any model.

$\mathbf{0}$ — a constant representing the empty set of properties
$\textsc{Varsp}$ — variables representing sets of properties
$\textsc{Vare}$ — variables representing individual entities
$\textsc{Varse}$ — variables representing sets of entities
$\{-, \cup, \cap\}$ — symbols for set-theoretic operations on sets of properties
$\{\neg, \vee, \wedge\}$ — symbols for set-theoretic operations on sets of entities
$\{\underline{sim}, \overline{sim}\}$ — symbols for the lower and upper approximations.

There are two kinds of expressions: terms, representing sets of properties, and formulas, representing sets of entities.

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

(i) $\textsc{Varsp} \cup \textsc{Conp} \cup \{\mathbf{0}\} \subseteq \textsc{Term}$,

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

The set $\textsc{Form}$ of formulas is the least set satisfying the following conditions:

(i) $\textsc{Vare} \cup \textsc{Varse} \subseteq \textsc{Form}$,

(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 $\underline{sim}(A)F, \overline{sim}(A)F \in \textsc{Form}$.

In what follows we shall also use a derived constructor $\longrightarrow$ defined by

$$F \longrightarrow G \overset{df}{\equiv} \neg F \vee G,$$

where $\overset{df}{\equiv}$ should be read as 'identical by definition'.

4. Semantics of the language

The semantics of the language we have introduced is defined in terms of the interepretation of terms and formulas in a model.

By a *model* we mean any pair $M = \langle U, v\rangle$, where $U = \langle \textsc{Ent}, \textsc{Prop}, \{sim(P)\}_{P \subseteq \textsc{Prop}}\rangle$ is a universe defined as in (1) of Section 2 (with $\textsc{Prop}$ being the parameter of our language), and v is a (multi-sorted) valuation such that:

$$v(P) \subseteq \textsc{Prop} \text{ for } P \in \textsc{Varsp}, \qquad v(E) \subseteq \textsc{Ent} \text{ for } E \in \textsc{Varse}$$
$$v(x) \in \textsc{Ent} \text{ for } x \in \textsc{Vare}.$$

For any model $M = \langle U, v\rangle$, the *interpretation of terms in* M is a function $\tau_M : \mathrm{TERM} \longrightarrow 2^{\mathrm{PROP}}$ defined inductively as follows:

(i) $\tau_M(\mathbf{p}) = \{p\}$ for any $p \in \mathrm{PROP}$, where $\mathbf{p}$ is the unique constant in CONP representing p, and $\tau_M(\mathbf{0}) = \emptyset$,

(ii) $\tau_M(P) = v(P)$ for $P \in \mathrm{VARSP}$,

(iii) for any $A, B \in \mathrm{TERM}$, $\tau_M(-A) = \mathrm{PROP}-\tau_M(A), \tau_M(A \cup B) = \tau_M(A) \cup \tau_M(B), \tau_M(A \cap B) = \tau_M(A) \cap \tau_M(B)$,

and the *interpretation of formulas in* M is a function $\varphi_M : \mathrm{FORM} \longrightarrow 2^{\mathrm{ENT}}$ such that:

(i) $\varphi_M(x) = \{v(x)\}$ for any $x \in \mathrm{VARE}$,

(ii) $\varphi_M(E) = v(E)$ for any $E \in \mathrm{VARSE}$,

(iii) for any $F, G \in \mathrm{FORM}$, $\varphi_M(\neg F) = \mathrm{ENT}-\varphi_M(F), \varphi_M(F \vee G) = \varphi_M(F) \cup \varphi_M(G), \varphi_M(F \wedge G) = \varphi_M(F) \cap \varphi_M(G)$

(iv) for any $A \in \mathrm{TERM}$ and for any $F \in \mathrm{FORM}$, $\varphi_M(\underline{sim}(A)F) = \underline{\mathbf{sim}}(\tau_M(A))\varphi_M(F), \varphi_M(\overline{sim}(A)F) = \overline{\mathbf{sim}}(\tau_M(A))\varphi_M(F)$, where $\underline{\mathbf{sim}}$, $\overline{\mathbf{sim}}$ are the operations of lower and upper approximations defined in Section 2 (see (2)).

A formula $F \in \mathrm{FORM}$ is said to be *true in a model* M, written $\models_M F$, iff it evaluates to the whole entity-universe of this model, i.e. iff $\varphi_M(F) = \mathrm{ENT}$. A formula F is said to be *valid* iff $\models_M F$ for every model M.

It can be easily seen that in general neither the formula F nor the formula $\neg F$ holds in a given model incorporating a valuation. Thus our logic is certainly non-classical, since it lacks the classical dichotomy 'either $M \models \varphi$ or $M \models \neg\varphi$', where M is a model incorporating a valuation. Moreover, the semantics of formulas is not compositional with respect to the satisfiablity relation, since the 'satisfaction' of a formula in a model cannot be defined in terms of satisfaction of its subformulas (see e.g. $F \vee G$). The reason for this is very simple: the satisfiability relation is defined in a way implying universal quantification (F is satisfied iff *every* entity is in $\varphi_M(F)$) – but we have no syntactic means for expressing quantification in the language, so the latter is not reflected in any way by the structure of the formula. This is one stumbling block to be overcome when defining the deduction system. The other is parametrisation of the approximation constructs $\underline{sim}, \overline{sim}$ by arbitrary terms, and the fact that these approximations correspond to a limited universal and existential quantification — again without an explicit use of quantifiers.

5. Signed formulas and the method of developing the deduction system

As we have mentioned in the introduction, we are going to develop a Gentzen calculus system complete for theories for the language introduced in the preceding section. To this end we shall use a method analogous to that employed in [5] (following [3]) for developing a sequent calculus for three-valued logic out of a Beth's tableau system ([1]). An important tool to be used in that process are 'signed formulas', representing 'truth' and 'non-truth' of an ordinary formula of our language. The Gentzen system will be developed out of a system of decomposition rules for signed formulas (see below) in Rasiowa-Sikorski style ([9]), obtained by extending the weakly complete deduction system for similarity logic given in [4]. It should be noted that though the introduction of signed formulas extends our language, yet the extension is only temporary: signed formulas are dropped after serving their purpose, and the goal Gentzen system we obtain at the end is a deduction system for the original language

Syntactically, signed formulas are simply formulas in $\textsc{Form}$ preceded by one of the signed formula constructors $\mathbf{T}, \mathbf{NT}$. Thus the set of signed formulas $\textsc{Sform}$ is

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

where $\mathbf{T}$ stands for 'true', and $\mathbf{NT}$ – for 'not true'.

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

$$\sigma_M(\mathbf{T}(F)) = \begin{cases} \mathbf{tt} & \text{iff } \models_M F, \\ \mathbf{ff} & \text{in the opposite case,} \end{cases}$$

$$\sigma_M(\mathbf{NT}(F)) = \begin{cases} \mathbf{tt} & \text{iff non } \models_M F, \\ \mathbf{ff} & \text{in the opposite case.} \end{cases}$$

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 $\models G$, iff $\models_M G$ for any model M.

In [4] we have also introduced the following notational abbreviations for some special types of formulas which played a prominent part in our deduction system:

$$x \in F \stackrel{df}{\equiv} x \longrightarrow F, \qquad x \notin F \stackrel{df}{\equiv} x \in \neg F \tag{3}$$

$$x\ sim(A)\ y \stackrel{df}{\equiv} x \in \overline{sim}(A)y, \qquad x\ dis(A)\ y \stackrel{df}{\equiv} x \notin \overline{sim}(A)y \tag{4}$$

where $x, y \in \text{VARE}, F \in \text{FORM}, A \in \text{TERM}$, and $F \longrightarrow G \equiv \neg F \vee G$.

It can be easily seen that since

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

the formulas (3), (4) actually have the semantic properties implied by their notation ('dis' is to mean 'dissimilar'). Thus for the corresponding signed formulas we have:

$$\models_M \ \mathbf{T}(x \in F) \text{ iff } v(x) \in \varphi_M(F) \tag{5}$$

$$\models_M \ \mathbf{T}(x \notin F) \text{ iff } v(x) \notin \varphi_M(F) \text{ iff } \models_M \ \mathbf{NT}(x \in F) \tag{6}$$

$$\models_M \ \mathbf{T}(x \; sim(A) \; y) \text{ iff } (v(x), v(y)) \in sim(\tau_M(A)) \tag{7}$$

$$\models_M \ \mathbf{T}(x \; dis(A) \; y) \text{ iff } (v(x), v(y)) \notin sim(\tau_M(A)) \text{ iff } \models_M \mathbf{NT}(x \; sim(A) \; y) \tag{8}$$

$$\models_M \ \mathbf{T}(x \in y) \text{ iff } v(x) = v(y) \tag{9}$$

$$\models_M \ \mathbf{T}(x \notin y) \text{ iff } v(x) \neq v(y) \text{ iff } \models_M \mathbf{NT}(x \in y) \tag{10}$$

Indeed: in (5) we have $\varphi_M(x \in F) = \varphi_M(\neg x \vee F) = (\text{ENT} - \{v(x)\}) \cup \varphi_M(F)$, whence $\models_M \ \mathbf{T}(x \in F)$ iff $\varphi_M(x \in F) = \text{ENT}$ iff $v(x) \in \varphi_M(F)$, and (6)-(10) are just special cases of (5).

In view of (6), (8), (10), when dealing with signed formulas we can dispense with formulas of the type $\mathbf{T}(x \notin F), \mathbf{T}(x \notin y), \mathbf{T}(x \, dis(A) \, y)$, replacing them by $\mathbf{NT}(x \in F), \mathbf{NT}(x \in y), \mathbf{NT}(x \; sim(A) \; y)$, respectively. However, all other signed formulas appearing in (5)–(10) will again play a fundamental role in the deduction system we shall develop. It is just these formulas that allow us to overcome the difficulties connected with the lack of classical dichotomy in our original language. This is due to the following result:

LEMMA 5.1. *For any formula* $F \in \text{FORM}$*:*

- (i) $\mathbf{T}(F)$ *is valid iff* $\mathbf{T}(x \in F)$ *is valid, where* $x \in \text{VARE}$ *is any variable not occuring in* F*,*
- (ii) $\mathbf{NT}(F)$ *is valid whenever for any model* M *there is a variable* $x \in \text{VARE}$ *such that* $\mathbf{NT}(x \in F)$ *is true in* M*.*

PROOF. **(i)** The forward implication is quite obvious. Assume now that $\models \ \mathbf{T}(x \in F)$, where $x \in \text{VARE}$ does not occur in F. To prove that $\models \mathbf{T}(F)$, we argue by contradiction. Suppose there exists a model $M = \langle U, v \rangle$ with $U = \langle \text{ENT}, \text{PROP}, \{sim(P)\}_{P \subseteq \text{PROP}} \rangle$ such that non $\models_M \ \mathbf{T}(F)$. Then non $\models_M F$, i.e. $\varphi_M(F) \neq \text{Ent}$. Thus there exists an entity $e \in \text{Ent}$ with $e \notin \varphi_M(F)$. Define a model M' as follows: $M' = \langle U, v' \rangle$, where $v'(x) = e$ and $v'(\xi) = v(\xi)$ for any argument $\xi \neq x$. As x does not occur in F, then, obviously, $\varphi_{M'}(F) = \varphi_M(F)$. Thus by (3) non $\models_{M'} (x \in F)$, because $v'(x) = e \notin \varphi_M(F) = \varphi_{M'}(F)$, which contradicts the validity of $\mathbf{T}(x \in F)$. Hence the backward implication must hold, too.

(ii) Assume the postulated condition is satified, and consider any model $M = \langle U, v \rangle$ with $U = \langle \text{Ent}, \text{Prop}, \{sim(P)\}_{P \subset \text{Prop}} \rangle$. Then by our assumption there exists an $x \in \text{Vare}$ such that $\models_M \mathbf{NT}(x \in F)$. Hence by (6) $v(x) \notin \varphi_M(F)$. Obviously, this yields $\varphi_M(F) \neq \text{Ent}$, whence non $\models_M F$, and in consequence $\models_M \mathbf{NT}(F)$. ■

6. DRS: a deduction system for signed formulas in Rasiowa-Sikorski style

Now we shall develop our intermediate deduction system, i.e. a Rasiowa-Sikorski (R-S) deduction system for the language of signed formulas. The R-S system consists of decomposition rules for sequences of signed formulas, and of fundamental sequences to be defined below. Using a more common terminology, we can say that the decomposition rules are the 'inference rules' of the system, whereas the fundamental sequences represent its 'axioms', or rather axiom schemes.

It should be noted that in essence an R-S system is dual to a tableau system of the type used by Beth ([1]) or Fitting ([2]). One basic difference is that in a tableau system we try to show that a formula is not satisfiable by showing that all the alternative ways of assuring its satisfaction lead to a contradiction. The latter holds iff the tableau of the formula — which is just a tree with vertices labeled by formulae and branches representing the above alternatives — is finite and closed, since a branch is closed if we encounter a contradiction. On the other hand, in the R-S system we try to show the validity of a formula or sequence of formulae. We construct a decomposition tree with vertices labeled by sequences of formulae whose branches terminate 'correctly' only if we encounter a simple, fundamental sequence of formulae – like $\mathbf{T}(F), \mathbf{NT}(F)$ — which is guaranteed to be valid. A sequence of formulae is valid iff it has a finite decomposition tree whose all branches end in fundamental 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 father (hence the labeling of vertices by sequences instead of individual formulae), and on applying the 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 formulas is said to be *true in a model* M, written $\models_M \Omega$, iff $\models_M G_i$ for some $i, 1 \leq i \leq n$. A sequence Ω is said to be *valid* iff $\models_M \Omega$ for every model M.

The reader should note here that our 'axioms', the fundamental sequences, will form a subclass of valid sequences, i.e. any fundamental sequence will be true in every model.

A decomposition rule is either a pair Ω_1, Ω_2 or a triple $\Omega_1, \Omega_2, \Omega_3$ of sequences of formulas in SFORM, written usually in the form

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

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, i.e. an inference rule of our system, is sound iff it leads from valid sequences to valid sequences in both directions - that is, both 'downwards' and 'upwards'. This is a strong property; to underline it, we separate the premises from the conclusion in the decomposition rules by a double line instead of a single one.

Before we introduce the actual rules, we need the notion of a normal form of a term and of a sequence of formulas. Namely, in order to cope with the modalities corresponding to various sets of properties, we must replace the terms appearing in signed formulas by unions of some special terms called 'components' which evaluate to a disjoint cover of PROP in any model (this is a syntactic counterpart of the well-known notion of components of a family of sets).

For an arbitrary sequence of signed formulas Ω, let

$$\text{CONP}(\Omega) = \{\mathbf{p} \in \text{CONP} : \mathbf{p} \text{ occurs in } \Omega\},$$
$$\text{VARSP}(\Omega) = \{P \in \text{VARSP} : P \text{ occurs in } \Omega\}$$

be the sets of all the constants in CONP and all the variables in VARSP that appear in the terms of Ω, respectively. Consider an arbitrary sequence Ω, and assume that

$$\text{CONP}(\Omega) = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}, \qquad \text{VARSP}(\Omega) = \{Q_1, \ldots, Q_m\},$$

where $n, m \geq 0$. The sequence Ω is said to be *nondegenerate* iff $m+n > 0$, i.e. iff it contains some symbols in VARSP $\cup$ CONP. [1] The notion of components

[1] The reader should note that $\mathbf{0}$ belongs neither to CONP nor to VARSP, whence any sequence whose terms contain no symbol for a property or sets of properties other than $\mathbf{0}$ is considered as degenerate.

will be defined for nondegenerate formulas only.

Assume that
$$A^{+} = A \text{ and } A^{-} = -A$$
for any term A, and that Ω is nondegenerate. We denote:

$$\textsc{Scomp}(\Omega) = \{Q_1^{i_1} \cap \ldots \cap Q_m^{i_m} : i_1, \ldots, i_m \in \{+, -\}\}$$

$\textsc{Comp}(\Omega) = \{\mathbf{p}_j \cap S : S \in \textsc{Scomp}(\Omega), 1 \leq j \leq n\} \cup$
$\{-\mathbf{p}_1 \cap \cdots \cap -\mathbf{p}_n \cap S : S \in \textsc{Scomp}(\Omega)\}$

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

The components and subcomponents have the following important properties:

LEMMA 6.1. *For any nondegenerate sequence Ω of signed formulas and any model M,*

1. *the sets $\{\tau_M(C)\}_{C \in \textsc{Comp}(\Omega)}$ form a disjoint cover of* PROP, *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) $\bigcup_{C \in \textsc{Comp}(\Omega)} \tau_M(C) = \textsc{Prop}$;
2. *the sets $\{\tau_M(S)\}_{S \in \textsc{Scomp}(\Omega)}$ also form a disjoint cover of* PROP;
3. *for any term A occuring in Ω, either*
   (i) *A is semantically equivalent to $N(A) \stackrel{df}{=} \mathbf{0}$, or*
   (ii) *there exists a unique subset $\{C_1, \ldots, C_k\}$ of $\textsc{Comp}(\Omega)$ such that the term*
   $$N(A) \stackrel{df}{=} C_1 \cup \cdots \cup C_k$$
   is semantically equivalent to A in all models.

PROOF (SKETCH). Consider first a slightly different set of 'component' terms, namely $\textsc{Comp}^{+}$, containing all the products of the form

$$\mathbf{p}_1^{j_1} \cap \cdots \cap \mathbf{p}_n^{j_n} \cap Q_1^{i_1} \cap \ldots \cap Q_m^{i_m}, \qquad (*)$$

where $i_k, j_l \in \{+, -\}$.

In any model M, the sets $\{\tau_M(C)\}_{C\in\textsc{Comp}^+}$ are simply set-theoretical components of the family of sets $\{\tau_M(\xi)\}_{\xi\in\textsc{Conp}(\Omega)\cup\textsc{Varsp}(\Omega)}$. Hence, by the well-known properties of components of a family of sets, the family $\{\tau_M(C)\}_{C\in\textsc{Comp}^+}$ forms a disjoint cover of $\textsc{Prop}$.

However, as for any two different constants $\mathbf{p}_i, \mathbf{p}_j$ in $\textsc{Conp}$ we have $\tau_M(\mathbf{p}_i \cap \mathbf{p}_j) = \{p_i\} \cap \{p_j\} = \emptyset$, then we can safely delete from $\textsc{Comp}^+$ all the products containing both $\mathbf{p}_i^+$ and $\mathbf{p}_j^+$ with $i \neq j$. Further, as $\tau_M(\mathbf{p}_1^- \cap \cdots \cap \mathbf{p}_{j-1}^- \cap \mathbf{p}_j^+ \cap \mathbf{p}_{j+1}^- \cap \cdots \cap \mathbf{p}_n^-) = \{p_j\} \cap \bigcap_{k\neq j}(\textsc{Prop} - \{p_k\}) = \{p_j\} = \tau_M(\mathbf{p}_j)$ for any model M, then any term $(*)$ of the form $\mathbf{p}_1^- \cap \cdots \cap \mathbf{p}_{j-1}^- \cap \mathbf{p}_j^+ \cap \mathbf{p}_{j+1}^- \cap \cdots \cap \mathbf{p}_n^- \cap S$ can be replaced by a component $p_j \cap S$ in $\textsc{Comp}(\Omega)$. Hence $\textsc{Comp}^+$ is semantically equivalent to $\textsc{Comp}(\Omega)$, which means that in any model M the family $\{\tau_M(C)\}_{C\in\textsc{Comp}(\Omega)}$ forms a disjoint cover of $\textsc{Prop}$.

2: The proof follows easily from Point 1, because $\textsc{Scomp}$ coincides with $\textsc{Comp}$ for $n = 0$.

3: Evidently, any term A appearing in Ω is a certain Boolean combination of the constants in $\textsc{Conp}(\Omega)$, variables in $\textsc{Varsp}(\Omega)$ and eventually the constant $\mathbf{0}$, Hence, applying to A the well-known algorithm for transforming a propositional formula into complete disjunctive normal form (see e.g. [6]), we can transform this term to a semantically equivalent term A' which is a union of products of the form $r_1^{l_1} \cap \cdots \cap r_k^{l_k}$, where $k = m + n + 1, l_i \in \{+, -\}, r_i \neq r_j$ for $i \neq r_j$, and each r_i is either $\mathbf{0}$, or an element of $\textsc{Conp}(\Omega) \cup \textsc{Varsp}(\Omega)$.

As in every model M we have $\tau_M(\mathbf{0}) = \emptyset$ and $\tau_M(-\mathbf{0}) = \textsc{Prop} - \tau_M(\mathbf{0}) = \textsc{Prop}$, then in every model M we also have

$$(1)\ \tau_M(\mathbf{0} \cap t) = \tau_M(\mathbf{0}),\ (2)\ \tau_M(-\mathbf{0} \cap t) = \tau_M(t),\ (3)\ \tau_M(\mathbf{0} \cup t) = \tau_M(t).$$

Using the above equalities, we can transform A' into a semantically equivalent term A'' being either $\mathbf{0}$, or a union of terms in the family $\textsc{Comp}^+$ defined above. [2]

If $A'' = \mathbf{0}$, then A is semantically equivalent to $\mathbf{0}$, and we take $N(A) = \mathbf{0}$; in the opposite case, proceeding in the way described above, we can replace A'' by an equivalent union $N(A) = C_1 \cup \cdots \cup C_n$ of components in $\textsc{Comp}(\Omega)$. The uniqueness of the set $\{C_1, \ldots, C_n\} \subseteq \textsc{Comp}(\Omega)$ is proved in a standard way, basing on the fact that any two different components evaluate to disjoint

[2] Remember that as Ω is nondegenerate, each product in A' must contain at least one element different than either $\mathbf{0}$ or $-\mathbf{0}$, whence (2) allows us to eliminate $-\mathbf{0}$ from A', and (1,3) imply that $\mathbf{0}$ cannot be eliminated from A' using these rules only if each product evaluates to $\mathbf{0}$, i.e. iff A' is semantically equivalent to $\mathbf{0}$.

sets in any model, and no component is semantically equivalent to $\mathbf{0}$ in all models. Hence Point 3 is satisfied, too. ∎

The term $N(A)$ appearing in the above lemma will be called *the normal form of* A (with respect to Ω). It is easy to see that every term A appearing in a degenerate sequence Ω is semantically equivalent to either $\mathbf{0}$ or $-\mathbf{0}$. In this case by the normal form of A, denoted $N(A)$, we shall accordingly mean either $\mathbf{0}$ or $-\mathbf{0}$.

The sequence obtained from any sequence Ω (degenerate or not) by replacing every term in Ω by its normal form with respect to Ω will be denoted by $N(\Omega)$ and called *the normal form of* Ω. It should be stressed that in our deduction system, the decomposition of a sequence of formulas will be preceded by transforming it to normal form. In other words,

All the decomposition rules will be applied to sequences in normal form only.

The reader should bear in mind that

FACT 6.2 *A sequence* Ω *can be transformed to normal form by means of a simple algorithm, and in any model the interpretation of* $N(\Omega)$ *coincides with the interpretation of* Ω.

Indeed: the algorithm for obtaining $N(\Omega)$ is just a simple modification of the standard algorithm of transformation into complete disjunctive normal form.

First let us define the 'axioms' of DRS, i.e. the class of fundamental sequences. A sequence Ω of formulas is said to be *fundamental* if it contains one of the formulas (i)–(iii) or pairs of formulas (iv)–(v) given below:

$$\textbf{(i)}\quad \mathbf{T}(x\ sim(\mathbf{0})\ y), \qquad \textbf{(ii)}\quad \mathbf{T}(x\ sim(A)\ x), \qquad \textbf{(iii)}\quad \mathbf{T}(x \in x),$$

$$\textbf{(iv)}\quad \mathbf{T}(F), \mathbf{NT}(F), \qquad \textbf{(v)}\quad \mathbf{T}(x\ sim(\mathbf{p}_j \cap S)\ y), \mathbf{T}(x'\ sim(\mathbf{p}_j \cap S')\ y'),$$

where S, S' are two different subcomponents in $\mathrm{S{\scriptstyle COMP}}(\Omega)$.

Due to (C1), (C2) and (5), formulas (i)-(iii) are obviously valid; there is no doubt that the pair (iv) is valid, too. Since the validity of (v) may be not so obvious, let us prove it formally. Consider any model M. As S, S' are two different subcomponents, then, by Lemma 6.1, the sets $\tau_M(S)$ and $\tau_M(S')$ are disjoint. Since $\tau_M(\mathbf{p_j}) = \{v(\mathbf{p_j})\}$ is a singleton, this means that either $\tau_M(\mathbf{p_j}) \cap \tau_M(S) = \emptyset$ or $\tau_M(\mathbf{p_j}) \cap \tau_M(S') = \emptyset$. In other words, either

$\tau_M(\mathbf{p_j} \cap S) = \emptyset$ or $\tau_M(\mathbf{p_j} \cap S') = \emptyset$. As by (C1) $\mathrm{sim}(\emptyset) = \textsc{Ent} \times \textsc{Ent}$ (where ENT is the set of entities of the model M), then either $sim(\tau_M(\mathbf{p_j} \cap S)) = \textsc{Ent} \times \textsc{Ent}$ or $sim(\tau_M(\mathbf{p_j} \cap S')) = \textsc{Ent} \times \textsc{Ent}$. In the first case we obviously have $(v(x), v(y)) \in sim(\tau_M(\mathbf{p_j} \cap S))$, and in the second — $(v(x'), v(y')) \in sim(\tau_M(\mathbf{p_j} \cap S'))$. Thus by (7) one of the formulas in the pair must be true in M, whence the pair (v) is valid, too. As any sequence of formulas containing either a valid formula or a valid subsequence (say, a pair) of formulas is also valid, we have shown that:

Lemma 6.3. *Every fundamental sequence is valid*

To define properly the decomposition rules, we need one more notion: that of an indecomposable signed formula or sequence of signed formulas.

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

(i) either $\mathbf{T}(x \in E)$ or $\mathbf{NT}(x \in E)$, where $x \in \textsc{Vare}, E \in \textsc{Varse} \cup \textsc{Vare}$,

(ii) either $\mathbf{T}(x\ sim(C)\ y)$ or $\mathbf{NT}(x\ sim(C)\ y)$, where $x, y \in \textsc{Vare}$, and C is $\mathbf{0}, -\mathbf{0}$, or:

  (a) any component, if $\textsc{Prop}$ is infinite;

  (b) any positive component, if $\textsc{Prop}$ is finite.

Otherwise a formula is said to be *decomposable.*

The notion of an indecomposable formula is intended to encompass all the 'atomic' formulas, which cannnot be be broken down into simpler formulas by the decomposition rules we shall give here. Point (ii) is justified by the fact that when decomposing sequences in normal form, we split the unions of components into individual components which are considered as 'atomic'; clearly, $\mathbf{0}$ is also 'indecomposable', and $-\mathbf{0}$ can be encountered only in the normal form of a 'degenerate' formula, which has no components at all — whence we cannot split $-\mathbf{0}$ into anything smaller. The condition (b) in (ii) follows from the fact that in case of a finite $\textsc{Prop}$ we can eliminate all negative components of the form $-\mathbf{P} \cap S$ replacing them by a union of positive components of the form $\bigcup_{\mathbf{p}' \in \textsc{Conp}-\mathbf{P}}(\mathbf{p}' \cap S)$.

A sequence Ω of signed formulas is said to be *indecomposable* iff all its elements are indecomposable. It should be noted that to some indecomposable sequences we will still be able to apply certain rules of our deduction system — namely, certain 'expansion' rules closing the sequence under specific symmetry and transitivity laws. All the rules will be applied to the leftmost suitable formula (or a pair of formulas) in a sequence Ω, following an indecomposable prefix Ω' of Ω.

In [4] we have developed a set DR of decomposition rules for the formulas in FORM. The only formulas appearing in the old deduction system were those of the form (3) and (4). Obviously, for any formula $F \in$ FORM, $\models_M F$ iff $\models_M \mathbf{T}(F)$, Hence, considering (3)–(6), all the (DR) rules for ordinary formulas remain valid after executing the following replacements: replacing $x \in F$ by $\mathbf{T}(x \in F)$, $x \notin F$ by $\mathbf{NT}(x \in F)$, $x\, sim(A)\, y$ by $\mathbf{T}(x\, sim(A)\, y)$, and $x\, dis(A)\, y$ by $\mathbf{NT}(x\, sim(A)\, y)$. In this way we obtain a modified set DR$'$ of decomposition rules semantically equivalent to the old ones, but expressed in terms of the signed formulas.

Since the language of signed formulas contains also formulas which are not of the form $\mathbf{T}(x \in F)$ or $\mathbf{NT}(x \in F)$, we have to augment the (DR$'$) rules by two new rules dealing with such formulas. Thus we add the following two rules:

$$(\mathbf{T}) \quad \frac{\Omega', \mathbf{T}(F), \Omega''}{\Omega', \mathbf{T}(y \in F), \Omega''}, \quad \text{where } y \in \text{VARE does not occur above the double line,}$$

$$(\mathbf{NT}) \quad \frac{\Omega', \mathbf{NT}(F), \Omega''}{\Omega', \mathbf{NT}(x \in F), \Omega'', \mathbf{NT}(F)}, \quad \text{where } x \text{ is any variable in VARE,}$$

for any $F \in$ FORM which is not of the form $z \in G$ for any $z \in$ VARE and any $G \in$ FORM, and any two sequences Ω', Ω'' of signed formulas such that Ω' is indecomposable.

Evidently, the above rules are sound (by Lemma 5.1) and allow us to manage the awkward problem of 'neither F nor $\neg F$ holds' by reducing the original formulas (in repeating steps in case of $(\mathbf{NT})$) to our nice dichotomous formulas of the form $\mathbf{T}(x \in F)$, $\mathbf{NT}(x \in F)$. The reader should note that the above reduction is quite analogous to that used in first-order logics in case of quantifiers. This is because the semantic conditions for $\mathbf{T}(F), \mathbf{NT}(F)$ to be true involve implicit quantification over all entities (universal in case of $\mathbf{T}(F)$, and existential in case of $\mathbf{NT}(F)$).

The condition limiting application of the $(\mathbf{T}), (\mathbf{NT})$ rules to formulas which *are not* of the form $z \in G$ is aimed at avoiding infinite loops caused by an iterated application of these rules. Indeed: this would result in 'decomposing' $\mathbf{T}(F)$ first to $\mathbf{T}(x \in F)$, and then to $\mathbf{T}(y \in (x \in F))$, $\mathbf{T}(z \in (y \in (x \in F)))$, ... — where all the steps except the first are both unnecessary and undesirable.

The final set DRS of decomposition rules for the language of signed formulas, consisting of the DR$'$ rules augmented by the $(\mathbf{T})$ and $(\mathbf{NT})$ rules, is given on the next page. Formally, the rules decompose sequences of signed formulas, but actually they only act upon a single formula (or a pair of formulas), the rest serving as a context. In all the rules we assume that the initial subsequence Ω' is indecomposable; in other words, we apply the rule to the first decomposable formula or pair of formulas.

DRS: the decomposition rules for signed formulas

(T) $$\frac{\Omega', \mathbf{T}(F), \Omega''}{\Omega', \mathbf{T}(x \in F), \Omega''},$$
where F is not of the form $z \in G$ for any $z \in \text{VARE}$, $G \in FORM$, $x \in \text{VARE}$, and x does not occur above the double line,

(NT) $$\frac{\Omega', \mathbf{NT}(F), \Omega''}{\Omega', \mathbf{NT}(y \in F), \Omega'', \mathbf{NT}(F)}$$
where F is not of the form $z \in G$ for any $z \in \text{VARE}, G \in FORM$, and y is any variable in VARE,

(T ∈ ¬F) $$\frac{\Omega', \mathbf{T}(x \in \neg F), \Omega''}{\Omega', \mathbf{NT}(x \in F), \Omega''}$$

(NT ∈ ¬F) $$\frac{\Omega', \mathbf{NT}(x \in \neg F), \Omega''}{\Omega', \mathbf{T}(x \in F), \Omega''}$$

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

(NT ∈ ∧) $$\frac{\Omega', \mathbf{NT}(x \in F \wedge G), \Omega''}{\Omega', \mathbf{NT}(x \in F), \mathbf{NT}(x \in G), \Omega''}$$

(T ∈ ∨) $$\frac{\Omega', \mathbf{T}(x \in F \vee G), \Omega''}{\Omega', \mathbf{T}(x \in F), \mathbf{T}(x \in G), \Omega''}$$

(NT ∈ ∨) $$\frac{\Omega', \mathbf{NT}(x \in F \vee G), \Omega''}{\Omega', \mathbf{NT}(x \in F), \Omega'' \mid \Omega', \mathbf{NT}(x \in G), \Omega''}$$

($\mathbf{T} \in \overline{\mathbf{sim}}$) $$\frac{\Omega', \mathbf{T}(x \in \overline{sim}(A)F), \Omega''}{\Omega', \mathbf{T}(y \in F), \Omega'', \mathbf{T}(x \in \overline{sim}(A)F) \mid \Omega', \mathbf{T}(x\ sim(A) y), \Omega'', \mathbf{T}(x \in \overline{sim}(A)F)}$$
where y is an arbitrary variable in VARE,

$$(\mathbf{NT} \in \overline{\mathbf{sim}}) \qquad \frac{\Omega', \mathbf{NT}(x \in \overline{sim}(A)F), \Omega''}{\Omega', \mathbf{T}(x \in \underline{sim}(A)\neg F), \Omega''}$$

$$(\mathbf{T} \in \underline{\mathbf{sim}}) \qquad \frac{\Omega', \mathbf{T}(x \in \underline{sim}(A)F), \Omega''}{\Omega', \mathbf{NT}(x\; sim(A)\; z), \mathbf{T}(z \in F), \Omega''},$$

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

$$(\mathbf{NT} \in \underline{\mathbf{sim}}) \qquad \frac{\Omega', \mathbf{NT}(x \in \underline{sim}(A)F), \Omega''}{\Omega', \mathbf{T}(x \in \overline{sim}(A)\neg F), \Omega''}$$

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

$$(\mathbf{NT\ sim(A \cup B)}) \qquad \frac{\Omega', \mathbf{NT}(x\; sim(A \cup B)\; y), \Omega''}{\Omega', \mathbf{NT}(x\; sim(A)\; y), \mathbf{NT}(x\; sim(B)\; y), \Omega''}$$

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

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

$$(\mathbf{sym\ T\ sim}) \qquad \frac{\Omega', \mathbf{T}(x\; sim(C)\; y), \Omega''}{\Omega', \mathbf{T}(y\; sim(C)\; x), \Omega''} \qquad (*)$$

$$(\mathbf{sym\ NT\ sim}) \qquad \frac{\Omega', \mathbf{NT}(x\; sim(C)\; y), \Omega''}{\Omega', \mathbf{NT}(y\; sim(C)\; x), \Omega''} \qquad (*)$$

$$(\mathbf{tran}\ \in\ \mathbf{sim}) \qquad \frac{\Omega', \mathbf{NT}(x \in y), \Omega'', \mathbf{T}(y\; sim(C)\; z), \Omega'''}{\Omega', \Omega'', \Omega''', \mathbf{NT}(x \in y), \mathbf{T}(y\; sim(C)\; z), \mathbf{T}(x\; sim(C)\; z)} \qquad (*)$$

Additional rules for finite PROP

$$(\mathbf{T\ sim}\ -\mathbf{P}) \qquad \frac{\Omega', \mathbf{T}(x\; sim(-\mathbf{P} \cap S)\; y), \Omega''}{\Omega', \mathbf{T}(x\; sim(\bigcup_{\mathbf{p}' \in \mathrm{CONP}-\mathbf{P}} \mathbf{p}' \cap S)\; y), \Omega''}$$

$$(\mathbf{NT\ sim} - \mathbf{P}) \qquad \frac{\Omega', \mathbf{NT}(x\; sim(-\mathbf{P} \cap S)\; y), \Omega''}{\Omega', \mathbf{NT}(x\; sim(\bigcup_{\mathbf{p}' \in \mathrm{CONP}-\mathbf{P}} \mathbf{p}' \cap S)\; y), \Omega''}$$

In all the rules, $x, y, z \in$ VARE, $F, G \in$ FORM, the Ω's are sequences of signed formulas, with the prerequisite that Ω' is indecomposable, $A, B \in$

Term, C is a component, and S — a subcomponent. It should be noted that the only terms encountered in the actually decomposed formulas will be unions of components, since the decomposition process will be applied to sequences in normal form only — and that is why we have no rules for term constructors other than the union.

The rules for finite Prop (where $\mathbf{P} \subseteq$ Conp) are justified by the fact that in case of a finite Prop each negative component of the form $-\mathbf{P} \cap S$ (where $\mathbf{P} \subseteq$ Conp) can be replaced by a finite union of components of the form $\mathbf{p}' \cap S$ over $\mathbf{p}' \in$ Conp $- \mathbf{P}$.

A natural question which arises here is that about the relation of the DRS system presented here to Fitting's tableaux for propositional modal logic like B (which is the closest counterpart of our logic), S4 or S5. As we have already said, the Rasiowa-Sikorski deduction system (of which DRS is an example) viewed as a general proof mechanism is dual to the tableaux system. However, apart from that, the DRS system has little in common with Fitting's tableaux for the said modal logics. One basic difference is that his tableau rules for B, S4 and S5 deal with the case when there is only one accessibility relation — and we have a whole family of them. The other is that Fitting's modality rules are global rules modifying the labels of whole branches of the tableau at one go, whereas the DRS modality rules are local rules whose application to a vertex v of a decomposition tree does not influence in any way the labels of vertices lying above v.

As the (DRS) rules consist of the (DR$'$) rules (see preceding discussion), which are sound in view od being equivalent to the sound rules given in [4], augmented by the equally sound (T)-(NT) rules, then we have:

Lemma 6.4. *The decomposition rules in the (DRS) system are sound.*

Of course, the rules can be easily proved to be sound in a direct way, basing on the definition of interpretation, (5)-(7) and Lemma 5.1.

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

The two 'quantifier-like' rules, namely (**NT**) and ($\mathbf{T} \in \overline{sim}$), are in fact borderline cases: though they do not replace the original formula by a simpler one, yet they do introduce some simpler formulas into the sequence. Note that, despite a more complicated notation, a formula $\mathbf{T}(x \in F)$ is in fact simpler that $\mathbf{T}(F)$ — because the latter corresponds in terms of satisfaction to a universal quantification of the former.

It can be easily seen that, consistently with our intuitions regarding decomposability:

Replacement rules are applicable to decomposable sequences only.

The reader can easily check that at most one replacement rule 'matches' any decomposable formula. As in any rule the initial sequence Ω' preceding the actual place of the rule's application is indecomposable, we can easily conclude that

To any decomposable sequence of formulas we can apply at most one replacement rule; namely, the unique rule applicable to the leftmost decomposable formula in this sequence.

However, the fact that a sequence is indecomposable need not terminate the decomposition process. It is easy to see that:

Expansion rules can also be applied to indecomposable sequences

It should be noted that in case of expansion rules the uniqueness principle is not preserved: there may be several such rules applicable to a given sequence. To avoid ambiguity, later on we shall give a method of choosing an appropiate rule to be applied, as well as a variable in the quantifier-like rules.

7. Decomposition trees for sequences of formulas and completeness of the DRS system

Now we shall describe the mechanism of using the DRS system introduced in the preceding section to prove the validity of sequences of signed formulas, and show the completeness of the system.

The validity proofs in DRS consist in constructing decomposition trees for sequences of formulas, 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 fundamental sequences only — which 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 formulas. A decomposition rule R in DRS is said to be *correctly applicable to* Ω if one of the following conditions is satisfied:

(i) R is a replacement 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 pair of formulas lying to the left of the formula or formulas, to which R can be applied[3].

As there is at most one replacement rule applicable to any sequence of formulas, and (ii) uniquely defines the expansion rule which is correctly applicable to Ω, then

At most one decomposition rule is correctly applicable to a any given sequence Ω of signed formulas.

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

To assure an unambiguous choice of variables in the decomposition process, in the rest of this paper we assume that VARE is a well-ordered set with respect to some ordering $\leq$.

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

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

(ii) Let v labeled by Σ be an end vertex of a branch B of the tree constructed up to now. Then:

   (a) **we terminate the branch B at the vertex v** if either:

      (a1) Σ is a fundamental sequence, or

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

[3] If we denote the relation of 'lying to the left' by $<$, then we assume here that $F < (F, G)$ for any $F, G \in \Omega$ such that $F < G$, and that for any F_i, G_i with $F_i < G_i$, $i = 1, 2$, we have $(F_1, G_1) < (F_2, G_2)$ iff either $F_1 < F_2$ or $F_1 = F_2$ and $G_1 < G_2$.

(b) **otherwise we expand the branch B beyond v by attaching to that vertex**:

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

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

where in a rule involving choice of a new variable we choose the first variable with respect to the ordering $\leq$ which does not appear in Σ, whereas in subsequent applications of a rule involving choice of an arbitrary variable to the same formula we choose the variables one after another, in the order determined by $\leq$ (see [9] for the details).

The definition merits some comment. The decomposition tree starts with a single node, labeled 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 'axiom' (in the form of a fundamental sequence) at its end. On the other hand, in case (a2) no replacement rule is applicable to Σ and we cannot augment this sequence by applying any expansion rule — whence branch B cannot be extended beyond the vertex 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 from now on we can assume that $DT(\Omega)$ is uniquely determined by Ω.

Further, the construction of $DT(\Omega)$ is based on the normal form of Ω rather than on the original sequence itself. This is justified by the fact that the normal form of Ω can be effectively obtained from Ω, and that Ω is valid iff $N(\Omega)$ is. Since $N(\Omega)$ contains only terms in normal form, i.e. terms being either $\mathbf{0}$ or $-\mathbf{0}$ or unions of some components in $\textsc{Comp}(\Omega)$, then the rules $\mathbf{T}(\in sim \cup)$ and $\mathbf{NT}(\in sim \cup)$ suffice to break up all the formulas of the form $\mathbf{T}(x\ sim(A)\ y)$, $\mathbf{NT}(x\ sim(A)\ y)$ appearing in $DT(\Omega)$ into elementary formulas of the form $\mathbf{T}(x\ sim(C)\ y)$, $\mathbf{NT}(x\ sim(C)\ y)$, where C is either a component, or $\mathbf{0}$ or $-\mathbf{0}$. Were Ω not 'preprocessed' to normal form before the actual construction of $DT(\Omega)$, then in order to achieve that result we would have to augment the proof system by a full set of decomposition rules corresponding to the Boolean algebra axioms for terms. Of course, this would have considerably complicated the proof theory — so it was better to adopt such a 'shortcut' policy.

It is easy to see that $DT(\Omega)$ may be infinite — due to the (**NT**) and ($\mathbf{T} \in \overline{sim}$) rules. Moreover, it is also clear that

a vertex v of $DT(\Omega)$ is a terminal vertex (a leaf) iff its label Σ is either a fundamental sequence, or an indecomposable sequence closed under all the expansion rules [4].

From now on, sequences of signed formulas labeling the terminal vertices of $DT(\Omega)$ will be referred to as the *terminal sequences* of Ω.

The notion of provability in our system is as follows:

A sequence Ω of signed formulas is said to be *provable*, in symbols $\vdash_{DRS} \Omega$, iff its decomposition tree $DT(\Omega)$ is finite and all its terminal sequences are fundamental.

It is quite evident that our deduction system is sound, i.e. the following result holds:

LEMMA 7.1. *Every provable sequence Ω of signed formulas is valid.*

PROOF. Suppose Ω is provable. Then $DT(\Omega)$ is a finite tree whose root is labeled by $N(\Omega)$ – the normal form of Ω, and whose leaves are labeled by fundamental sequences only. Hence from the definition of the decomposition tree it follows that $N(\Omega)$ can be obtained from a finite set of fundamental sequences by applying the decomposition rules 'backwards' finitely many times. By Lemma 6.3, every fundamental sequence is valid. Since all the decomposition rules are sound, i.e. their conclusions are valid iff their premises are valid, this implies the validity of $N(\Omega)$, and hence the validity of Ω. ■

Now we are going to prove the converse result: namely, that every valid formula is provable, which amounts to the completeness of our proof system. Since the proof is just a simple modification of the completeness proof for the old DR system for ordinary formulas given in [4], we shall omit some details here, particularly those parts of the proof which amount to simple checks.

The cornerstone of the complexity proof is the following lemma:

LEMMA 7.2. *For any sequence Ω of signed formulas, a terminal sequence Σ in $DT(\Omega)$ is valid iff it is fundamental.*

[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 formulas to that sequence.

PROOF. Let Ω and Σ satisfy the assumption of the Lemma. The backward implication is obvious, since every fundamental sequence is valid. In order to show the forward implication — i.e. that Σ has to be fundamental in order to be valid — we argue by contradiction.

Suppose Σ is not fundamental. As Σ is a terminal sequence of $DT(\Omega)$, this implies that Σ is indecomposable and closed under all the expansion rules. Hence, by the definition of an indecomposable sequence, each element of Σ must have one of the following forms:

$$\mathbf{T}(x \in E), \mathbf{NT}(x \in E), \mathbf{T}(x\; sim(C)\; y), \mathbf{NT}(x\; sim(C)\; y),$$

where $x, y \in$ VARE, and C is either $\mathbf{0}$ or $-\mathbf{0}$, or any component if PROP is infinite, and any positive component if PROP is finite. Let

$$\text{CONP}(\Omega) = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}, \quad \text{VARSP}(\Omega) = \{Q_1, \ldots, Q_m\},$$

where $n, m \geq 0$. Denote CONP$(\Omega) = \mathbf{P}$. Then the terms C which appear in Σ are as follows:

(i) C can be $\mathbf{0}$, but C can be $-\mathbf{0}$ only if Ω is degenerate, i.e. if all the terms in Ω are Boolean combinations of $\mathbf{0}$'s, since by the proof of Lemma 6.1 the normal form of any term A with respect to a non-degenerate sequence Ω is different from $-\mathbf{0}$.

(ii) If $C \notin \{\mathbf{0}, -\mathbf{0}\}$ and PROP is infinite, then C can be either of the form $\mathbf{p}_j \cap S$ or of the form $-\mathbf{P} \cap S$, where $S \in$ SCOMP(Ω); if PROP is finite, then C can only be of the form $\mathbf{p}_j \cap S$.

The reader is reminded that any $S \in$ SCOMP(Ω), called a subcomponent, is of the form

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

where $i_j \in \{+, -\}$ and $A^+ = A,\ A^- = -A$ for any term A.

Moreover, the sequence Σ must be closed under all the expansion rules understood as transformations of formulas.

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

Let $\rho \subseteq$ VARE $\times$ VARE be a binary relation defined as follows:

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

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

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

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

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

with

$$U_H = \langle \textsc{Ent}_H, \textsc{Prop}, \{sim_H(P)\}_{P \subseteq \textsc{Prop}}\rangle$$

where

$$\textsc{Ent}_H = \textsc{Vare}_{/\rho^*}$$

is the set of all equivalence classes of ρ^*.

The multi-sorted valuation v is defined as follows:

- For any $x \in \textsc{Vare}$,
$$v(x) = [x]_{\rho^*},$$
where $[x]_{\rho^*}$ is the unique equivalence class of ρ^* which contains x. Obviously, for any $x, y \in \textsc{Vare}, x \neq y$, we have
$$v(x) = v(y) \text{ iff the formula } \mathbf{NT}(x \in y) \text{ occurs in } \Sigma.$$

- For any $E \in \textsc{Varse}$,
$$v(E) = \{v(x) : \mathbf{NT}(x \in E) \text{ occurs in } \Sigma\}.$$

- To define the valuation of the variables ranging over sets of properties, recall first that $\mathbf{p}_1, \ldots, \mathbf{p}_n$ and $Q_1, \ldots, Q_m$ are all distinct elements of, respectively, $\textsc{Conp}$ and $\textsc{Varsp}$ which occur in Ω; obviously, Σ cannot contain any other symbols from the latter sets. We consider the following two cases:

  (a) $\textsc{Prop}$ is infinite:
  In this case $P' = \textsc{Prop} - \{p_1, \ldots, p_n\}$ is also infinite. As the set of subcomponents $\textsc{Scomp}(\Omega)$ is finite, there exists a one-to-one mapping $\Phi : \textsc{Scomp}(\Omega) \longrightarrow P'$ with
  $$\Phi(S) \neq p_j \quad \text{for any } S \in \textsc{Scomp}(\Omega) \text{ and any } j,\ 1 \leq j \leq n.$$

As Σ is not fundamental, then for any $j, 1 \leq j \leq n$, there is at most one $S \in \textsc{Scomp}(\Omega)$ such that for some $x, y \in \textsc{Vare}$ the formula $\mathbf{T}(x\ sim(\mathbf{p}_j \cap S)\ y)$ occurs in Σ (because any sequence containing both $\mathbf{T}(x\ sim(\mathbf{p}_j \cap S)\ )$ and $\mathbf{T}(x'\ sim(\mathbf{p}_j \cap S')\ y')$, where $S \neq S'$, is fundamental by clause (v) of the definition of a fundamental sequence). 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 $Q \in \textsc{Varsp} - \{Q_1, \ldots, Q_m\}$ and

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

(b) $\textsc{Prop}$ is finite. In this case the only components that can occur in Σ are ones of the form $\mathbf{p}_j \cap S$, where $S \in \textsc{Scomp}(\Omega)$. The definition of $v|_{\textsc{Varsp}}$ is a simplification of that given above; we put $v(Q) = \emptyset$ for any $Q \in \textsc{Varsp} - \{Q_1, \ldots, Q_m\}$ and

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

One can show that for the interpretation of terms τ_H induced by the above valuation v we have:

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

(ii) If $\textsc{Prop}$ is infinite, then $\tau_H(-\mathbf{P} \cap S) = \{\Phi(S)\} \neq \{p_j\}$ for any $j, 1 \leq j \leq n$.

(Recall that by convention $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$, and $-\mathbf{P} = -\mathbf{p}_1 \cap \cdots \cap -\mathbf{p}_n$.)

Thus τ_H evaluates all the components occuring in Σ to singletons. This is the key property which helps us to define the family of similarity relations $\{sim_H(P)\}_{P \subseteq \textsc{Prop}}$ on $\textsc{Ent}_H \times \textsc{Ent}_H$ – the last element needed to complete the definition of the model. Indeed: due to the property (C3) of any universe U, similarity relations corresponding to individual properties in $\textsc{Prop}$ generate the whole family $\{sim(P)\}_{p \subseteq \textsc{Prop}}$; they can also be defined independently of each other, which solves the basic technical problem connected with modalities parametrized by arbitrary sets of properties.

We begin with defining a family of auxiliary relations $\{R(p)\}_{p \in \textsc{Prop}}$ (the relations of dissimilarity with respect to individual properties). We consider the following two cases:

$\textsc{Case}$ 1: Σ contains the term $-\mathbf{0}$. As we recall, this can happen only if Ω is degenerate. Hence in this case, for any formula of the form either

$\mathbf{T}(x\ sim(C)\ y)$ or $\mathbf{NT}(x\ sim(C)\ y)$ occuring in Σ, we have either $C = \mathbf{0}$ or $C = -\mathbf{0}$. We put

$$R(p) = \{(v(x), v(y)) : \text{ the formula } \mathbf{T}(x\ sim(-\mathbf{0})\ y) \text{ is in } \Sigma\}$$

for each $p \in \textsc{Prop}$.

Case 2. Σ does not contain $-\mathbf{0}$, whence, for any formula of the form either $\mathbf{T}(x\ sim(C)\ y)$ or $\mathbf{NT}(x\ sim(C)\ y)$ occuring in Σ, the term C is either a component or $\mathbf{0}$.

We have to consider two subcases:

(i) Prop **is infinite**. Then, for any $p \in \textsc{Prop}$, we define

$$R(p) = \begin{cases} \{(v(x), v(y)) : & \text{if } p = p_j, \text{ where } 1 \leq j \leq n \\ \qquad \mathbf{T}(x\ sim(\mathbf{p}_j \cap S_j)\ y) \text{ is in } \Sigma\} & \text{and } j \text{ is positive in } \Sigma, \\ \{(v(x), v(y)) : & \text{if } p = \Phi(S), \text{ where } S \in \\ \qquad \mathbf{T}(x\ sim(-\mathbf{P} \cap S)\ y) \text{ is in } \Sigma\} & \textsc{Scomp}(\Omega), \\ \emptyset & \text{otherwise.} \end{cases}$$

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

(ii) Prop **is finite**. The definiton is again a simplification of that for infinite Prop: we put

$$R(p) = \begin{cases} \{(v(x), v(y)) : & \text{if } p = p_j, \text{ where } 1 \leq j \leq n \\ \qquad \mathbf{T}(x\ sim(\mathbf{p}_j \cap S_j)\ y) \text{ is in } \Sigma\} & \text{and } j \text{ is positive in } \Sigma, \\ \emptyset & \text{otherwise.} \end{cases}$$

Having defined the auxiliary relations, we put

$$sim_H(\emptyset) = \textsc{Ent}_H \times \textsc{Ent}_H, \quad sim_H(p) = \textsc{Ent}_H \times \textsc{Ent}_H - R(p)$$

for any property $p \in \textsc{Prop}$, and

$$sim_H(P) = \bigcap_{p \in P} sim_H(p)$$

for any $\emptyset \neq P \subseteq \textsc{Prop}$.

It is easy to see that the family $\{sim_H(P)\}_{P \subseteq \textsc{Prop}}$ defined above is indeed a family of similarity relations which satisfies conditions (C1)-(C3) of Section 2. In consequence, $U_H = \langle \textsc{Ent}_H, \textsc{Prop}, \{sim_H(P)\}_{P \subseteq \textsc{Prop}} \rangle$ is a universe satisfying the conditions of Section 2, and $H = \langle \overline{U}, v \rangle$ is is a correctly defined model for our language.

The proof that non $\models_H \Sigma$ is by considering, one by one, all the types of signed formulas that can occur in Σ, i.e. $\mathbf{T}(x \in y), \mathbf{NT}(x \in y), \mathbf{T}(x \in$

$E), \mathbf{NT}(x \in E), \mathbf{T}(x sim(C) y)$, and $\mathbf{NT}(x sim(C) y)$ (where $x, y \in$ VARE, $E \in$ VARSP, C is either a component or $\mathbf{0}$ or $-\mathbf{0}$), and proving that they cannot be true in H. For example, if $\mathbf{NT}(x \in y)$ is in Σ, then by the definition of valuation we have $v(x) = v(y)$, whence by (10) non $\models_{M_H}$ $\mathbf{NT}(x \in y)$. When proving that the $'\mathbf{T}'$-type formulas are not true, we also make use of the fact that Σ is not fundamental. For example, if $\mathbf{T}(x \in y)$ is in Σ, then $x \neq y$, because Σ is not fundamental, and for the same reason $\mathbf{NT}(x \in y)$ cannot occur in Σ; as for $x \neq y$ we have $v(x) = v(y)$ iff the latter formula is in Σ, this yields $v(v) \neq v(y)$, whence by (9) non $\models_H$ $\mathbf{T}(x \in y)$. The proofs for the sim-type formulas follow from the equalites giving $v(C)$ for any component C that can occur in Σ, as well as from the definition of the 'dissimilarity' relations $R(p)$. The details of an analogous proof for the nonsigned formulas in FORM are given in [4].

In this way we arrive at a contradiction, whence a terminal sequence Σ must be fundamental in order to be valid. This ends the proof of the Lemma.

■

Now we can state the completeness theorem:

THEOREM 7.3. *Every valid sequence of signed formulas $\Omega \in$ SFORM is provable, i.e. it has a finite decomposition tree whose terminal sequences are all fundamental.*

PROOF. Suppose Ω is valid, and recall that the root of $DT(\Omega)$ is labeled 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 we have just proved, the later holds iff all of them are fundamental. 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 fundamental.

Hence to complete the proof we have to prove that if $DT(\Omega)$ is infinite, then Ω cannot be valid. We do it by modifying the by now standard proof used in [9] (p. 302) to suit the structure of our language. The details for the case of FORM can again be found in [4], so we shall give only the basic outline of the method here.

Suppose $DT(\Omega)$ is infinite. As $DT(\Omega)$ is a finitely-branching (in fact, binary) tree, then from Koenig's lemma it follows that it must possess an infinite branch B starting at the root. Let us denote by Δ the set of all indecomposable formulas which occur in the sequences labeling the vertices of B. It is quite evident that Δ cannot contain any finite fundamental sequence. Indeed: as any subsequence of Δ consists of indecomposable formulas only, and each vertex in $DT(\Omega)$ inherits all the indecomposable formulas from its ancestors, then such a hypothetical fundamental subsequence of Δ would be a subsequence of the label Ω' of some vertex v of B. This is impossible, because in such a case Ω' would be fundamental, too, and B would terminate at v instead of being infinite. In consequence, Δ can contain neither any of the three single formulae (i-iii) nor any of the two pairs of formulae (iv-v) (given on page 13) whose presence in a sequence makes the latter fundamental.

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

By definition of a decomposition tree, the top vertex of B (coinciding with the root of $DT(\Omega)$) is labeled by $N(\Omega)$, i.e. the normal form of Ω. Hence, arguing by induction on the order (i.e. complexity) of a formula, we can deduce from the fact that all indecomposable formulas in the labels of B are not true in H that $N(\Omega)$ is not true in H, too[5]. As $N(\Omega)$ is semantically equivalent to Ω, this implies Ω itself is not true in H.

A detailed reasoning follows the lines given in [4]. The only real difference is the case of formulas $\mathbf{T}$(F), $\mathbf{NT}$(F), where F is not of the form $x \in F'$, and $\mathbf{NT}(x \in \overline{sim}(A)F), \mathbf{NT}(x \in \underline{sim}(A)F)$, because the decomposition rules applied to these formulas yield formulas having formally a higher order that the original ones. However, after getting at most two steps down the branch B (by applying the appropiate decomposition rules to the these 'higher-order' formulas) we get either an indecomposable formula, or a formula of a lower order, which must be true in order that the original formula be true – and that allows the induction proof to go through.

The fact that non $\models_H \Omega$ shows that Ω is not valid, which ends the proof. ■

[5]Intuitively, this follows from the fact that the decomposition rules are designed to extract from a signed formula G all simpler formulas, which must be true in order that G be true. If the rule 'splits' the tree in two branches, then in order that G be true, the simpler formulas derived from G lying on both these branches must be true. In other words, in order that any formula G in the sequence $N(\Omega)$ labeling the top of B be true, all the simpler formulas on B extracted from the elements of G in the decomposition process must be true. Repeating this reasoning for the descendants of G, we conclude that in order that G be true, all the indecomposable formulas on B obtained in the long run from the elements of G must be true. As no indecomposable formulas are true in H, this implies that no G in $N(\Omega)$ can be true in H — and hence $N(\Omega)$ cannot be true in H

8. Sequents versus signed formulas

Now we can pass to our main task: developing a sequent calculus out of the complete proof system for sequences of signed formulas we have just described.

Since the sequent calculus is intended to be a deduction system for the original language described in [4], then the sequents will involve ordinary, non-signed formulas in FORM only. Sets of such formulas will be denoted by Γ, Δ with suitable indices. A model M of our language will be said to be a *model of a set* Γ iff every formula in Γ is true in M.

By a *sequent* we mean a pair (Γ, Δ) of finite sets Γ and Δ of formulas in FORM, written usually in the form $\Gamma \vdash \Delta$. In the sequent notation of the type $\Gamma, F \vdash \Delta, G$ used in the sequel, commas denote set-theoretical union, and individual formulas $F, G \in$ FORM are identified with respective singletons.

The sequent $\Gamma \vdash \Delta$ is said to be *valid* iff, for any model M of Γ, at least one formula $F \in \Delta$ is true in M. Thus a sequent $\Gamma \vdash \Delta$ is valid iff, for every model M such that every formula in Γ is true in M, some formula in Δ is also true in M.

For any finite set of formulas $Ax \subset$ FORM, and any formula $F \in$ 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 the 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 axiomatization of the sequent calculus will automatically provide a deduction system complete for theories.

It turns out that there is a simple connection between validity of sequents and validity of sequences of signed formulas. For any finite set $\Gamma = \{F_1, F_2, \ldots, F_k\} \subseteq$ FORM, let $\mathbf{T}(\Gamma)$ and $\mathbf{NT}(\Gamma)$ be sequences of signed formulas obtained by preceding all elements of Γ (taken in some standard order, e.g. 'alphabetical' order) with the operators $\mathbf{T}$ and $\mathbf{NT}$, respectively; i.e.

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

Then we have:

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

PROOF. Recall that a sequence of signed formulas is valid iff, for any model M, some formula in this sequence is true in M. By definition, a sequent $\Gamma \vdash \Delta$ is valid iff, for every model M such that every $F \in \Gamma$ is true in M, some $F' \in \Delta$ is true in M. In other words, $\Gamma \vdash \Delta$ is valid iff, for every model M, either some $F \in \Gamma$ is not true in M or some $F' \in \Delta$ is true in M — or, equivalently, for every M, either $\mathbf{NT}(F)$ is true in M for some $F \in \Gamma$ or $\mathbf{T}(F')$ is true in M for some $F' \in \Delta$. As the latter condition is tantamount to the validity of the sequence $\mathbf{NT}(\Gamma), \mathbf{T}(\Delta)$, this ends the proof. ■

Thus from Theorem 7.3 of the preceding section it follows that

FACT 8.2 *A sequent* $\Gamma \vdash \Delta$ *is valid whenever the sequence* $\mathbf{NT}(\Gamma), \mathbf{T}(\Delta)$ *of signed formulas has a finite decomposition tree with all terminal sequences fundamental.*

The above fact suggests a straightforward method of developing a sequent calculus for our original formal language basing on the DRS deduction system for sequences of signed formulas presented in the preceding section; 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.

However, to make the subsequent reasoning clear, let us define first what kind of a sequent calculus we have in mind. The sequent calculus (SC) to be developed here will consist of *axioms*, having the form of single (valid) sequents, and *inference rules*, leading from valid sequents to valid sequents. A deduction rule is of the form either

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

respectively. 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* (*premises*). A rule is said to be *sound* iff its conclusion is valid whenever all its premises are valid.

Thus we should bear in mind that there are two basic differences between the decomposition rules for signed formulas and the inference rules of the sequent calculus:

- The decomposition rules for signed formulas are two-way rules saying that the conclusion is valid iff all the premises are valid, whereas the sequent calculus rules are in principle one-way rules saying that the conclusion is valid whenever all the premises are valid.

- When constructing a proof, the decomposition rules were used 'downwards' to construct a decomposition tree of a sequence of signed formulas, whereas the sequent calculus rules will be used 'upwards' to deduce a sequent from the axioms.

For any sequence Ω of signed formulas, let us denote

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

Further, for the sake of brevity, let us call a decomposition tree 'valid' if it is finite and all its terminal sequences are fundamental.

Then Lemma 8.1 can be rephrased as follows: for any finite sequence Ω of signed formulas, the sequent $\Omega^- \vdash \Omega^+$ is valid iff Ω has a valid decomposition tree.

Now suppose that

$$\frac{\Omega', \Pi, \Omega''}{\overline{\Omega', \Sigma, \Omega''}}$$

is a decomposition rule in DRS. Then, evidently, for any finite sets $\Gamma, \Delta \subset \textsc{Form}$ such that $\mathbf{NT}(\Gamma)$ is indecomposable, the sequence $\mathbf{NT}(\Gamma), \Pi, \mathbf{T}(\Delta)$ has a valid decomposition tree whenever $\mathbf{NT}(\Gamma), \Sigma, \mathbf{T}(\Delta)$ has a valid decomposition tree. Moreover, the above equivalence holds even when $\mathbf{NT}(\Gamma)$ is decomposable. Indeed: as the DRS rules remain sound even after dropping the requirement that the initial sequence Ω' be indecomposable, then $\mathbf{NT}(\Gamma), \Pi, \mathbf{T}(\Delta)$ is obviously valid whenever $\mathbf{NT}(\Gamma), \Sigma, \mathbf{T}(\Delta)$ is, and in view of the completeness of DRS, validity is tantamount to having a valid decomposition tree.

In consequence, considering Lemma 8.1, the sequent $\Gamma, \Pi^- \vdash \Delta, \Pi^+$ is valid whenever the sequent $\Gamma, \Sigma^- \vdash \Delta, \Sigma^+$ is valid. In other words, we have:

Lemma 8.3. *For every rule*

$$\frac{\Omega', \Pi, \Omega''}{\overline{\Omega', \Sigma, \Omega''}}$$

in DRS,

$$\frac{\Gamma, \Sigma^- \vdash \Delta, \Sigma^+}{\Gamma, \Pi^- \vdash \Delta, \Pi^+}$$

is a valid inference rule of the sequent calculus.

For example, taking $\Pi \equiv \mathbf{T}(F)$ and $\Sigma \equiv \mathbf{T}(x \in F)$, we conclude that the decomposition rule of the form

$$\frac{\Omega', \mathbf{T}(F), \Omega''}{\overline{\Omega', \mathbf{T}(x \in F), \Omega''}},$$

where $x \in \textsc{Var}E$ is a new variable and F is not of the form $y \in F'$, gives rise to the following rule for deriving valid sequents:

$$\frac{\Gamma \vdash \Delta, x \in F}{\Gamma \vdash \Delta, F},$$

where $x \in \textsc{Var}E$, x does not occur in F, Δ or Γ, and F is not of the form $y \in F'$.

The situation of the splitting rules of the form

$$\frac{\Omega', \Pi, \Omega''}{\overline{\Omega', \Sigma_1, \Omega'' \mid \Omega', \Sigma_2, \Omega''}}$$

is quite analogous. In this case the sequence $\mathbf{NT}(\Gamma), \Pi, \mathbf{T}(\Delta)$ has a valid decomposition tree whenever both $\mathbf{NT}(\Gamma), \Sigma_1, \mathbf{T}(\Delta)$ and $\mathbf{NT}(\Gamma), \Sigma_2, \mathbf{T}(\Delta)$ have valid decomposition trees (the justification is similar as in case of single-premise rules). In consequence, the sequent $\Gamma, \Pi^- \vdash \Delta, \Pi^+$ is valid whenever the sequents $\Gamma, \Sigma_1^- \vdash \Delta, \Sigma_1^+$ and $\Gamma, \Sigma_2^- \vdash \Delta, \Sigma_2^+$ are both valid. Thus we have:

Lemma 8.4. *For every rule* $\dfrac{\Omega', \Pi, \Omega''}{\overline{\Omega', \Sigma_1, \Omega'' \mid \Omega', \Sigma_2, \Omega''}}$ *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.*

For example, taking $\Pi \equiv \mathbf{T}(x \in F_1 \wedge F_2)$ and $\Sigma_1 \equiv \mathbf{T}(x \in F_1)$, $\Sigma_2 \equiv \mathbf{T}(x \in F_2)$, we conclude that the decomposition rule of the form

$$\frac{\Omega', \mathbf{T}(x \in F_1 \wedge F_2), \Omega''}{\overline{\Omega', \mathbf{T}(x \in F_1), \Omega'' \mid \Omega', \mathbf{T}(x \in F_2), \Omega''}}$$

gives rise to the following rule for deriving valid sequents:

$$\frac{\Gamma \vdash \Delta, x \in F_1 \qquad \Gamma \vdash \Delta, x \in F_2}{\Gamma \vdash \Delta, x \in F_1 \wedge F_2}.$$

The above observations are summarized in the following table which gives the ready-made forms of inference rules for deriving valid sequents induced by typical decomposition rules encountered in the DRS system.

Decomposition rule	Inference rule for sequents
$\dfrac{\Omega', \mathbf{T}(F), \Omega''}{\Omega', \mathbf{T}(F_1), \Omega''}$	$\dfrac{\Gamma \vdash \Delta, F_1}{\Gamma \vdash \Delta, F}$
$\dfrac{\Omega', \mathbf{NT}(F), \Omega''}{\Omega', \mathbf{NT}(F_1), \Omega''}$	$\dfrac{\Gamma, F_1 \vdash \Delta}{\Gamma, F \vdash \Delta}$
$\dfrac{\Omega', \mathbf{T}(F), \Omega''}{\Omega', \mathbf{NT}(F_1), \Omega''}$	$\dfrac{\Gamma, F_1 \vdash \Delta}{\Gamma \vdash \Delta, F}$
$\dfrac{\Omega', \mathbf{NT}(F), \Omega''}{\Omega', \mathbf{T}(F_1), \Omega''}$	$\dfrac{\Gamma \vdash \Delta, F_1}{\Gamma, F \vdash \Delta}$
$\dfrac{\Omega', \mathbf{T}(F), \Omega''}{\Omega', \mathbf{T}(F_1), \mathbf{T}(F_2), \Omega''}$	$\dfrac{\Gamma \vdash \Delta, F_1, F_2}{\Gamma \vdash \Delta, F}$
$\dfrac{\Omega', \mathbf{NT}(F), \Omega''}{\Omega', \mathbf{NT}(F_1), \mathbf{NT}(F_2), \Omega''}$	$\dfrac{\Gamma, F_1, F_2 \vdash \Delta}{\Gamma, F \vdash \Delta}$
$\dfrac{\Omega', \mathbf{T}(F), \Omega''}{\Omega', \mathbf{NT}(F_1), \mathbf{T}(F_2), \Omega''}$	$\dfrac{\Gamma, F_1 \vdash \Delta, F_2}{\Gamma \vdash \Delta, F}$
$\dfrac{\Omega', \mathbf{T}(F), \Omega''}{\Omega', \mathbf{T}(F_1), \Omega'' \mid \Omega', \mathbf{T}(F_2), \Omega''}$	$\dfrac{\Gamma \vdash \Delta, F_1 \qquad \Gamma \vdash \Delta, F_2}{\Gamma \vdash \Delta, F}$
$\dfrac{\Omega', \mathbf{NT}(F), \Omega''}{\Omega', \mathbf{NT}(F_1), \Omega'' \mid \Omega', \mathbf{NT}(F_2), \Omega''}$	$\dfrac{\Gamma, F_1 \vdash \Delta \qquad \Gamma, F_2 \vdash \Delta}{\Gamma, F \vdash \Delta}$

(Here all the F's and F_i's are formulas in FORM.) The forms of inference rules corresponding to the few decomposition rules which do not conform to the above patterns can be either obtained by combining those given above, or derived directly from Lemmas 8.3 and 8.4.

The resulting inference rules of the sequent calculus (SC) are given in Table 1, alongside the original DRS rules they were derived from. Note that in addition to the rules derived from DRS, we have also included in the sequent calculus the universally accepted *thinning rules*, which allow us simplify some of the aforementioned derived rules, for in their presence we can avoid repeating certain formulas appearing in the premise(s) of the rule in its conclusion (this concerns especially the expansion rules).

Table 1.

Decomposition rules: DRS	Sequent calculus rules: SC
$\begin{array}{c} \Omega', \mathbf{T}(F), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in F), \Omega'' \end{array}$ $*$	$\begin{array}{c} \Gamma \vdash \Delta, x \in F \\ \hline \Gamma \vdash \Delta, F \end{array}$ $*$
$\begin{array}{c} \Omega', \mathbf{NT}(F), \Omega'' \\ \hline\hline \Omega', \mathbf{NT}(y \in F), \Omega'', \mathbf{NT}(F) \end{array}$ $**$	$\begin{array}{c} \Gamma, y \in F \vdash \Delta \\ \hline \Gamma, F \vdash \Delta \end{array}$ $**$
$\begin{array}{c} \Omega', \mathbf{T}(x \in \neg F), \Omega'' \\ \hline\hline \Omega', \mathbf{NT}(x \in F), \Omega'' \end{array}$	$\begin{array}{c} \Gamma, x \in F \vdash \Delta \\ \hline \Gamma \vdash \Delta, x \in \neg F \end{array}$
$\begin{array}{c} \Omega', \mathbf{NT}(x \in \neg F), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in F), \Omega'' \end{array}$	$\begin{array}{c} \Gamma \vdash \Delta, x \in F \\ \hline \Gamma, x \in \neg F \vdash \Delta \end{array}$
$\begin{array}{c} \Omega', \mathbf{T}(x \in F \wedge G), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in F), \Omega'' \mid \Omega', \mathbf{T}(x \in G), \Omega'' \end{array}$	$\begin{array}{c} \Gamma \vdash \Delta, x \in F \qquad \Gamma \vdash \Delta, x \in G \\ \hline \Gamma \vdash \Delta, x \in F \wedge G \end{array}$
$\begin{array}{c} \Omega', \mathbf{NT}(x \in F \wedge G), \Omega'' \\ \hline\hline \Omega', \mathbf{NT}(x \in F), \mathbf{NT}(x \in G), \Omega'' \end{array}$	$\begin{array}{c} \Gamma, x \in F, x \in G \vdash \Delta \\ \hline \Gamma, x \in F \wedge G \vdash \Delta \end{array}$
$\begin{array}{c} \Omega', \mathbf{T}(x \in F \vee G), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in F), \mathbf{T}(x \in G), \Omega'' \end{array}$	$\begin{array}{c} \Gamma \vdash \Delta, x \in F, x \in G \\ \hline \Gamma \vdash \Delta, x \in F \vee G \end{array}$
$\begin{array}{c} \Omega', \mathbf{NT}(x \in F \vee G), \Omega'' \\ \hline\hline \Omega', \mathbf{NT}(x \in F), \Omega'' \mid \Omega', \mathbf{NT}(x \in G), \Omega'' \end{array}$	$\begin{array}{c} \Gamma, x \in F \vdash \Delta \qquad \Gamma, x \in G \vdash \Delta \\ \hline \Gamma, x \in F \vee G \vdash \Delta \end{array}$
$\begin{array}{c} \Omega', \mathbf{T}(x \in \overline{sim}(A)F), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(y \in F), \Omega'', \mathbf{T}(x \in \overline{sim}(A)F) \mid \Omega', \mathbf{T}(x\ sim(A) y), \Omega'', \mathbf{T}(x \in \overline{sim}(A)F) \end{array}$	$\begin{array}{c} \Gamma \vdash \Delta, y \in F \qquad \Gamma \vdash \Delta, x\ sim(A)\ y \\ \hline \Gamma \vdash \Delta, x \in \overline{sim}(A)F \end{array}$
$\begin{array}{c} \Omega', \mathbf{NT}(x \in \overline{sim}(A)F), \Omega'' \\ \hline\hline \Omega', \mathbf{T}(x \in \underline{sim}(A)\neg F), \Omega'' \end{array}$	$\begin{array}{c} \Gamma \vdash \Delta, x \in \underline{sim}(A)\neg F \\ \hline \Gamma, x \in \overline{sim}(A)F \vdash \Delta \end{array}$
$\begin{array}{c} \Omega', \mathbf{T}(x \in \underline{sim}(A)F), \Omega'' \\ \hline\hline \Omega', \mathbf{NT}(x\ sim(A)\ z), \mathbf{T}(z \in F), \Omega'' \end{array}$ $***$	$\begin{array}{c} \Gamma, x\ sim(A)\ z \vdash \Delta, z \in F \\ \hline \Gamma \vdash \Delta, x \in \underline{sim}(A)F \end{array}$ $***$

Decomposition rules: DRS	Sequent calculus rules: SC
$\dfrac{\Omega', \mathbf{NT}(x \in \underline{sim}(A)F), \Omega''}{\Omega', \mathbf{T}(x \in \overline{sim}(A)\neg F), \Omega''}$	$\dfrac{\Gamma \vdash \Delta, x \in \overline{sim}(A)\neg F}{\Gamma, x \in \underline{sim}(A)F \vdash \Delta}$
$\dfrac{\Omega', \mathbf{T}(x\ sim(A \cup B)\ y), \Omega''}{\Omega', \mathbf{T}(x\ sim(A)\ y), \Omega'' \mid \Omega', \mathbf{T}(x\ sim(B)\ y), \Omega''}$	$\dfrac{\Gamma \vdash \Delta, x\ sim(A)\ y \qquad \Gamma \vdash \Delta, x\ sim(B)\ y}{\Gamma \vdash \Delta, x\ sim(A \cup B)\ y}$
$\dfrac{\Omega', \mathbf{NT}(x\ sim(A \cup B)\ y), \Omega''}{\Omega', \mathbf{NT}(x\ sim(A)\ y), \mathbf{NT}(x\ sim(B)\ y), \Omega''}$	$\dfrac{\Gamma, x\ sim(A)\ y, x\ sim(B)\ y \vdash \Delta}{\Gamma, x\ sim(A \cup B)\ y \vdash \Delta}$
$\dfrac{\Omega', \mathbf{NT}(x \in y), \Omega''}{\Omega', \Omega'', \mathbf{NT}(x \in y), \mathbf{NT}(y \in x)}$	$\dfrac{\Gamma, y \in x \vdash \Delta}{\Gamma, x \in y \vdash \Delta}$
$\dfrac{\Omega', \mathbf{NT}(x \in y), \Omega'', \mathbf{NT}(y \in F), \Omega'''}{\Omega', \Omega'', \Omega''', \mathbf{NT}(x \in y), \mathbf{NT}(y \in F), \mathbf{NT}(x \in F)}$	$\dfrac{\Gamma, x \in F \vdash \Delta}{\Gamma, x \in y, y \in F \vdash \Delta}$ $\bullet$
$\dfrac{\Omega', \mathbf{T}(x\ sim(C)\ y), \Omega''}{\Omega', \mathbf{T}(y\ sim(C)\ x), \Omega''}$	$\dfrac{\Gamma \vdash \Delta, y\ sim(C)\ x}{\Gamma \vdash \Delta, x\ sim(C)\ y}$
$\dfrac{\Omega', \mathbf{NT}(x\ sim(C)\ y), \Omega''}{\Omega', \mathbf{NT}(y\ sim(C)\ x), \Omega''}$	$\dfrac{\Gamma, y\ sim(C)\ x \vdash \Delta}{\Gamma, x\ sim(C)\ y \vdash \Delta}$
$\dfrac{\Omega', \mathbf{NT}(x \in y), \Omega'', \mathbf{T}(y\ sim(C)\ z), \Omega'''}{\Omega', \Omega'', \Omega''', \mathbf{NT}(x \in y), \mathbf{T}(y\ sim(C)\ z), \mathbf{T}(x\ sim(C)\ z)}$	$\dfrac{\Gamma \vdash \Delta, x\ sim(C)\ z}{\Gamma, x \in y \vdash \Delta, y\ sim(C)\ z}$ $\bullet$

Additional rules for finite $\textsc{Prop}$: DRS	Additional rules for finite $\textsc{Prop}$: SC
$\dfrac{\Omega', \mathbf{T}(x\ sim(-\mathbf{P} \cap \mathbf{S})\ y), \Omega''}{\Omega', \mathbf{T}(x\ sim(\bigcup_{\mathbf{p}' \in \textsc{Conp}-\mathbf{P}}\ \mathbf{p}' \cap S)\ y), \Omega''}$	$\dfrac{\Gamma \vdash \Delta, x\ sim(\bigcup_{\mathbf{p}' \in \textsc{Conp}-\mathbf{P}}\ \mathbf{p}' \cap S)\ y}{\Gamma \vdash \Delta, x\ sim(-\mathbf{P} \cap S)\ y}$
$\dfrac{\Omega', \mathbf{NT}(x\ sim(-\mathbf{P} \cap S)\ y), \Omega''}{\Omega', \mathbf{NT}(x\ sim(\bigcup_{\mathbf{p}' \in \textsc{Conp}-\mathbf{P}}\ \mathbf{p}' \cap S)\ y), \Omega''}$	$\dfrac{\Gamma, x\ sim(\bigcup_{\mathbf{p}' \in \textsc{Conp}-\mathbf{P}}\ \mathbf{p}' \cap S)\ y \vdash \Delta}{\Gamma, x\ sim(-\mathbf{P} \cap S)\ y \vdash \Delta}$
	Thinning rules
	$\dfrac{\Gamma \vdash \Delta}{\Gamma, F \vdash \Delta}, \quad \dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, F}$

In all rules given in Table 1, $x, y, z \in \textsc{Vare}, F, G \in \textsc{Form}$, the Ω's are sequences of signed formulas, Γ, Δ are finite sets of formulas in $\textsc{Form}$, $A, B \in \textsc{Term}$, $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \subseteq \textsc{Conp}$, $-\mathbf{P} = -\mathbf{p}_1 \cap \cdots \cap -\mathbf{p}_n$, C is a component, and S — a subcomponent. Naturally enough, the components and subcomponents for a sequent Γ, Δ are defined as components and subcomponents for the sequence of signed formulas $\mathbf{NT}(\Gamma), \mathbf{T}(\Delta)$ 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 $z \in \textsc{Vare}$ and any $G \in FORM$, x is a a new variable in $\textsc{Vare}$, 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 $z \in \textsc{Vare}$ and any $G \in FORM$, y is an arbitrary variable in $\textsc{Vare}$

*** 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 $\textsc{Vare}$

Of course, conditions defining fundamental sequences of signed formulas give rise to axioms of the sequent calculus. That is, we have

LEMMA 8.5. *If any sequence of signed formulas containing a sequence Ω is fundamental, then $\Omega^- \vdash \Omega^+$ is an axiom of the sequence calculus, i.e. an universally valid sequent.*

The axioms of sequent calculus corresponding to the conditions defining fundamental sequences of signed formulas are given in Table 2, again alongside the original fundamental subsequences. Note that if for the sequence Ω considered in the above Lemma we have $\Omega^+ = \emptyset$, then as an axiom corresponding to Ω we take $\Omega^- \vdash \Delta$, where Δ is arbitrary sequence of formulas, because in this case every sequence containing $\Omega, \mathbf{T}(\Delta)$ is also fundamental.

Table 2

Fundamental subsequence: DRS	Sequent axiom: SC
$\mathbf{T}(F), \mathbf{NT}(F)$	$F \vdash F$
$\mathbf{NT}(F), \mathbf{NT}(\neg F)$	$F, \neg F \vdash \Delta$
$\mathbf{T}(x \in x)$	$\vdash x \in x$
$\mathbf{T}(x\ sim(\mathbf{0})\ y)$	$\vdash x\ sim(\mathbf{0})\ y$
$\mathbf{T}(x\ sim(A)\ x)$	$\vdash x\ sim(A)\ x$
$\mathbf{T}(x\ sim(\mathbf{p}_j \cap \mathbf{S})\ y)$, $\mathbf{T}(x'\ sim(\mathbf{p}_j \cap \mathbf{S}')\ y')$, where $\mathbf{S} \neq \mathbf{S}'$ are subcomponents	$\vdash x\ sim(\mathbf{p}_j \cap \mathbf{S})\ y, x'\ sim(\mathbf{p}_j \cap \mathbf{S}')\ y'$, where $\mathbf{S} \neq \mathbf{S}'$ are subcomponents

The notational conventions in Table 2 are the same as in Table 1.

Before defining the notion of provability in our sequent calculus, let us recall that all the decomposition rules in DRS were tailored to sequences of signed formulas 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 are in fact tailored to sequents in normal form (defined analogously as the normal form of sequences of formulas); hence, just like in the DRS system, the deduction mechanism of SC should be applied to the normal form of a sequent instead of the original sequent.

To be precise, by the *a normal form* of a sequent $\mathbf{S}$ of the form $\Gamma \vdash \Delta$ we shall mean 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 occuring in these sets by its normal form $N(A)$ (defined with respect to the sequence $\mathbf{NT}(\Gamma), \mathbf{T}(\Delta)$ in the way described in Section 6).

Thus 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 2 by a finite number of applications of the inference rules given in Table 1. Again, such a definition is justified by th 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 (because the interpretion of A coincides with the interpretation of $N(A)$ in every model).

Obviously, all the SC rules are sound, and all the axioms — valid; this follows from Lemmas 7.2-8.5. Hence the sequent calculus we have developed is sound, i.e.

THEOREM 8.6. *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. Hence $N(\mathbf{S})$ is valid – and so must be $\mathbf{S}$, because these sequents are semantically equivalent.

It is quite easy to prove that the sequent calculus SC we have developed is also complete. ■

Namely we have the following:

THEOREM 8.7. *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 $\mathbf{NT}(\Gamma), \mathbf{T}(\Delta)$ is also valid. By the completeness theorem for the DRS system this means that the above sequence has a finite decomposition tree DT with only fundamental sequences at its leaves. Basing on that tree, we can easily construct a proof of the normal form of the original sequent in the SC calculus. Namely, we prove by induction that the sequent $\Omega^- \vdash \Omega^+$ corresponding to any sequence Ω of signed fomulas labeling a vertex of DT is provable.

We start from the leaves: they are labeled by fundamental sequences, so the corresponding sequents can be derived from the axioms of the sequent calculus by means of the thinning rules. Now we can go upwards in DT, replacing each downward application of a decomposition rule by an upward application of the corresponding sequent calculus rule.

For simplicity, denote by $seq.\Omega$ the sequent $\Omega^- \vdash \Omega+$ corresponding to a sequence Ω of signed formulas, and by $l(v)$ — the label of a vertex v of DT. Consider any vertex v on DT, and assume that, for all the vertices v' below v, $seq.l(v')$ is provable in SC.

Assume first that v has a single son v'. Then $l(v')$ has been obtained from $l(v)$ by applying a single-premise rule R of the form

$$\frac{\Omega', \Pi, \Omega''}{\overline{\Omega', \Sigma, \Omega''}},$$

where $l(v) = \Omega', \Pi, \Omega'', l(v') = \Omega', \Sigma, \Omega''$. Moreover, $seq.l(v')$, which is just $seq.(\Omega', \Sigma, \Omega'') \equiv \Omega'^-, \Omega''^-, \Sigma^- \vdash \Omega'^+, \Omega''^+, \Sigma^+$, is provable by inductive assumption. But $seq.l(v) \equiv \Omega'^-, \Omega''^-, \Pi^- \vdash \Omega'^+, \Omega''^+, \Pi^+$ can be obtained from $seq.l(v')$ by the SC rule

$$\frac{\Gamma, \Sigma^- \vdash \Delta, \Sigma^+}{\Gamma, \Pi^- \vdash \Delta, \Pi^+}$$

corresponding to the DRS rule R — more precisely, its instance with $\Gamma = \Omega'^-, \Omega''^-$, $\Delta = \Omega'^+, \Omega''^+$. Hence $seq.l(v)$ is also provable.

The case when v has two sons – v' and v'' – is quite similar; in this case $seq.l(v)$ is obtained from $seq.l(v')$ and $seq.l(v'')$ by a double-premise SC rule corresponding to the double-premise DRS rule used for obtainig $l(v')$ and $l(v'')$ out of $l(v)$, which again implies that $seq.l(v)$ is provable in SC.

We finish our induction at the root, which is labeled by the sequence $N(\mathbf{NT}(\Gamma), \mathbf{T}(\Delta))$ (recall that the root of $DT(\Omega)$ is labeled by $N(\Omega)$ — the normal form of Ω). Thus $seq.N(\mathbf{NT}(\Gamma), \mathbf{T}(\Delta))$ can be derived from the axioms of SC by means of the inference rules. However, as it is easy to see that $seq.N(\mathbf{NT}(\Gamma), \mathbf{T}(\Delta))$ is just $N(\Gamma \vdash \Delta)$, i.e. the normal form of the original sequent $\Gamma \vdash \Delta$. By the definition of provability, this means that $\Gamma \vdash \Delta$ is provable in SC. ■

As the syntactic entailment $\vdash$ in a sequent corresponds to the semantic consequence relation, Theorem 8.7 means that the deduction system we have developed in this section is complete for theories, i.e. we have the following

THEOREM 8.8. *If $Ax = \{F_1, F_2, \dots, F_n\}$ is any finite set of formulas in FORM, then, for any $F \in$ FORM, $Ax \models F$ (i.e. F is true in every model of Ax) iff the sequent $Ax \vdash F$ is provable in SC.*

9. Conclusions

The language presented here, together with its complete Gentzen calculus system, can be used for reasoning about any family of similarity relations defined by subsets of a given set of attributes. The logic supports the operations of lower and upper approximation with respect to any similarity relations; they are represented by modalities parametrised by sets of properties. Since such approximations are of interest in many AI problems, this gives a wide range of possible applications of the system.

The two main technical tools which allowed us to get the completeness result for what is in fact a multi-modal logic based on a whole family of accessibility relations were: the mechanism of the DRS deduction system –

involving an intricate use of variables to model both the membership relation and modalities — and representation of the set P being the modality parameter as a union of components. Due to the latter, we could replace formulae with arbitrary modalities of the form $\underline{sim}(P), \overline{sim}(P)$ by formulae with modalities $\underline{sim}(C), \overline{sim}(C)$ parametrised by atomic, disjoint sets. This together with the special DRS rules for modalities allowed us to reduce reasoning about formulae with arbitrary modalities to reasoning about simple formulae of the form $\mathbf{T}(x\, sim(C)\, y), \mathbf{NT}(x\, sim(C)\, y)$, where C is a component and x, y are individual variables. Clearly, this allowed us to avoid the usual problems arising in case of considering multiple accessibility relations, like the fact that the inclusion $R_1 \cap R_2 \subset R$ is not expressible by any modal formula: as the components are disjoint, the latter problem does not arise at all. It should be noted that the DRS modality rules could not be expressed in the form of Hilbert-style axioms within propositional modal logic, because in the DRS rules the use of variables to model the 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 system presented here can be easily generalized to the case of indiscernibility relations, which are just transitive similarity relations. Namely, if we denote by $ind(P)$ indiscernibility with respect to the set of properties $P \subseteq \textsc{Prop}$, then the decomposition rules for $ind(P)$ are just those for $sim(P)$ plus the following transitivity rule:

$$\textbf{(tran)} \qquad \frac{\mathbf{NT}(x\, ind(C)\, y), \mathbf{NT}(y\, ind(C)\, z)}{\mathbf{NT}(x\, ind(C)\, y), \mathbf{NT}(y\, ind(C)\, z), \mathbf{NT}(x\, ind(C)\, z)}$$

(where C is any component, and $x, y, z \in VARE$). In the sequent calculus, its counterpart is the rule

$$\frac{\Gamma, x\, ind(C)\, z \vdash \Delta}{\Gamma, x\, ind(C)\, y, y\, ind(C)\, z \vdash \Delta}$$

It is easy to see that the completeness proofs of the DRS system given in the paper can be generalized to the case of indiscernibility rules. Thanks to the above transitivity rule (tran), the relations obtained in the counter-model construction will be indiscernibility relations, so we will get a correct counter-model. Of course, the transition to the Gentzen calculus and the completeness proof for the latter system carry over without any changes.

Finally. let us note that the logical language given here can be easily extended to a first-order language; however, if we want to preserve the completeness of the deduction system, we have to limit quantification to the individual variables in VARE only (quantfication over the set variables in VARSE and VARSP would take us into second-order logic).

A first-order language would give e.g. the possibility of expressing the axioms of a theory as closed formula. On the other hand, the development of such a language would require some careful decisions as to the choice of primitives for representing the various kinds of relationships we have to consider in our language (membership relation, similarity relations, inclusion of sets), as well as expressing the approximation operations. One obvious choice would be e.g. to treat formulae of the form $x\ sim(P)\ y$, expressing the similarity relation itself, as a primary concept, and introduce the operations of lower and upper approximation as derived constructs. Another primary concept could be the membership relation $x \in E$; having this, we could introduce inclusion of sets and all set-theoretical operations as derived concepts. The appropiate language will be presented in a separate paper.

Yet another expansion to be investigated is considering the case when there are several similarity relations corresponding to a given individual property – perhaps forming a certain hierarchy of refinements. For example, if we take the property H representing 'height', then we can consider several 'threshold' similarity relations $sim_a(H)$ connected with height, where $sim_a(x, x')$ iff $|height(x) - height(x') \leq a|$. Such an approach, besides having obvious practical applications, can lead to certain connections with uniform topologies.

References

[1] BETH, E. W., 1959, *The Foundations of Mathematics,* North-Holland.

[2] FITTING, M., 1983, *Proof Methods for Modal and Intuitionistic Logics,* D. Reidel Publishing Company, Dordrecht.

[3] KOLETSOS, G., 1978, *Sequent calculus and partial logic,* M.Sc. Thesis, University of Manchester.

[4] KONIKOWSKA, B., 'A logic for reasoning about similarity', to appear in: E. Orlowska, ed., *Reasoning with incomplete information: The rough set approach.*

[5] KONIKOWSKA, B., A. TARLECKI, A. BLIKLE, 1991, 'A three-valued logic for software specification and validation', *Fundamenta Informaticae* **14,** 411–453.

[6] LYNDON, C., 1964, *Notes on Logic,* D. Van Nostrand, Princeton.

[7] ORLOWSKA, E., 1985, 'Logic of indiscernibility relations', *Bull. Pol. Acad. Sci.* **33,** 467-485.

[8] PAWLAK, Z., 1982, 'Rough sets', *Intl. Journ. of Comp. and Inf. Sci.* **11,** 341–356.

[9] RASIOWA, H., R. SIKORSKI, 1963, *The mathematics of metamathematics,* Warsaw, PWN.

[10] Vakarelov, D., 1991, 'Modal logics for similarity relations in Pawlak knowledge representation systems', *Fundamenta Informaticae* **15,** 61–79.

[11] Vakarelov, D., 1991, 'Logical analysis of positive and negative similarity relations in property systems', Proceedings of First World Conference on the Fundamentals of AI, Paris.

Institute of Computer Science
Polish Academy of Sciences
Ordona 21, 01–237 Warsaw,
POLAND
e-mail: beatak@ipipan.waw.pl