

A binary modality for reasoning about conjoined situations in a hybrid logic

David Ahn and Lenhart Schubert

*Computer Science Department
University of Rochester
(davidahn) | (schubert)@cs.rochester.edu*

Abstract

In this paper, we present a hybrid logic with a sound and complete tableau system for reasoning about situations. We identify our satisfaction relation between states and formulas with Schubert's characterization relation between situations and formulas [6] and introduce a binary modality for reasoning about conjoined situations.

1 Describing eventualities

This paper presents a hybrid logic for reasoning about the relation between sentences and those entities of which we take sentences to hold. We adopt, in general outline, the theory of situations and their relation to sentences presented by [6,7], which is motivated by reference to these entities in natural language. This theory provides two relations between arbitrary formulas and situations: characterization, a tight, Davidsonian-like coupling; and support, a looser, Situation-Semantics-like relation. In our hybrid logic, we identify satisfaction with characterization and provide support as a derived relation.

We take as primitive a relation between atoms of our language and situations that is exactly the relation between event predicates and events in a Davidsonian theory [5]. Thus, an atom and a situation stand in this characterization relation just in case the atom is a description of the situation as a whole. On this view, situations are identified with events, and atoms, with Davidsonian event predicates. Situations are entities which are classified by atoms. As an example, the atom *sneeze* (with the obvious intended interpretation) would characterize an event of John sneezing but not an episode of John having a cold, even if that episode includes many events of John sneezing.

There are several ways to extend this characterization relation to arbitrary formulas. It may be the case that a given situation is characterized independently by each of several atoms. In that case, we would like to extend the characterization relation so that the conjunction of these atoms characterizes

the situation. For example, given a situation of John running, each of the atoms *run* and *move*, on their own, characterizes this situation, since every situation of John running is also a situation of John moving. Thus, we would like the conjunction of *run* and *move* to characterize this situation.

Schubert argues for a related way to extend the characterization relation. It may be the case that for a given situation, there are multiple formulas, none of which independently characterizes the situation, but which together manage to describe it as a whole. For example, a situation of John sneezing (once) and coughing (once) in direct succession is not characterized by either atomic formula *sneeze* or *cough* alone, but is in some sense, a conjoined situation of a sneezing subsituation and a coughing subsituation. We would like some way to represent the relation between these two formulas, on the one hand, and the situation, on the other, as an extension of the characterization relation.

Another way in which we would like to extend the characterization relation is to allow a negated formula to characterize a situation. Schubert argues that negative sentences in natural language can introduce situations which are referred to by demonstrative pronouns and which stand in causal relations with other situations. These negative situations are not simply the situations of which the argument of the negation fails to hold, but rather situations of which the negation itself holds. What constitutes such a situation is a question to which we do not have a definitive answer (see [6] and [4] for discussion). In general, though, we allow our characterization relation to be partial, so that there are situations and formulas for which neither the formula nor its negation characterize the situation.

Seligman [8] presents a hybrid logic (with predication and quantification) in which the nominals refer to situations and the satisfaction relation resembles in many ways the characterization relation we have described here. His logic does not, however, allow extend to the second case described above, in which multiple formulas together characterize a conjoined situation which none of them characterize independently.

We present in this paper a propositional hybrid logic, HLC^{**} , in which the satisfaction relation is identified with the characterization relation described here. To handle of conjoined situations, we introduce a binary modal operator which relates its arguments to a situation consisting of the subsituations characterized by the arguments. An HLC^{**} model is a join semi-lattice of situations based on a partial inclusion ordering. Unlike other situation logics with an inclusion ordering, our basic relation between formulas and situations is not persistent with respect to this ordering. It is possible, however, to define in HLC^{**} a persistent relation along the lines of the support relation of Schubert or Barwise and Perry [2]. HLC^{**} can be thought of as a modal reconstruction of the propositional fragment of Schubert's FOL^{**} , although unlike Schubert, we allow for situations to be characterized by both boolean and modal conjunction, thus distinguishing the first two kinds of extension to primitive characterization discussed above.

2 Syntax and semantics

2.1 Syntax

An HLC** language is determined by a set of propositional symbols $\text{PROP} = \{p, q, p', q', \dots\}$ and a set of nominals $\text{NOM} = \{i, j, k, \ell, \ell_1, \dots, \ell_i, \dots\}$. Members of $\text{PROP} \cup \text{NOM}$ are called atoms. The set of well-formed formulas \mathcal{WFF} of an HLC** language defined over PROP and NOM is given by the following grammar, where $p \in \text{PROP}$ and $i \in \text{NOM}$:

$$S ::= p \mid i \mid S**i \mid \neg S \mid \diamond S \mid \square S \mid S \circ S \mid S?S \mid S \wedge S \mid S \vee S$$

2.2 Semantics

Models for HLC** are like models for other modal logics. There are two primary differences. The first is that states have an intended interpretation as situations, and thus, following [6], the set of states is a join semi-lattice based on a partial inclusion ordering, \sqsubseteq . We add a ternary relation on this set, \sqcup , which relates pairs of situations to their suprema. The second difference is that interpretation is performed by a pair of functions, one that assigns to each atom the situations that are characterized by the atom, and one that assigns the situations that are characterized by the negation of the atom.

Definition 2.1 Models, standard models, and frames

A *model* \mathcal{M} for an HLC** language is a pair $\langle \langle \mathcal{S}, \sqsubseteq, \sqcup \rangle, \langle \mathcal{I}^+, \mathcal{I}^- \rangle \rangle$ s.t.:

- (i) $\langle \mathcal{S}, \sqsubseteq \rangle$ is a join semi-lattice, and \sqcup , a ternary relation on \mathcal{S} such that $\forall s, t, u \in \mathcal{S}. \sqcup(s, t, u)$ iff $s = \sup\{t, u\}$. $\langle \mathcal{S}, \sqsubseteq, \sqcup \rangle$ is called the *frame* underlying \mathcal{M} .
- (ii) \mathcal{I}^+ and \mathcal{I}^- are interpretation functions with domain $\text{PROP} \cup \text{NOM}$ and range $\text{Pow}(\mathcal{S})$. We call \mathcal{I}^+ the positive interpretation function and \mathcal{I}^- the negative interpretation, or anti-interpretation, function.

\mathcal{I}^+ (or \mathcal{I}^-) is a *standard* (anti-)interpretation iff for all $i \in \text{NOM}$, $\mathcal{I}^+(i)$ (or $\mathcal{I}^-(i)$) is a singleton subset of \mathcal{S} . A model \mathcal{M} is a *standard model* iff:

- (i) both \mathcal{I}^+ and \mathcal{I}^- are standard,
- (ii) for all $i \in \text{NOM}$, $\mathcal{I}^+(i) = \mathcal{I}^-(i)$,
- (iii) \mathcal{I}^+ and \mathcal{I}^- do not collide on propositional variables: for all $p \in \text{PROP}$, $\mathcal{I}^+(p) \cap \mathcal{I}^-(p) = \emptyset$.

The semantic definition of an HLC** language is given in terms of satisfaction conditions. The satisfaction relation \models is mirrored in the language by the characterization relation $*$.

Definition 2.2 Satisfaction conditions for HLC**

Let \mathcal{M} be a standard model $\langle \langle \mathcal{S}, \sqsubseteq, \sqcup \rangle, \langle \mathcal{I}^+, \mathcal{I}^- \rangle \rangle$ and $s \in \mathcal{S}$. We give both

positive satisfaction \Vdash and anti-satisfaction \Vdash^- conditions:

$$\begin{aligned}
\mathcal{M}, s \Vdash p & \quad \text{iff } s \in \mathcal{I}^+(p) \\
\mathcal{M}, s \Vdash i & \quad \text{iff } \mathcal{I}^+(i) = \{s\} \\
\mathcal{M}, s \Vdash \neg\phi & \quad \text{iff } \mathcal{M}, s \Vdash^- \phi \\
\mathcal{M}, s \Vdash \diamond\phi & \quad \text{iff } \exists s'. s' \sqsubseteq s \text{ and } \mathcal{M}, s' \Vdash \phi \\
\mathcal{M}, s \Vdash \square\phi & \quad \text{iff } \forall s'. \text{ if } s' \sqsubseteq s, \text{ then } \mathcal{M}, s' \Vdash \phi \\
\mathcal{M}, s \Vdash \phi \circ \psi & \quad \text{iff } \exists t, u. \sqcup(s, t, u) \text{ and } \mathcal{M}, t \Vdash \phi \text{ and } \mathcal{M}, u \Vdash \psi \\
\mathcal{M}, s \Vdash \phi? \psi & \quad \text{iff } \forall t, u. \text{ if } \sqcup(s, t, u), \text{ then } (\mathcal{M}, t \Vdash \phi \text{ or } \mathcal{M}, u \Vdash \psi) \\
\mathcal{M}, s \Vdash \phi \wedge \psi & \quad \text{iff } \mathcal{M}, s \Vdash \phi \text{ and } \mathcal{M}, s \Vdash \psi \\
\mathcal{M}, s \Vdash \phi \vee \psi & \quad \text{iff } \mathcal{M}, s \Vdash \phi \text{ or } \mathcal{M}, s \Vdash \psi \\
\mathcal{M}, s \Vdash \phi**i & \quad \text{iff } \mathcal{M}, t \Vdash \phi, \text{ where } \mathcal{I}^+(i) = \{t\}
\end{aligned}$$

In the anti-satisfaction (\Vdash^-) conditions, the atomic conditions and the condition for the characterization operator are different. Otherwise, the conditions for \square and \diamond ; \circ and $?$; and \wedge and \vee are exchanged, with \Vdash^- exchanged everywhere for \Vdash .

$$\begin{aligned}
\mathcal{M}, s \Vdash^- p & \quad \text{iff } s \in \mathcal{I}^-(p) \\
\mathcal{M}, s \Vdash^- i & \quad \text{iff } s \notin \mathcal{I}^-(i) \\
\mathcal{M}, s \Vdash^- \phi**i & \quad \text{iff } \mathcal{M}, t \not\Vdash \phi, \text{ where } \mathcal{I}^-(i) = \{t\}
\end{aligned}$$

Definition 2.3 Satisfiability and validity

If for some standard model \mathcal{M} and some state s , $\mathcal{M}, s \Vdash \phi$, ϕ is *satisfied* in \mathcal{M} at s . Satisfiability and unsatisfiability are defined in the obvious way. If ϕ is satisfied at all states in a standard model \mathcal{M} , ϕ is *globally satisfied* in \mathcal{M} ($\mathcal{M} \Vdash \phi$). If for a frame \mathcal{F} and for every standard model \mathcal{M} based on \mathcal{F} , $\mathcal{M} \Vdash \phi$, ϕ is *valid on \mathcal{F}* ($\mathcal{F} \Vdash \phi$). If ϕ is valid on all frames, ϕ is *valid* ($\Vdash \phi$).

Note that we do not constrain our interpretation functions to guarantee persistence with respect to the inclusion ordering. We can define a relation between sentences and nominals that is persistent with respect to the inclusion ordering: $\phi * i$ (read ϕ supports i) iff $(\diamond\phi)**i$.

We briefly note some of the algebraic properties of the connectives of our logic. First, \diamond and \square ; \circ and $?$; and \wedge and \vee are duals with respect to negation and thus satisfy DeMorgan's laws. Also, double negation can be eliminated. We will therefore assume for the remainder of this paper that negation has been driven in and double negation has been eliminated where possible. For example, we will take the tableau rule for \vee to apply directly to formulas of the form $(\neg(\phi \wedge \psi))**i$, resulting in two outputs, $(\neg\phi)**i$ and $(\neg\psi)**i$.

Because both the satisfaction and the anti-satisfaction conditions for characterization are defined in terms of positive satisfaction, however, negation cannot be driven across characterization.

The boolean connectives, \wedge and \vee , are commutative, associative and idempotent. The modal connectives, \circ and $?$, are commutative and associative, like the underlying join operation on which they are based. $?$ is also idempotent, but \circ is not fully idempotent. If a state satisfies ϕ , then it also satisfies $\phi \circ \phi$, but a state may satisfy $\phi \circ \phi$ and yet not satisfy ϕ .

3 Proof theory

For our proof theory for HLC^{**} , we present an unsigned tableau system [9] that is based largely on Blackburn's system for hybrid logic [3], although there are, of course, more rules to handle the additional binary modal operators.

Normally, given a sound and complete tableau system, we would prove ϕ by contradiction. In HLC^{**} , however, the unsatisfiability of $\neg\phi$ does not guarantee the validity of ϕ . Because there are separate interpretation functions and satisfaction conditions for negative and positive statements, it is possible to have a single state which satisfies neither ϕ nor $\neg\phi$. What the validity of ϕ in an HLC^{**} language does guarantee is that for any choice of i , there is no state that satisfies $\neg(\phi**i)$. In fact, the following lemma holds (we omit proofs of this and other lemmas for brevity; see [1] for full proofs):

Lemma 3.1 *An HLC^{**} formula ϕ is valid if and only if for every nominal i , $\neg(\phi**i)$ is unsatisfiable.*

In order to prove ϕ we build a tableau starting with $\neg(\phi**i)$, where i is a nominal that does not occur in ϕ . If the tableau closes, then ϕ is valid; otherwise, it is not.

Since our tableau proofs always begin with a statement of the form $\neg(\phi**i)$, it turns out that it is sufficient for completeness for our tableau rules to deal only with formulas of the form $\phi**i$ or $\neg(\phi**i)$, which we call characterization statements. As Blackburn [3] points out, a tableau system in which all the formulas are characterization statements essentially internalizes the labels of a labelled deduction system.

3.1 Tableau rules

The tableau rules are given in Figures (1)–(7) at the end of this paper. In these rules, ϕ and ψ represent arbitrary formulas, i, i', j, j', k, k' represent arbitrary nominals, and ℓ, ℓ_1, ℓ_2 must be new nominals that do not occur in any formula on the branch to which the outputs are being added. The intended meaning of these rules should be clear: the formulas above the line are the inputs, and the formulas below the line are outputs. Branching rules have vertical lines which separate the outputs into alternative sets of outputs.

Figures (1)–(4) give rules for simplifying embedded formulas in characterization statements. Recall that we assume that negation is driven in across all connectives except the characterization operator, across which negation cannot be driven in. Thus, we need two rules for each of the unary modalities, binary modalities, and booleans. Note that rule $[\circ]$ in Figure (3) does not apply when both ϕ and ψ are nominals ($[\text{Join1}]$ and $[\text{Join2}]$ apply instead).

Rules to regulate the frame are given in Figures (5)–(7). Note that the rule $[\text{Lattice}]$ in Figure (7) yields an undecidable system. We could generalize our models from join semi-lattices to posets and constrain the ternary relation \sqcup so that if $\sqcup(i, j, k)$, then $i = \sup\{j, k\}$. Then, the $[\text{Lattice}]$ rule would be unnecessary for completeness with respect to the intended models.

3.2 Tableau construction

We now present a tableau construction algorithm (again based on [3]) for building a tableau from a (possibly infinite) set of formulas. This algorithm is sufficient to define derivability, provability, and consistency.

Definition 3.2 Systematic construction

Let PROP and NOM be countable sets. Let PAR be countable set that is disjoint from each of PROP and NOM. Elements of PAR will be used as new nominals, as needed. Assume that each of these sets is enumerated. Let Φ be a set of formulas in an HLC** language over PROP and NOM. Let i be the first nominal in the enumeration of PAR. Let $\Phi^i = \{\phi ** i \mid \phi \in \Phi\}$. Enumerate Φ^i following the enumeration of Φ . Construction proceeds as follows:

Step 1: Let the first characterization statement in Φ^i be the root node of a tree. This tree, which we call T_1 , is clearly finite.

Step $n + 1$: Let T_n be the finite tree constructed at step n . Apply all rules that are applicable to formulas or tuples of formulas in T_n . Since T_n has only finitely many nodes, there are only finitely many possible rule applications. The following constraints must be obeyed in applying the rules:

- (i) Apply $[\text{**Ref}]$, $[\diamond\text{Ref}]$, and $[\text{Lattice}]$ once for each distinct nominal (or pair of nominals) on each branch of T_n .
- (ii) When a non-branching rule is applied, the output formulas are added to the end of each branch containing the input formula(s).
- (iii) When a branching rule is applied, each branch containing the input formula(s) is split into as many branches as the rule has output alternatives. Each set of output alternatives is added to its own branch.
- (iv) When a rule that introduces a new nominal or nominals is applied, the next unused nominal(s) in PAR are used.

After all the rules have been applied, add the $n + 1$ -th element of Φ^i to the end of every branch. The resulting tree T_{n+1} is clearly finite.

T , the tree obtained as the limit of this sequence, is a *completed tableau*.

For any branch B of a tableau, B is *closed* iff it contains either:

- both $\neg(\phi**i)$ and $\phi**i$, for arbitrary formula ϕ and arbitrary nominal i ; or
- both $(\neg\alpha)**i$ and $\alpha**i$, for arbitrary atom α and arbitrary nominal i .

A branch B is *open* iff it is not closed. A tableau is closed iff all of its branches are closed. A completed tableau is open iff at least one of its branches is open.

Definition 3.3 Derivability and provability

A contradiction is *derivable* from a set of formulas Φ iff the application of the systematic tableau construction to Φ yields a closed tableau. A formula ϕ is *provable* iff there is a finite closed tableau whose root node is $\neg(\phi**i)$, for i a nominal that does not occur in ϕ .

Definition 3.4 Consistency

A formula ϕ is consistent iff for some nominal i that does not occur in ϕ , $\neg(\phi**i)$ is not provable. A finite set of formulas $\Phi = \{\phi_1, \dots, \phi_n\}$ is consistent iff for some nominal i that does not occur in any formula in Φ , $\neg((\phi_1**i) \wedge \dots \wedge (\phi_n**i))$ is not provable. A set of formulas Φ is consistent iff every finite subset of Φ is consistent.

4 Soundness and completeness

4.1 Soundness

We first define an appropriate notion of satisfiability for a tableau branch.

Definition 4.1 Satisfiability by label

Let Σ be a set of characterization statements and $\mathcal{M} = \langle \langle \mathcal{S}, \sqsubseteq, \sqcup \rangle, \langle \mathcal{I}^+, \mathcal{I}^- \rangle \rangle$ be a standard model. Then, Σ is *satisfied by label* in \mathcal{M} iff for all $\sigma \in \Sigma$:

- (i) if σ is of the form $\phi**i$, then $\mathcal{M}, s \Vdash \phi$, where $\mathcal{I}^+(i) = \{s\}$,
- (ii) if σ is of the form $\neg(\phi**i)$, then $\mathcal{M}, s \not\Vdash \phi$, where $\mathcal{I}^+(i) = \{s\}$.

We say that Σ is *satisfiable by label* iff there is a standard model in which Σ is satisfiable.

Theorem 4.2 (Soundness) *Let Σ be a set of characterization statements that is satisfiable by label. Then, for each rule R , at least one of the sets obtainable by expanding Σ by R is satisfiable by label.*

4.2 Completeness

Our proof of completeness follows Blackburn's, with complications due to the binary modalities and to the construction of more complicated models.

Definition 4.3 Hintikka set

A Hintikka set H is a set of characterization statements that satisfy:

- (i) Satisfiability conditions: these conditions ensure satisfiability at the atomic level in a standard model (α is an atom; i and j are nominals).

- (a) If $\alpha**i \in H$, then $\neg(\alpha**i) \notin H$.
 - (b) If $(\neg\alpha)**i \in H$, then $\neg((\neg\alpha)**i) \notin H$.
 - (c) If $\alpha**i \in H$, then $(\neg\alpha)**i \notin H$.
 - (d) If $(\diamond j)**i \in H$, then $\neg((\diamond j)**i) \notin H$.
- (ii) Closure conditions: H must be closed under application of the tableau rules. For branching rules, for every input formula (or tuple of input formulas) in H , only one output alternative set need be in H . For existential rules, the new nominals need not actually be new to H .

Definition 4.4 Let H be a Hintikka set. Define $\text{Nom}(H)$ to be the set:

$$\{ |i| \mid i \text{ is a nominal that occurs in some formula in } H \}.$$

Define the binary relation \sim_H on $\text{Nom}(H)$ by $i \sim_H j$ iff $i**j \in H$. \sim_H is obviously an equivalence relation. Define $|i|$ as the equivalence class of i under \sim_H .

Lemma 4.5 Let H be Hintikka set, and let $i, j \in \text{Nom}(H)$. Then, $i \sim_H j$ iff for every formula ϕ , $\phi**i \in H$ iff $\phi**j \in H$.

Definition 4.6 Induced model

Let H be a Hintikka set. Define a *standard model induced by H* as $\mathcal{H} = \langle \langle \mathcal{S}^H, \sqsubseteq^H, \sqcup^H \rangle, \langle \mathcal{I}^{H+}, \mathcal{I}^{H-} \rangle \rangle$, where:

- (i) $\mathcal{S}^H = \{ |i| \mid i \in \text{Nom}(H) \}$,
- (ii) $|i| \sqsubseteq^H |j|$ iff $(\diamond i)**j \in H$,
- (iii) $\sqcup^H(|i|, |j|, |k|)$ iff $(j \circ k)**i \in H$,
- (iv) For any propositional atom p that occurs in H , $\mathcal{I}^{H+}(p) = \{ |i| \mid p**i \in H \}$ and $\mathcal{I}^{H-}(p) = \{ |i| \mid (\neg p)**i \in H \}$. For any other propositional atom q , $\mathcal{I}^{H+}(q) = \mathcal{I}^{H-}(q) = \emptyset$. For any nominal $j \in \text{Nom}(H)$, $\mathcal{I}^{H+}(j) = \mathcal{I}^{H-}(j) = \{ |i| \mid j**i \in H \}$. For other nominals k , $\mathcal{I}^{H+}(k) = \mathcal{I}^{H-}(k) = \{ \emptyset \}$.

Lemma 4.7 Let H be a Hintikka set. Then any standard model induced by H , $\mathcal{H} = \langle \langle \mathcal{S}^H, \sqsubseteq^H, \sqcup^H \rangle, \langle \mathcal{I}^{H+}, \mathcal{I}^{H-} \rangle \rangle$, is, in fact, a standard model.

Lemma 4.8 (Hintikka's Lemma) Every Hintikka set is satisfiable by label (in a standard model induced by the set).

Lemma 4.9 Every open branch on a completed tableau is a Hintikka set.

As a consequence of (4.8) and (4.9), the completed tableau constructed from an unsatisfiable set of formulas must be closed.

Theorem 4.10 (Completeness) Every consistent set Φ is satisfiable.

Proof. Let Φ be an arbitrary consistent set of formulas. Assume that Φ is unsatisfiable. Then, the tableau T constructed from Φ must be closed. Prune each branch of T at the point at which it closes. This pruned tree is finite (by König's Lemma) and is thus contained in T_n for some n . Now, consider the

set of formulas $\Phi_n = \{\phi_1, \dots, \phi_n\}$, the first n formulas in the enumeration of Φ . Since Φ_n is a finite subset of Φ , Φ_n is consistent. Thus, for any new nominal i , $\neg((\phi_1 **i) \wedge \dots \wedge (\phi_n **i))$ is not provable. But, in fact, we will demonstrate that $\neg((\phi_1 **i) \wedge \dots \wedge (\phi_n **i))$ is provable. Form a tableau with $\neg(\neg((\phi_1 **i) \wedge \dots \wedge (\phi_n **i))) **j$, where j is a new nominal, as the root. Apply the $[\neg(\vee)]$ rule $n - 1$ times, yielding $\neg(\neg(\phi_1 **i)) **j, \dots, \neg(\neg(\phi_n **i)) **j$ along the initial branch of the tableau. Now, apply the $[\neg(\neg **)]$ rule to each of these formulas, yielding $\phi_1 **i, \dots, \phi_n **i$ along the initial branch of the tableau. Now, we can apply the sequence of rule applications used to generate the closed tableau T_n to this tableau, yielding a closed tableau. □

5 Conclusion

Recasting expressive situation logics in a modal framework in which the inclusion ordering on situation provides interrelated unary and binary modalities provides several benefits. In addition to capturing the characterization relation we would like to model, the diamond modality allows us to represent a support relation, and the box modality provides a simple way of expressing homogeneity, or inward persistence, which is an important property of eventualities that typically distinguishes atelic eventualities from telic ones.

References

- [1] Ahn, D. and L. Schubert, *A hybrid logic reconstruction of an expressive situation logic*, Technical report, University of Rochester Computer Science Dept. (2003).
- [2] Barwise, J. and J. Perry, “Situations and Attitudes,” MIT Press, 1983.
- [3] Blackburn, P., *Internalizing labelled deduction*, Journal of Logic and Computation **10** (2000), pp. 137–168.
- [4] Cooper, R., *Austinian propositions, Davidsonian events and perception complements*, in: *Tbilisi Symposium on Language, Logic and Computation*, CSLI, 1998 .
- [5] Davidson, D., *The logical form of action sentences*, in: *Essays on Actions and Events*, Oxford University Press, 1967 .
- [6] Schubert, L., *The situations we talk about*, in: J. Minker, editor, *Logic-Based Artificial Intelligence*, Kluwer Academic Publishers, 2000 .
- [7] Schubert, L. and C. H. Hwang, *Episodic Logic meets Little Red Riding Hood*, in: *Natural Language Processing and Knowledge Representation*, MIT Press, 2000 .
- [8] Seligman, J., *The logic of correct description*, in: M. de Rijke, editor, *Advances in Intensional Logic*, Kluwer, 1997 .
- [9] Smullyan, R., “First-Order Logic,” Springer-Verlag, 1968.

$$\frac{(\phi ** j) ** i}{\phi ** j} [**] \quad \frac{\neg((\phi ** j) ** i)}{\neg(\phi ** j)} [\neg(**)]$$

$$\frac{\neg(\phi ** j) ** i}{\neg(\phi ** j)} [\neg(**)] \quad \frac{\neg(\neg(\phi ** j) ** i)}{\phi ** j} [\neg(\neg(**))]$$

Fig. 1. Characterization rules

$$\frac{(\diamond \phi) ** i}{(\diamond \ell) ** i \ \phi ** \ell} [\diamond] \quad \frac{\neg((\diamond \phi) ** i) \ (\diamond j) ** i}{\neg(\phi ** j)} [\neg(\diamond)]$$

$$\frac{(\square \phi) ** i \ (\diamond j) ** i}{\phi ** j} [\square] \quad \frac{\neg((\square \phi) ** i)}{(\diamond \ell) ** i} [\neg(\square)]$$

$$\neg(\phi ** \ell)$$

Fig. 2. Unary modality rules

$$\frac{(\phi \circ \psi) ** i \ [\phi \text{ and } \psi \text{ not both nominals}]}{(\ell_1 \circ \ell_2) ** i} [\circ] \quad \frac{\neg((\phi \circ \psi) ** i) \ (\diamond j) ** i \ (\diamond k) ** i}{\neg(\phi ** j) \mid \neg(\psi ** j) \mid \neg((j \circ k) ** i)} [\neg(\circ)]$$

$$\frac{\phi ** \ell_1}{\psi ** \ell_2} \quad \frac{\neg(\phi ** k) \mid \neg(\psi ** k)}{\neg(\phi ** \ell_1) \mid \neg(\psi ** \ell_2)}$$

$$\frac{(\phi ? \psi) ** i \ (j \circ k) ** i}{\phi ** j \mid \psi ** k} [?] \quad \frac{\neg((\phi ? \psi) ** i)}{(\ell_1 \circ \ell_2) ** i} [\neg(?)]$$

$$\neg(\phi ** \ell_1) \quad \neg(\psi ** \ell_2)$$

Fig. 3. Binary modality rules

$$\frac{(\phi \wedge \psi) ** i}{\phi ** i} [\wedge] \quad \frac{\neg((\phi \wedge \psi) ** i)}{\neg(\phi ** i) \mid \neg(\psi ** i)} [\neg(\wedge)]$$

$$\psi ** i$$

$$\frac{(\phi \vee \psi) ** i}{\phi ** i \mid \psi ** i} [\vee] \quad \frac{\neg((\phi \vee \psi) ** i)}{\neg(\phi ** i)} [\neg(\vee)]$$

$$\neg(\psi ** i)$$

Fig. 4. Boolean rules

$$\frac{[i \text{ on branch}]}{i ** i} [** \text{ Ref}] \quad \frac{j ** i}{i ** j} [** \text{ Sym}]$$

$$\frac{\neg(\neg j) ** i}{j ** i} [** \text{ Double - negation}] \quad \frac{j ** i \ \phi ** j}{\phi ** i} [** \text{ Nom}]$$

$$\frac{(\diamond j) ** i \ k ** j}{(\diamond k) ** i} [** / \diamond \text{ Bridge}] \quad \frac{(j \circ k) ** i \ j' ** j \ k' ** k}{(j' \circ k') ** i} [** / \circ \text{ Bridge}]$$

Fig. 5. Equality rules

$$\frac{[i \text{ on branch}]}{(\diamond i) ** i} [\diamond \text{ Ref}] \quad \frac{(\diamond j) ** i \ (\diamond k) ** j}{(\diamond k) ** i} [\diamond \text{ Trans}] \quad \frac{(\diamond j) ** i \ (\diamond i) ** j}{i ** j} [\diamond \text{ Antisym}]$$

Fig. 6. Part-of rules

$$\frac{(i \circ j) ** k}{(\diamond i) ** k} [\text{Join1}] \quad \frac{(i \circ j) ** k \ (\diamond i) ** k' \ (\diamond j) ** k'}{(\diamond k) ** k'} [\text{Join2}]$$

$$(\diamond j) ** k$$

$$\frac{i, j \text{ on branch}}{(i \circ j) ** \ell} [\text{Lattice}] \quad \frac{\neg((i \circ j) ** k) \ (\diamond i) ** k \ (\diamond j) ** k}{(i \circ j) ** \ell} [\neg \text{Join}]$$

$$\neg(k ** \ell)$$

Fig. 7. Join rules