# Hybrid Logic Tango

## University of Buenos Aires

## February 2008

Carlos Areces and Patrick Blackburn

TALARIS team
INRIA Nancy Grand Est
France
`areces@loria.fr`
`patrick.blackburn@loria.fr`

Lecture 1: 11 Febraury 2008

# Goals of the course

This course introduces and explores hybrid logic, a form of modal logic in which it is possible to name worlds (or times, or computational states, or situations, or nodes in parse trees, or people — indeed, whatever it is that the elements of Kripke Models are taken to represent).

The course has two main goals. The first is to convey, as clearly as possible, the ideas and intuitions that have guided the development of hybrid logic. The second is to go deeper into into a number of more technical aspects, notably completeness, complexity, and model theory.

By the end of the course you will have ample evidence that modal logic can be useful in a wide range of circumstances, and that hybrid logic is a particularly simple way of doing modal logic. Moreover, you will have a good understanding of *why* this is so.

# Course outline

**Monday:** **From modal logic to hybrid logic**

**Tuesday:** **The downarrow binder**

**Wednesday:** **Completeness**

**Thursday:** **Complexity**

**Friday:** **Model theory**

# Lecture 1: From modal logic to hybrid logic

In today's lecture we discuss:

- Orthodox modal logic — from an Amsterdam perspective.
- A problem with orthodox modal logic.
- Fixing this problem with basic hybrid logic.
- Why basic hybrid logic is genuinely modal: bisimulations.
- Why basic hybrid logic is good for you: tableau systems.

# What is modal logic?

# What is modal logic?

Slogan 1: Modal languages are simple yet expressive languages for talking about relational structures.

# What is modal logic?

Slogan 1: Modal languages are simple yet expressive languages for talking about relational structures.

Slogan 2: Modal languages provide an internal, local perspective on relational structures.

# What is modal logic?

Slogan 1: Modal languages are simple yet expressive languages for talking about relational structures.

Slogan 2: Modal languages provide an internal, local perspective on relational structures.

Slogan 3: Modal languages are not isolated formal systems.

# What is modal logic?

Slogan 1: Modal languages are simple yet expressive languages for talking about relational structures.

Slogan 2: Modal languages provide an internal, local perspective on relational structures.

Slogan 3: Modal languages are not isolated formal systems.

These slogans pretty much sum up the Amsterdam perspective on modal logic.

# Propositional Modal Logic

Given propositional symbols $\mathrm{PROP} = \{\mathrm{p}, \mathrm{q}, \mathrm{r}, \ldots\}$, and modality symbols $\mathrm{MOD} = \{m, m', m'', \ldots\}$ the basic modal language (over PROP and MOD) is defined as follows:

$$\begin{array}{rcl} \mathrm{WFF} & := & \mathrm{p} \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \\ & & \mid \varphi \rightarrow \psi \mid \langle m \rangle \varphi \mid [m]\varphi \end{array}$$

If there's just one modality symbol in the language, we usually write $\Diamond$ and $\Box$ for its diamond and box forms.

$[m]\varphi$ can be regarded as shorthand for $\neg\langle m \rangle \neg\varphi$. Sometimes useful to add primitive atomic symbols $\top$ (true) and $\bot$ (false).

# Kripke Models

- A Kripke model $\mathcal{M}$ is a triple $(W, \mathcal{R}, V)$, where:
  - $W$ is a non-empty set, whose elements can be thought of possible worlds, or epistemic states, or times, or states in a transition system, or geometrical points, or people standing in various relationships, or nodes in a parse tree — indeed, pretty much anything you like.
  - $\mathcal{R}$ is a collection of binary relation on $W$ (one for each modality)
  - $V$ is a valuation assigning subsets of $W$ to propositional symbols.
- The component $(W, \mathcal{R})$ traditionally call a frame.

# Satisfaction Definition

$$
\begin{array}{lll}
\mathcal{M}, w \Vdash \mathrm{p} & \text{iff} & w \in V(\mathrm{p}), \text{ where } \mathrm{p} \in \mathrm{PROP} \\
\mathcal{M}, w \Vdash \neg\varphi & \text{iff} & \mathcal{M}, w \nVdash \varphi \\
\mathcal{M}, w \Vdash \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \Vdash \varphi \text{ and } \mathcal{M}, w \Vdash \psi \\
\mathcal{M}, w \Vdash \varphi \vee \psi & \text{iff} & \mathcal{M}, w \Vdash \varphi \text{ or } \mathcal{M}, w \Vdash \psi \\
\mathcal{M}, w \Vdash \varphi \rightarrow \psi & \text{iff} & \mathcal{M}, w \nVdash \varphi \text{ or } \mathcal{M}, w \Vdash \psi \\
\mathcal{M}, w \Vdash \langle m \rangle\varphi & \text{iff} & \exists w'(wR^m w' \;\&\; \mathcal{M}, w' \Vdash \varphi) \\
\mathcal{M}, w \Vdash [m]\varphi & \text{iff} & \forall w'(wR^m w' \;\Rightarrow\; \mathcal{M}, w' \Vdash \varphi).
\end{array}
$$

*Note the internal perspective: we evaluate formulas inside models, at particular states. Modal formulas are like little creatures that explore models by moving between related points. This is a key modal intuition, gives rise to the notion of bisimulation, and is the driving force for at least one traditional application.*

# Tense logic

- $\langle \text{F} \rangle$ means "at some *F*uture state", and $\langle \text{P} \rangle$ means "at some *P*ast state".
- $\langle \text{P} \rangle$ mia-unconscious is true iff we can look back in time from the current state and see a state where Mia is unconscious. Works a bit like the sentence *Mia has been unconscious*.
- $\langle \text{F} \rangle$ mia-unconscious requires us to scan the states that lie in the future looking for one where Mia is unconscious. Works a bit like the sentence *Mia will be unconscious*.

# Feature logic

Consider the following Attribute Value Matrix (AVM):

$$\begin{bmatrix} \textsc{agreement} & \begin{bmatrix} \textsc{person} & \textit{1st} \\ \textsc{number} & \textit{der} \end{bmatrix} \\ \textsc{case} & -\textit{dative} \end{bmatrix}$$

# Feature logic

Consider the following Attribute Value Matrix (AVM):

$$\begin{bmatrix} \textsc{agreement} & \begin{bmatrix} \textsc{person} & \textit{1st} \\ \textsc{number} & \textit{der} \end{bmatrix} \\ \textsc{case} & -\textit{dative} \end{bmatrix}$$

This is a notational variant of the following modal formula:

$$\begin{array}{ll} & \langle \textsc{agreement} \rangle \, (\langle \textsc{person} \rangle \, \text{1st} \wedge \langle \textsc{number} \rangle \, \text{singular}) \\ \wedge & \langle \textsc{case} \rangle \, \neg \text{dative} \end{array}$$

# Description logic

And, moving into the heart of ordinary *extensional* logic, consider the following $\mathcal{ALC}$ term:

$$\mathsf{killer} \sqcap \exists \textsc{employer}.\mathsf{gangster}$$

# Description logic

And, moving into the heart of ordinary *extensional* logic, consider the following $\mathcal{ALC}$ term:

$$\mathsf{killer} \sqcap \exists \textsc{employer}.\mathsf{gangster}$$

This means exactly the same thing as the modal formula:

$$\mathsf{killer} \wedge \langle \textsc{employer} \rangle \mathsf{gangster}$$

# But there's lots of other ways of talking about graphs

- There's nothing magic about frames or Kripke models.
- Frames $(W, \mathcal{R})$, are just a directed multigraphs (or labelled transition systems).
- Valuations simply decorate states with properties.
- So a Kripke model for the basic modal language are just (very simple) relational structures in the usual sense of first-order model theory.
- So we don't have to talk about Kripke models using modal logic — we could use first-order logic, or second-order logic, or infinitary logic, or fix-point logic, or indeed any logic interpreted over relational structures.
- Let's see how…

# First-order logic for Kripke models

Suppose we have a Kripke model $(W, \mathcal{R}, V)$, for the modal language over MOD and PROP. We talk about this model in first-order logic by making use of the first-order language built from the following symbols:

- For each propositional symbol p it has a unary predicate symbol P. We'll use $V$ to interpret these predicate symbols.
- For each modality $\langle \mathrm{R} \rangle$, it has a binary relation symbol R. We'll use the binary relations in $\mathcal{R}$ to interpret these symbols.

The first-order language built over these symbols is called the first-order correspondence language (for the modal language over MOD and PROP).

# Doing it first-order style (I)

Consider the modal representation

$$\langle \mathrm{F} \rangle \, \text{mia} - \text{unconscious}$$

# Doing it first-order style (I)

Consider the modal representation

$$\langle \text{F} \rangle \, \text{mia} - \text{unconscious}$$

we could use instead the first-order representation

$$\exists t (t_o < t \wedge \text{MIA} - \text{UNCONSCIOUS}(t)).$$

# Doing it first-order style (II)

And consider the modal representation

$$\mathsf{killer} \wedge \langle \textsc{employer} \rangle \mathsf{gangster}$$

# Doing it first-order style (II)

And consider the modal representation

$$\mathsf{killer} \wedge \langle \textsc{employer} \rangle \mathsf{gangster}$$

We could use instead the first-order representation

$$\mathsf{KILLER}(x) \wedge \exists y (\mathrm{EMPLOYER}(x, y) \wedge \mathsf{GANGSTER}(y))$$

# Standard Translation

And in fact, any modal representation can by converted into an equi-satisfiable first-order representation:

$$
\begin{array}{lcl}
\textsc{st}_x(\mathrm{p}) & = & \mathrm{P}x \\
\textsc{st}_x(\neg\varphi) & = & \neg\,\textsc{st}_x(\varphi) \\
\textsc{st}_x(\varphi \wedge \psi) & = & \textsc{st}_x(\varphi) \wedge \textsc{st}_x(\psi) \\
\textsc{st}_x(\langle R\rangle\varphi) & = & \exists y(Rxy \wedge \textsc{st}_y(\varphi))
\end{array}
$$

Note that $\textsc{st}_x(\varphi)$ always contains exactly one free variable (namely $x$).

**Proposition:** For any modal formula $\varphi$, any Kripke model $\mathcal{M}$, and any state $w$ in $\mathcal{M}$ we have that: $\mathcal{M}, w \Vdash \varphi$ iff $\mathcal{M} \models \textsc{st}_x(\varphi)[x \leftarrow w]$.

# So aren't we better off with first-order logic . . . ?

- We've just seen that any modal formula can be systematically converted into an equi-satisfiable first-order formula.
- And as we'll later see, the reverse is not possible: first-order logic can describe models in far more detail that modal logic can. Some first-order formulas have no modal equivalent. That is, modal languages are weaker than their corresponding first-order languages.
- So why bother with modal logic?

# Reasons for going modal

# Reasons for going modal

- **Simplicity**. The standard translation shows us that modalities are essentially 'macros' encoding a quantification over related states. Modal notation hides the bound variables, resulting in a compact, easy to read, representations.

# Reasons for going modal

- **Simplicity**. The standard translation shows us that modalities are essentially 'macros' encoding a quantification over related states. Modal notation hides the bound variables, resulting in a compact, easy to read, representations.
- **Computability**. First-order logic is undecidable over arbitrary models. Modal logic is decidable over arbitrary models (indeed, decidable in PSPACE). Modal logic trades expressivity for computability.

# Reasons for going modal

- **Simplicity**. The standard translation shows us that modalities are essentially 'macros' encoding a quantification over related states. Modal notation hides the bound variables, resulting in a compact, easy to read, representations.
- **Computability**. First-order logic is undecidable over arbitrary models. Modal logic is decidable over arbitrary models (indeed, decidable in PSPACE). Modal logic trades expressivity for computability.
- **Internal perspective**. A natural way of thinking about models. And taken seriously, leads to an elegant characterization of what modal logic can say about models. Let's take a closer look...

# Bisimulation (I)

# Bisimulation (I)

The fundamental notion of equivalence between states for modal logic.

# Bisimulation (I)

The fundamental notion of equivalence between states for modal logic.

Bisimulations are used in other disciplines besides modal logic. Its role in all of them is to provide an appropriate notion of equivalence.

# Bisimulation (I)

The fundamental notion of equivalence between states for modal logic.

Bisimulations are used in other disciplines besides modal logic. Its role in all of them is to provide an appropriate notion of equivalence.

**Social Network Theory** Here they capture the notion of two social networks being functionally identical, even though they are not isomorphic. It's the configurations in which the agents stand in various roles that render two social networks "the same".

# Bisimulation (I)

The fundamental notion of equivalence between states for modal logic.

Bisimulations are used in other disciplines besides modal logic. Its role in all of them is to provide an appropriate notion of equivalence.

**Social Network Theory** Here they capture the notion of two social networks being functionally identical, even though they are not isomorphic. It's the configurations in which the agents stand in various roles that render two social networks "the same".

**Theoretical Computer Science** Here they embody the notion of behavioural equivalence for processes.

# Bisimulation (I)

The fundamental notion of equivalence between states for modal logic.

Bisimulations are used in other disciplines besides modal logic.

Its role in all of them is to provide an appropriate notion of equivalence.

**Social Network Theory** Here they capture the notion of two social networks being functionally identical, even though they are not isomorphic. It's the configurations in which the agents stand in various roles that render two social networks "the same".

**Theoretical Computer Science** Here they embody the notion of behavioural equivalence for processes.

**Non-well-founded Set Theory** Here they replace the extensionality as the criterion of equality: two non-well-founded sets (graphs) are equal iff they are bisimilar.

# Bisimulation (II)

Let $\mathcal{M} = (W, \mathcal{R}, V)$ and $\mathcal{M}' = (W', \mathcal{R}', V')$ be models for the same basic modal language. A relation $Z \subseteq W \times W'$ is a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ if the following conditions are met:

# Bisimulation (II)

Let $\mathcal{M} = (W, \mathcal{R}, V)$ and $\mathcal{M}' = (W', \mathcal{R}', V')$ be models for the same basic modal language. A relation $Z \subseteq W \times W'$ is a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ if the following conditions are met:

1. Atomic harmony: if $wZw'$ then $w \in V(p)$ iff $w' \in V'(p)$, for all propositional symbols $p$.

# Bisimulation (II)

Let $\mathcal{M} = (W, \mathcal{R}, V)$ and $\mathcal{M}' = (W', \mathcal{R}', V')$ be models for the same basic modal language. A relation $Z \subseteq W \times W'$ is a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ if the following conditions are met:

1. Atomic harmony: if $wZw'$ then $w \in V(p)$ iff $w' \in V'(p)$, for all propositional symbols $p$.
2. Forth: if $wZw'$ and $wRv$ then there is a $v'$ such that $w'R'v'$ and $vZv'$.

# Bisimulation (II)

Let $\mathcal{M} = (W, \mathcal{R}, V)$ and $\mathcal{M}' = (W', \mathcal{R}', V')$ be models for the same basic modal language. A relation $Z \subseteq W \times W'$ is a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ if the following conditions are met:

1. Atomic harmony: if $wZw'$ then $w \in V(p)$ iff $w' \in V'(p)$, for all propositional symbols $p$.
2. Forth: if $wZw'$ and $wRv$ then there is a $v'$ such that $w'R'v'$ and $vZv'$.
3. Back: if $wZw'$ and $w'R'v'$ then there is a $v$ such that $wRv$ and $vZv'$.

# Modal formulas are invariant under bisimulation

**Proposition:** Let $\mathcal{M} = (W, \mathcal{R}, V)$ and $\mathcal{M}' = (W', \mathcal{R}', V')$ be models for the same basic modal language, and let $Z$ be a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$. Then for all modal formulas $\varphi$, and all points $w$ in $\mathcal{M}$ and $w'$ in $\mathcal{M}$ such that $w$ is bisimilar to $w'$:

$$\mathcal{M}, w \Vdash \varphi \text{ iff } \mathcal{M}', w' \Vdash \varphi.$$

In words: bisimilar points are modally equivalent, or to put it another way: modal formulas are invariant under bisimulations.

**Proof:** Induction on the structure of $\varphi$.

# Not all first-order formulas are bisimulation invariant

- A first-order formula in one free variable $\varphi(x)$ is bisimulation-invariant if for all bisimulations $Z$ between models $\mathcal{M}$ and $\mathcal{M}'$, if $wZw'$ then $\mathcal{M} \models \varphi[w]$ iff $\mathcal{M}' \models \varphi[w']$.
- Not all first-order formulas are bisimulation invariant (which shows that not all first-order formulas can be translated into modal formulas).
- But bisimulation invariance seems to be a natural property in various domains, so it is natural to ask: precisely which first-order formulas are bisimulation invariant? The answer is elegant ...

# The van Benthem Characterization Theorem

For all first-order formulas $\varphi$ (in the correspondence language) containing exactly one free variable, $\varphi$ is bisimulation-invariant iff $\varphi$ is equivalent to the standard translation of a modal formula.

In short, modal logic is a simple notation for capturing exactly the bisimulation-invariant fragment of first-order logic.

**Proof:**

($\Rightarrow$) Immediate from the invariance of modal formula under bisimulation.

($\Leftarrow$) Non-trivial (usually proved using elementary chains or by appealing to the existence of saturated models).

# Back to slogan 3

Slogan 3: Modal languages are not isolated formal systems.

Modal languages over models are essentially simple fragments of first-order logic. These fragments have a number of attractive properties such as robust decidability and bisimulation invariance. Traditional modal notation is essentially a nice (quantifier free) 'macro' notation for working with this fragment.

# Back to slogan 2

Slogan 2: Modal languages provide an internal, local perspective on relational structures.

This is not just an intuition: the notion of bisimulation, and the results associated with it, shows that this is the key model theoretic fact at work in modal logic.

# Back to slogan 1

Slogan 1: Modal languages are simple yet expressive languages for talking about relational structures.

You can use modal logic for just about anything. Anywhere you see a graph, you can use a modal language to talk about it.

# That was the good news — now comes the bad

Orthodox modal languages have an obvious drawback for many applications: they don't let us refer to individual states (worlds, times, situations, nodes, ...). That is, they don't allow us to say things like

- this happened *there*; or
- this happened *then*; or
- *this* state has property $\varphi$; or
- node $i$ is marked with the information $p$.

and so on.

# Temporal logic

- Temporal representations in Artificial Intelligence (such as Allen's system, and the situation calculus) based around temporal reference — and for good reasons.
- Worse, standard modal logics of time are completely inadequate for the temporal semantics of natural language. *Vincent accidentally squeezed the trigger* doesn't mean that at some completely unspecified past time Vincent did in fact accidentally squeeze the trigger, it means that at some *particular*, contextually determined, past time he did so. The representation, $\langle \text{P} \rangle\, vincent-accidentally-squeeze-trigger$ fails to capture this.

# Tense in text

*Vincent woke up. Something felt very wrong. Vincent reached under his pillow for his Uzi.*

The states described by the first two sentences hold at the same time. The event described by the second takes place a little later. In orthodox modal logics there is no way assert the identity of the times needed for the first two sentences, nor to capture the move forward in time needed by the third.

In fact, for this reason modal languages for temporal representation have not been the tool of choice in natural language semantics for over 15 years.

# Feature logic

As we've mentioned, the following Attribute Value Matrix (AVM):

$$\begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & \mathit{1st} \\ \text{NUMBER} & \mathit{plural} \end{bmatrix} \\ \text{CASE} & -\mathit{dative} \end{bmatrix}$$

is a notational variant of the following modal formula:

$$\begin{array}{ll} & \langle \text{AGREEMENT} \rangle \, (\langle \text{PERSON} \rangle \, \text{1st} \wedge \langle \text{NUMBER} \rangle \, \text{plural}) \\ \wedge & \langle \text{CASE} \rangle \, \neg \text{dative} \end{array}$$

# Feature logic

But full AVM notation is richer. It can assert re-entrancies:

$$\begin{bmatrix} \text{SUBJ} & \boxed{1}\begin{bmatrix} \text{AGR} & \textit{foo} \\ \text{PRED} & \textit{bar} \end{bmatrix} \\ \text{COMP} & \begin{bmatrix} \text{SUBJ} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

This cannot be captured in orthodox modal logic.

# Description logic I

As we have already said, there is a transparent correspondence between simple DL terms and modal formulas:

$$\mathsf{killer} \sqcap \exists \textsc{employer}.\mathsf{gangster}$$

$$\mathsf{killer} \wedge \langle \textsc{employer} \rangle \mathsf{gangster}$$

Nonetheless, this correspondence only involves what description logicians call the TBox (Terminological Box).

Orthodox modal logic does not have anything to say about the ABox (Assertional Box):

$$\mathsf{mia} : \mathsf{Beautiful}$$

$$(\mathsf{jules}, \mathsf{vincent}) : \mathsf{Friends}$$

That is, it can't make assertions about individuals, for it has no tools for naming individuals.

# Ambition

- Want to be able to refer to states, but want to do so without destroying the simplicity of propositional modal logic.
- But how can we do this — propositional modal logic has very few moving parts?
- Answer: sort the atomic symbols. Use formulas as terms.
- This will fix the obvious shortcoming — and as we shall learn, it will fix a lot more besides.

# Extension #1

- Take a language of basic modal logic (with propositional variables p, q, r, and so on) and add a second sort of atomic formula.
- The new atoms are called nominals, and are typically written $i$, $j$, $k$, and $l$.
- Both types of atom can be freely combined to form more complex formulas in the usual way; for example,

$$\Diamond(i \wedge \mathrm{p}) \wedge \Diamond(i \wedge \mathrm{q}) \rightarrow \Diamond(\mathrm{p} \wedge \mathrm{q})$$

  is a well formed formula.
- Insist that each nominal be true at exactly one world in any model. A nominal names a state by being true there and nowhere else.

# We already have a richer logic

Consider the orthodox formula

$$\Diamond(r \wedge p) \wedge \Diamond(r \wedge q) \rightarrow \Diamond(p \wedge q)$$

This is easy to falsify.

# We already have a richer logic

Consider the orthodox formula

$$\Diamond(\mathrm{r} \wedge \mathrm{p}) \wedge \Diamond(\mathrm{r} \wedge \mathrm{q}) \rightarrow \Diamond(\mathrm{p} \wedge \mathrm{q})$$

This is easy to falsify.

On the other hand, the hybrid formula

$$\Diamond(i \wedge \mathrm{p}) \wedge \Diamond(i \wedge \mathrm{q}) \rightarrow \Diamond(\mathrm{p} \wedge \mathrm{q})$$

is valid (unfalsifiable). Nominals name, and this adds to the expressive power at our disposal.

# Extension #2

- Add formulas of form $@_i\varphi$.
- Such formulas assert that $\varphi$ is satisfied at the point named by the nominal $i$.
- Expressions of the form $@_i$ are called satisfaction operators.

Let's make these ideas precise . . .

# Syntax

- Given ordinary propositional symbols PROP $= \{\mathrm{p}, \mathrm{q}, \mathrm{r}, \ldots\}$, and modalities MOD, let NOM $= \{i, j, k, l, \ldots\}$ be a nonempty set disjoint from PROP.
- The elements of NOM are called nominals; they are second sort of atomic symbol which will be used to name states. g The basic hybrid language (over PROP, MOD and NOM) is defined as follows:

$$\begin{aligned} \mathrm{WFF} \quad := \quad & i \mid \mathrm{p} \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \\ & \mid \varphi \rightarrow \psi \mid \langle \mathrm{M} \rangle \varphi \mid [\mathrm{M}] \varphi \mid @_i \varphi \end{aligned}$$

# Semantics

- As before, a model $\mathcal{M}$ is a triple $(W, \mathcal{R}, V)$.
- As before, $(W, \mathcal{R})$ is just a frame (a labelled transition system).
- The difference lies in $V$. Now a valuation $V$ is a function with domain PROP$\cup$NOM and range $Pow(W)$ such that for all $i \in$ NOM, $V(i)$ is a singleton subset of $W$.
- That is, a valuation makes each nominal true at a unique state; the nominal labels this state by being true there and nowhere else.
- We call the unique state $w$ that belongs to $V(i)$ the denotation of $i$ under $V$.

# Satisfaction Definition

| | | |
|---|---|---|
| $\mathcal{M}, w \Vdash a$ | iff | $w \in V(a)$, where $a \in \text{PROP} \cup \text{NOM}$ |
| $\mathcal{M}, w \Vdash \neg\varphi$ | iff | $\mathcal{M}, w \not\Vdash \varphi$ |
| $\mathcal{M}, w \Vdash \varphi \wedge \psi$ | iff | $\mathcal{M}, w \Vdash \varphi$ and $\mathcal{M}, w \Vdash \psi$ |
| $\mathcal{M}, w \Vdash \varphi \vee \psi$ | iff | $\mathcal{M}, w \Vdash \varphi$ or $\mathcal{M}, w \Vdash \psi$ |
| $\mathcal{M}, w \Vdash \varphi \rightarrow \psi$ | iff | $\mathcal{M}, w \not\Vdash \varphi$ or $\mathcal{M}, w \Vdash \psi$ |
| $\mathcal{M}, w \Vdash \langle\text{M}\rangle\,\varphi$ | iff | $\exists w'(wR^m w'\ \&\ \mathcal{M}, w' \Vdash \varphi)$ |
| $\mathcal{M}, w \Vdash [\text{M}]\,\varphi$ | iff | $\forall w'(wR^m w' \ \Rightarrow\ \mathcal{M}, w' \Vdash \varphi)$. |
| $\mathcal{M}, w \Vdash @_i\varphi$ | iff | $\mathcal{M}, \mathrm{i} \Vdash \varphi$, where i is the denotation of $i$ under $V$. |

# Tense logic

- On the road to capturing AI temporal representation formalisms such as Allen's logic of temporal reference; @ can play the role of **Holds**.
- And we can now handle natural language examples more convincingly: $\langle \mathsf{P} \rangle (i \wedge \mathsf{Vincent\text{-}accidentally\text{-}squeeze\text{-}the\text{-}trigger})$ locates the trigger-squeezing not merely in the past, but at a specific temporal state there, namely the one named by $i$ — capturing the meaning of *Vincent accidentally squeezed the trigger*. Let's take this a little further...

# Reichenbach in hybrid logic

| Structure | Name | English example | Representatio |
|---|---|---|---|
| E–R–S | Pluperfect | I had seen | $\langle P \rangle (i \wedge \langle P \rangle \phi)$ |
| E,R–S | Past | I saw | $\langle P \rangle (i \wedge \phi)$ |
| R–E–S | Future-in-the-past | I would see | $\langle P \rangle (i \wedge \langle F \rangle \phi)$ |
| R–S,E | Future-in-the-past | I would see | $\langle P \rangle (i \wedge \langle F \rangle \phi)$ |
| R–S–E | Future-in-the-past | I would see | $\langle P \rangle (i \wedge \langle F \rangle \phi)$ |
| E–S,R | Perfect | I have seen | $\langle P \rangle \phi$ |
| S,R,E | Present | I see | $\phi$ |
| S,R–E | Prospective | I am going to see | $\langle F \rangle \phi$ |
| S–E–R | Future perfect | I will have seen | $\langle F \rangle (i \wedge \langle P \rangle \phi)$ |
| S,E–R | Future perfect | I will have seen | $\langle F \rangle (i \wedge \langle P \rangle \phi)$ |
| E–S–R | Future perfect | I will have seen | $\langle F \rangle (i \wedge \langle P \rangle \phi)$ |
| S–R,E | Future | I will see | $\langle F \rangle (i \wedge \phi)$ |
| S–R–E | Future-in-the-future | (Latin: abiturus ero) | $\langle F \rangle (i \wedge \langle F \rangle \phi)$ |

# Tense in text

*Vincent woke up. Something felt very wrong. Vincent reached under his pillow for his Uzi.*

# Tense in text

*Vincent woke up. Something felt very wrong. Vincent reached under his pillow for his Uzi.*

$$P(i \wedge \text{vincent-wake-up})$$

# Tense in text

*Vincent woke up. Something felt very wrong. Vincent reached under his pillow for his Uzi.*

$$
\begin{aligned}
& P(i \wedge \text{vincent-wake-up}) \\
\wedge \quad & P(j \wedge \text{something-feel-very-wrong})
\end{aligned}
$$

# Tense in text

*Vincent woke up. Something felt very wrong. Vincent reached under his pillow for his Uzi.*

$$
\begin{aligned}
& P(i \wedge \text{vincent-wake-up}) \\
\wedge \quad & P(j \wedge \text{something-feel-very-wrong}) \qquad \wedge \ @_j i
\end{aligned}
$$

# Tense in text

*Vincent woke up. Something felt very wrong. Vincent reached under his pillow for his Uzi.*

$$
\begin{array}{lll}
 & P(i \wedge \text{vincent-wake-up}) & \\
\wedge & P(j \wedge \text{something-feel-very-wrong}) & \wedge\ @_j i \\
\wedge & P(k \wedge \text{vincent-reach-under-pillow-for-uzi}) &
\end{array}
$$

# Tense in text

*Vincent woke up. Something felt very wrong. Vincent reached under his pillow for his Uzi.*

$$
\begin{array}{lll}
 & P(i \wedge \text{vincent-wake-up}) & \\
\wedge & P(j \wedge \text{something-feel-very-wrong}) & \wedge\ @_j i \\
\wedge & P(k \wedge \text{vincent-reach-under-pillow-for-uzi}) & \wedge\ @_k P i
\end{array}
$$

# Feature logic

$$\begin{bmatrix} \textsc{subj} & \boxed{1}\begin{bmatrix} \textsc{agr} & \textit{foo} \\ \textsc{pred} & \textit{bar} \end{bmatrix} \\ \textsc{comp} & [\textsc{subj}\ \boxed{1}\ ] \end{bmatrix}$$

This corresponds to the following hybrid wff:

$$\begin{array}{ll} & \langle \textsc{subj} \rangle\, (i \wedge \langle \textsc{agr} \rangle\, \text{foo} \wedge \langle \textsc{pred} \rangle\, \rangle \text{bar}) \\ \wedge & \langle \textsc{comp} \rangle\, \langle \textsc{subj} \rangle\, i \end{array}$$

# Description logic (I)

We can now make ABox statements. For example, to capture the effect of the (conceptual) ABox assertion

$$\mathsf{mia} : \mathsf{Beautiful}$$

we can write

$$@_{\mathsf{mia}}\mathsf{Beautiful}$$

# Description logic (II)

Similarly, to capture the effect of the (relational) ABox assertion

$$(\mathsf{jules}, \mathsf{vincent}) : \mathsf{Friends}$$

we can write

$$@_{\mathsf{jules}}\langle\mathsf{Friends}\rangle\mathsf{vincent}$$

# Basic hybrid language clearly modal

Neither syntactical nor computational simplicity, nor general 'style' of modal logic, has been compromised.

- Nominals just atomic formulas.
- Satisfaction operators are normal modal operators. That is, for any nominal $i$ we have that:
  - $@_i(\varphi \to \psi) \to (@_i\varphi \to @_i\psi)$ is valid.
  - If $\varphi$ is valid, then so is $@_i\varphi$.
- Indeed, satisfaction operators are even self-dual modal operators: $@_i\phi$ and $\neg @_i \neg \phi$ say exactly the same thing.

# Basic hybrid logic is computable

Enriching ordinary propositional modal logic with both nominals and satisfaction operators does not effect computability. The basic hybrid logic is decidable. Indeed we even have:

**Theorem:** The satisfiability problem for basic hybrid languages over arbitrary models is PSPACE-complete (Areces, Blackburn, and Marx).

That is (up to a polynomial) the hybridized language has the same complexity as the orthodox modal language we started with.

# Standard Translation

Any basic hybrid formula can by converted into an equi-satisfiable first-order formula. All we have to do is add a first-order constant (or variable) $i$ for each nominal $i$ and translate as follows (note the use of equality):

$$
\begin{array}{lcl}
\mathrm{ST}_x(\mathrm{p}) & = & \mathrm{P}x \\
\mathrm{ST}_x(i) & = & (i = x) \\
\mathrm{ST}_x(\neg\varphi) & = & \neg\,\mathrm{ST}_x(\varphi) \\
\mathrm{ST}_x(\varphi \wedge \psi) & = & \mathrm{ST}_x(\varphi) \wedge \mathrm{ST}_x(\psi) \\
\mathrm{ST}_x(\langle R\rangle\varphi) & = & \exists y(Rxy \wedge \mathrm{ST}_y(\varphi)) \\
\mathrm{ST}_x(@_i\varphi) & = & \mathrm{ST}_i(\varphi)
\end{array}
$$

Note that $\mathrm{ST}_x(\varphi)$ always contains at most free variable (namely $x$).

**Proposition:** For any basic hybrid formula $\varphi$, any Kripke model $\mathcal{M}$, and any state $w$ in $\mathcal{M}$ we have that:

$\mathcal{M}, w \Vdash \varphi$ iff $\mathcal{M} \models \mathrm{ST}_x(\varphi)[x \leftarrow w]$.

# Basic hybrid logic can state Robinson Diagrams

# Basic hybrid logic can state Robinson Diagrams

- $@_i p$ says that the states labelled $i$ bears the information $p$, while $\neg @_i p$ denies this. That is, we can specify how atomic properties are distributed modally.

# Basic hybrid logic can state Robinson Diagrams

- $@_i p$ says that the states labelled $i$ bears the information $p$, while $\neg @_i p$ denies this. That is, we can specify how atomic properties are distributed modally.
- $@_i j$ says that the states labelled $i$ and $j$ are identical, while $\neg @_i j$ says they are distinct. That is, we can specify theories of state equality modally.

# Basic hybrid logic can state Robinson Diagrams

- $@_i p$ says that the states labelled $i$ bears the information $p$, while $\neg @_i p$ denies this. That is, we can specify how atomic properties are distributed modally.
- $@_i j$ says that the states labelled $i$ and $j$ are identical, while $\neg @_i j$ says they are distinct. That is, we can specify theories of state equality modally.
- $@_i \Diamond j$ says that the state labelled $j$ is a successor of the state labelled $i$, and $\neg @_i \Diamond j$ denies this. That is, we can specify theories of state succession modally.

# Basic hybrid logic can state Robinson Diagrams

- $@_i p$ says that the states labelled $i$ bears the information $p$, while $\neg @_i p$ denies this. That is, we can specify how atomic properties are distributed modally.
- $@_i j$ says that the states labelled $i$ and $j$ are identical, while $\neg @_i j$ says they are distinct. That is, we can specify theories of state equality modally.
- $@_i \Diamond j$ says that the state labelled $j$ is a successor of the state labelled $i$, and $\neg @_i \Diamond j$ denies this. That is, we can specify theories of state succession modally.

That is, we have all the tools needed to completely describe models (that is, what model theorists call Robinson diagrams). This makes life very straightforward when it comes to proving completeness and interpolation results.

# Basic hybrid logic can state Robinson Diagrams

- $@_i p$ says that the states labelled $i$ bears the information $p$, while $\neg @_i p$ denies this. That is, we can specify how atomic properties are distributed modally.
- $@_i j$ says that the states labelled $i$ and $j$ are identical, while $\neg @_i j$ says they are distinct. That is, we can specify theories of state equality modally.
- $@_i \Diamond j$ says that the state labelled $j$ is a successor of the state labelled $i$, and $\neg @_i \Diamond j$ denies this. That is, we can specify theories of state succession modally.

That is, we have all the tools needed to completely describe models (that is, what model theorists call Robinson diagrams). This makes life very straightforward when it comes to proving completeness and interpolation results.

# But what *is* basic hybrid logic?

We have seen many examples of what basic hybrid logic can do in various applications.

We've also seen that a number of the properties we liked about modal logic are inherited by the basic hybrid language.

This is all very nice — but none of it gives us a clear mathematical characterization of what basic hybrid logic actually is.

And it is possible to give such a characterization, and a genuinely modal one at that. Let's take a look …

# Bisimulation-with-constants

Let $\mathcal{M} = (W, \mathcal{R}, V)$ and $\mathcal{M}' = (W', \mathcal{R}', V')$ be models for the same basic hybrid language. A relation $Z \subseteq W \times W'$ is a bisimulation-with-constants between $\mathcal{M}$ and $\mathcal{M}'$ if the following conditions are met:

# Bisimulation-with-constants

Let $\mathcal{M} = (W, \mathcal{R}, V)$ and $\mathcal{M}' = (W', \mathcal{R}', V')$ be models for the same basic hybrid language. A relation $Z \subseteq W \times W'$ is a bisimulation-with-constants between $\mathcal{M}$ and $\mathcal{M}'$ if the following conditions are met:

1. Atomic harmony: if $wZw'$ then $w \in V(p)$ iff $w' \in V'(p)$, for all propositional symbols $p$, and all nominals $i$.
2. Forth: if $wZw'$ and $wRv$ then there is a $v'$ such that $w'R'v'$ and $vZv'$.
3. Back: if $wZw'$ and $w'R'v'$ then there is a $v$ such that $wRv$ and $vZv'$.
4. All points named by nominals are related by $Z$.

# Basic hybrid formulas are invariant under bisimulations-with-constants

**Proposition:** Let $\mathcal{M} = (W, \mathcal{R}, V)$ and $\mathcal{M}' = (W', \mathcal{R}', V')$ be models for the same basic hybrid language, and let $Z$ be a bisimulation-with-constants between $\mathcal{M}$ and $\mathcal{M}'$. Then for all basic hybrid formulas $\varphi$, and all points $w$ in $\mathcal{M}$ and $w'$ in $\mathcal{M}$ such that $w$ is bisimilar to $w'$:

$$\mathcal{M}, w \Vdash \varphi \text{ iff } \mathcal{M}', w' \Vdash \varphi.$$

**Proof:** Induction on the structure of $\varphi$.

# Lifting the van Benthem Characterization theorem

For all first-order formulas $\varphi$ (in the correspondence language with constants and equality) containing at most one free variable, $\varphi$ is bisimulation-with-constants invariant iff $\varphi$ is equivalent to the standard translation of a basic hybrid formula iff (Areces, Blackburn, ten Cate, and Marx)

In short, basic hybrid logic is a simple notation for capturing exactly the bisimulation-invariant fragment of first-order logic when we make use of constants and equality.

**Proof:**
($\Rightarrow$) Immediate from the invariance of hybrid formulas under bisimulation.
($\Leftarrow$) Can be proved using elementary chains or by appealing to the existence of saturated models.

# Summing up . . .

- We learned about some of the good points of orthodox modal logic, but also saw that it's inability to refer to states is a weakness for various applications.
- We saw that adding nominals and satisfaction operators fixes these weaknesses without sacrificing what we liked about modal logic in the first place. Basic hybrid logic is a natural generalization of orthodox modal logic.

# Summing up . . .

- We learned about some of the good points of orthodox modal logic, but also saw that it's inability to refer to states is a weakness for various applications.
- We saw that adding nominals and satisfaction operators fixes these weaknesses without sacrificing what we liked about modal logic in the first place. Basic hybrid logic is a natural generalization of orthodox modal logic.
- But as we shall soon learn, hybridization has fixed some less obvious shortcomings of orthodox modal logic too. In particular, it has given us a logical formalism that is is easy to use deductively — as we shall see after a break!

# Hybrid deduction

Let's continue with an example-driven introduction to hybrid deduction. We concentrate on tableau systems. We shall:

- Discuss the goals and problems of orthodox modal deduction.
- Present a hybrid tableau system for reasoning about arbitrary models.
- Show how this can be extended to hybrid tableau systems for special classes of models.
- Round off by discussing further themes in hybrid deduction, including their implementation.

# Different models, different logics

Key fact about modal logic: when you work with different kinds of models (graphs) the logic typically changes. For example:

- $\Box p \wedge \Box q \rightarrow \Box(p \wedge q)$ is valid on all models: it's part of the basic, universally applicable, logic.
- But $\Diamond\Diamond p \rightarrow \Diamond p$ is only valid on transitive graphs. It's not part of the basic logic, rather it's part of the special (stronger) logic that we need to use when working with transitive models.

# Modal deduction should be general

- Quite rightly, modal logicians have insisted on developing proof methods which are general — that is, which can be easily adapted to cope with the logics of many kinds of models (transitive, reflexive, symmetric, dense, and so on).
- They achieve this goal by making use of Hilbert-style systems (that is, axiomatic systems).
- There is a basic axiomatic systems (called **K**) for dealing with arbitrary models.
- To deal with special classes of models, further axioms are added to **K**. For example, adding $\Diamond\Diamond p \rightarrow \Diamond p$ as an axiom gives us the logic of transitive frames.

# Generality clashes with easy of use

- Unfortunately, Hilbert systems are hard to use and completely unsuitable for computational implementation.
- For ease of use we want (say) natural deduction systems or tableau systems. For computational implementation we want (say) resolution systems or tableau systems.
- But it is hard to develop tableau, or natural deduction, or resolution in a general way in orthodox modal logic.
- Why is this?

# Getting behind the diamonds

- The difficulty is extracting information from under the scope of diamonds.
- That is, given $\Diamond\varphi$, how do we lay hands on $\varphi$? And given $\neg\Box\varphi$ (that is, $\Diamond\neg\varphi$), how do we lay hands on $\neg\varphi$?
- In first order logic, the analogous problem is trivial. There is a simple rule for stripping away existential quantifiers: from $\exists x\varphi$ we conclude $\varphi[x \leftarrow a]$ for some brand new constant $a$ (this rule is usually called Existential Elimination).
- But in orthodox modal logic there is no simple way of stripping off the diamonds.

# Hybrid deduction

- Hybrid deduction is based on a simple observation: it's easy to get at the information under the scope of diamonds — for there is an natural way of stripping the diamonds away.
- We shall explore this idea in the setting of tableau — but it can (and has been) used in a variety of proof styles, including resolution and natural deduction.
- Moreover, once the tableau system for reasoning about arbitrary models has been defined, it is straightforward to extend it to to cover the logics of special classes of models. That is, hybridization enables us to achieve the traditional modal goal of generality without resorting to Hilbert-systems.

# Moreover…

Hybrid reasoning is arguably quite natural.

In what follows we shall sometimes give an informal proof before we give the tableau proof. As we shall see, our tableau proofs mimic the informal reasoning fairly closely.

# Hip and cute

Consider the following statement:

> *If everyone you hate is hip, and someone you hate is cute, then someone you hate is both hip and cute.*

We can represent it as follows:

$$[\textsc{hate}]\,\text{hip} \wedge \langle\textsc{hate}\rangle\,\text{cute} \rightarrow \langle\textsc{hate}\rangle\,(\text{hip} \wedge \text{cute})$$

This is a valid statement, and it's validity is easy to establish informally...

# Informal argument

# Informal argument

- Suppose "If everyone you hate is hip, and someone you hate is cute, then someone you hate is both hip and cute" is not true.

# Informal argument

- Suppose "If everyone you hate is hip, and someone you hate is cute, then someone you hate is both hip and cute" is not true.
- Then everyone you hate is hip, and someone you hate is cute. However no one you hate is both hip and cute.

# Informal argument

- Suppose "If everyone you hate is hip, and someone you hate is cute, then someone you hate is both hip and cute" is not true.
- Then everyone you hate is hip, and someone you hate is cute. However no one you hate is both hip and cute.
- So there is someone that you hate (let's call him Jim) who is cute.

# Informal argument

- Suppose "If everyone you hate is hip, and someone you hate is cute, then someone you hate is both hip and cute" is not true.
- Then everyone you hate is hip, and someone you hate is cute. However no one you hate is both hip and cute.
- So there is someone that you hate (let's call him Jim) who is cute.
- But as Jim is someone you hate, he be hip as well as cute (for everyone you hate is hip).

# Informal argument

- Suppose "If everyone you hate is hip, and someone you hate is cute, then someone you hate is both hip and cute" is not true.
- Then everyone you hate is hip, and someone you hate is cute. However no one you hate is both hip and cute.
- So there is someone that you hate (let's call him Jim) who is cute.
- But as Jim is someone you hate, he be hip as well as cute (for everyone you hate is hip).
- But Jim can't be both hip and cute (for no one you hate is both hip and cute). Contradiction!. So the original statement was true after all.

$$[\textsc{hate}]\,\mathrm{hip} \wedge \langle\textsc{hate}\rangle\,\mathrm{cute} \rightarrow \langle\textsc{hate}\rangle\,(\mathrm{hip} \wedge \mathrm{cute})$$

$$[\text{HATE}]\,\text{hip} \wedge \langle\text{HATE}\rangle\,\text{cute} \rightarrow \langle\text{HATE}\rangle\,(\text{hip} \wedge \text{cute})$$

1 $\neg @_i([\text{HATE}]\,\text{h} \wedge \langle\text{HATE}\rangle\,\text{c} \rightarrow \langle\text{HATE}\rangle\,(\text{h} \wedge \text{c}))$

# $[\textsc{hate}]\,\text{hip} \wedge \langle\textsc{hate}\rangle\,\text{cute} \rightarrow \langle\textsc{hate}\rangle\,(\text{hip} \wedge \text{cute})$

| | |
|---|---|
| 1 | $\neg @_i([\textsc{hate}]\,\text{h} \wedge \langle\textsc{hate}\rangle\,\text{c} \rightarrow \langle\textsc{hate}\rangle\,(\text{h} \wedge \text{c}))$ |
| 2 | $@_i([\textsc{hate}]\,\text{h} \wedge \langle\textsc{hate}\rangle\,\text{c})$ |
| $2'$ | $\neg @_i\langle\textsc{hate}\rangle\,(\text{h} \wedge \text{c})$ |

## $[\textsc{hate}]\,\text{hip} \wedge \langle\textsc{hate}\rangle\,\text{cute} \rightarrow \langle\textsc{hate}\rangle\,(\text{hip} \wedge \text{cute})$

| | |
|---|---|
| 1 | $\neg @_i([\textsc{hate}]\,\text{h} \wedge \langle\textsc{hate}\rangle\,\text{c} \rightarrow \langle\textsc{hate}\rangle\,(\text{h} \wedge \text{c}))$ |
| 2 | $@_i([\textsc{hate}]\,\text{h} \wedge \langle\textsc{hate}\rangle\,\text{c})$ |
| $2'$ | $\neg @_i\langle\textsc{hate}\rangle\,(\text{h} \wedge \text{c})$ |
| 3 | $@_i[\textsc{hate}]\,\text{h}$ |
| $3'$ | $@_i\langle\textsc{hate}\rangle\,\text{c}$ |

# $[\textsc{hate}]\,\text{hip} \wedge \langle\textsc{hate}\rangle\,\text{cute} \rightarrow \langle\textsc{hate}\rangle\,(\text{hip} \wedge \text{cute})$

| | |
|---|---|
| 1 | $\neg @_i([\textsc{hate}]\,\text{h} \wedge \langle\textsc{hate}\rangle\,\text{c} \rightarrow \langle\textsc{hate}\rangle\,(\text{h} \wedge \text{c}))$ |
| 2 | $@_i([\textsc{hate}]\,\text{h} \wedge \langle\textsc{hate}\rangle\,\text{c})$ |
| 2′ | $\neg @_i\langle\textsc{hate}\rangle\,(\text{h} \wedge \text{c})$ |
| 3 | $@_i[\textsc{hate}]\,\text{h}$ |
| 3′ | $@_i\langle\textsc{hate}\rangle\,\text{c}$ |
| 4 | $@_i\langle\textsc{hate}\rangle\,j$ |
| 4′ | $@_j\text{c}$ |

## $[\textsc{hate}]\, \text{hip} \wedge \langle \textsc{hate} \rangle\, \text{cute} \rightarrow \langle \textsc{hate} \rangle\, (\text{hip} \wedge \text{cute})$

| | |
|---|---|
| 1 | $\neg @_i([\textsc{hate}]\, \text{h} \wedge \langle \textsc{hate} \rangle\, \text{c} \rightarrow \langle \textsc{hate} \rangle\, (\text{h} \wedge \text{c}))$ |
| 2 | $@_i([\textsc{hate}]\, \text{h} \wedge \langle \textsc{hate} \rangle\, \text{c})$ |
| 2′ | $\neg @_i \langle \textsc{hate} \rangle\, (\text{h} \wedge \text{c})$ |
| 3 | $@_i [\textsc{hate}]\, \text{h}$ |
| 3′ | $@_i \langle \textsc{hate} \rangle\, \text{c}$ |
| 4 | $@_i \langle \textsc{hate} \rangle\, j$ |
| 4′ | $@_j \text{c}$ |
| 5 | $@_j \text{h}$ |

# $[\textsc{hate}]\,\text{hip} \wedge \langle\textsc{hate}\rangle\,\text{cute} \rightarrow \langle\textsc{hate}\rangle\,(\text{hip} \wedge \text{cute})$

| | |
|---|---|
| 1 | $\neg @_i([\textsc{hate}]\,\text{h} \wedge \langle\textsc{hate}\rangle\,\text{c} \rightarrow \langle\textsc{hate}\rangle\,(\text{h} \wedge \text{c}))$ |
| 2 | $@_i([\textsc{hate}]\,\text{h} \wedge \langle\textsc{hate}\rangle\,\text{c})$ |
| 2′ | $\neg @_i\langle\textsc{hate}\rangle\,(\text{h} \wedge \text{c})$ |
| 3 | $@_i[\textsc{hate}]\,\text{h}$ |
| 3′ | $@_i\langle\textsc{hate}\rangle\,\text{c}$ |
| 4 | $@_i\langle\textsc{hate}\rangle\,j$ |
| 4′ | $@_j\text{c}$ |
| 5 | $@_j\text{h}$ |
| 6 | $\neg @_j(\text{h} \wedge \text{c})$ |

## $[\textsc{hate}]\,\text{hip} \wedge \langle\textsc{hate}\rangle\,\text{cute} \rightarrow \langle\textsc{hate}\rangle\,(\text{hip} \wedge \text{cute})$

| | | | |
|---|---|---|---|
| 1 | | $\neg @_i([\textsc{hate}]\,\text{h} \wedge \langle\textsc{hate}\rangle\,\text{c} \longrightarrow \langle\textsc{hate}\rangle\,(\text{h} \wedge \text{c}))$ | |
| 2 | | $@_i([\textsc{hate}]\,\text{h} \wedge \langle\textsc{hate}\rangle\,\text{c})$ | |
| 2′ | | $\neg @_i\langle\textsc{hate}\rangle\,(\text{h} \wedge \text{c})$ | |
| 3 | | $@_i[\textsc{hate}]\,\text{h}$ | |
| 3′ | | $@_i\langle\textsc{hate}\rangle\,\text{c}$ | |
| 4 | | $@_i\langle\textsc{hate}\rangle\,j$ | |
| 4′ | | $@_j\text{c}$ | |
| 5 | | $@_j\text{h}$ | |
| 6 | | $\neg @_j(\text{h} \wedge \text{c})$ | |
| 7 | $\neg @_j\text{h}$ | | $\neg @_j\text{c}$ |

# $[\textsc{hate}]\,\text{hip} \wedge \langle\textsc{hate}\rangle\,\text{cute} \rightarrow \langle\textsc{hate}\rangle\,(\text{hip} \wedge \text{cute})$

| | | |
|---|---|---|
| 1 | $\neg @_i([\textsc{hate}]\,\mathrm{h} \wedge \langle\textsc{hate}\rangle\,\mathrm{c} \rightarrow \langle\textsc{hate}\rangle\,(\mathrm{h} \wedge \mathrm{c}))$ | |
| 2 | $@_i([\textsc{hate}]\,\mathrm{h} \wedge \langle\textsc{hate}\rangle\,\mathrm{c})$ | |
| 2′ | $\neg @_i\langle\textsc{hate}\rangle\,(\mathrm{h} \wedge \mathrm{c})$ | |
| 3 | $@_i[\textsc{hate}]\,\mathrm{h}$ | |
| 3′ | $@_i\langle\textsc{hate}\rangle\,\mathrm{c}$ | |
| 4 | $@_i\langle\textsc{hate}\rangle\,j$ | |
| 4′ | $@_j\,\mathrm{c}$ | |
| 5 | $@_j\,\mathrm{h}$ | |
| 6 | $\neg @_j(\mathrm{h} \wedge \mathrm{c})$ | |
| 7 | $\neg @_j\,\mathrm{h}$ | $\neg @_j\,\mathrm{c}$ |
| | $\bot_{5,7}$ | |

# $[\text{HATE}]\,\text{hip} \wedge \langle\text{HATE}\rangle\,\text{cute} \rightarrow \langle\text{HATE}\rangle\,(\text{hip} \wedge \text{cute})$

| | | | |
|---|---|---|---|
| 1 | | $\neg @_i([\text{HATE}]\,h \wedge \langle\text{HATE}\rangle\,c \rightarrow \langle\text{HATE}\rangle\,(h \wedge c))$ | |
| 2 | | $@_i([\text{HATE}]\,h \wedge \langle\text{HATE}\rangle\,c)$ | |
| 2′ | | $\neg @_i\langle\text{HATE}\rangle\,(h \wedge c)$ | |
| 3 | | $@_i[\text{HATE}]\,h$ | |
| 3′ | | $@_i\langle\text{HATE}\rangle\,c$ | |
| 4 | | $@_i\langle\text{HATE}\rangle\,j$ | |
| 4′ | | $@_j c$ | |
| 5 | | $@_j h$ | |
| 6 | | $\neg @_j(h \wedge c)$ | |
| 7 | $\neg @_j h$ | | $\neg @_j c$ |
| | $\bot_{5,7}$ | | $\bot_{4',7}$ |

# Internalizing Labelled Deduction

| | | |
|---|---|---|
| $\neg$ rules | $\dfrac{@_i\neg\varphi}{\neg @_i\varphi}$ | $\dfrac{\neg @_i\neg\varphi}{@_i\varphi}$ |
| $\wedge$ rules | $\dfrac{@_i(\varphi\wedge\psi)}{\begin{array}{c}@_i\varphi\\ @_i\psi\end{array}}$ | $\dfrac{\neg @_i(\varphi\wedge\psi)}{\neg @_i\varphi \mid \neg @_i\psi}$ |
| $@$ rules | $\dfrac{@_i @_j\varphi}{@_j\varphi}$ | $\dfrac{\neg @_i @_j\varphi}{\neg @_j\varphi}$ |

# Extracting information from modal contexts

In the statement of these rules we write $j$ to indicate a nominal new to the branch where the rule is being applied.

# Extracting information from modal contexts

In the statement of these rules we write $j$ to indicate a nominal new to the branch where the rule is being applied.

$\Diamond$ rules

$$\frac{@_i\langle \mathrm{R}\rangle\,\varphi}{\begin{array}{l}@_i\langle \mathrm{R}\rangle\,j\\ @_j\varphi\end{array}}$$

$$\frac{\neg @_i\langle \mathrm{R}\rangle\,\varphi \qquad @_i\langle \mathrm{R}\rangle\,k}{\neg @_k\varphi}$$

# Extracting information from modal contexts

In the statement of these rules we write $j$ to indicate a nominal new to the branch where the rule is being applied.

$\Diamond$ rules

$$\frac{@_i\langle R\rangle\varphi}{\begin{array}{l}@_i\langle R\rangle j\\ @_j\varphi\end{array}}\qquad\qquad\frac{\neg @_i\langle R\rangle\varphi\quad @_i\langle R\rangle k}{\neg @_k\varphi}$$

$\Box$ rules

$$\frac{@_i[R]\varphi\quad @_i\langle R\rangle k}{@_k\varphi}\qquad\qquad\frac{\neg @_i[R]\varphi}{\begin{array}{l}@_i\langle R\rangle j\\ \neg @_j\varphi\end{array}}$$

# Link with first-order deduction (Studio Version)

# Link with first-order deduction (Studio Version)

- The hybrid rule from $@_i\Diamond\varphi$ conclude $@_i\Diamond j$ and $@_j\varphi$ is essentially the first-order rule of Existential Elimination (from $\exists x\varphi$ conclude $\varphi[x \leftarrow j]$).

# Link with first-order deduction (Studio Version)

- The hybrid rule from $@_i\Diamond\varphi$ conclude $@_i\Diamond j$ and $@_j\varphi$ is essentially the first-order rule of Existential Elimination (from $\exists x\varphi$ conclude $\varphi[x \leftarrow j]$).
- Recall that (via the Standard Translation) we know that $\Diamond\varphi$ is shorthand for $\exists y(Riy \wedge \textsc{st}_y(\varphi))$.

# Link with first-order deduction (Studio Version)

- The hybrid rule from $@_i\Diamond\varphi$ conclude $@_i\Diamond j$ and $@_j\varphi$ is essentially the first-order rule of Existential Elimination (from $\exists x\varphi$ conclude $\varphi[x \leftarrow j]$).
- Recall that (via the Standard Translation) we know that $\Diamond\varphi$ is shorthand for $\exists y(Riy \wedge \mathrm{ST}_y(\varphi))$.
- Applying Existential Elimination to this yields $Rij \wedge \mathrm{ST}_j(\varphi)$. But this is just $@_i\Diamond j \wedge @_j\varphi$, the output of the tableau rule.

# Link with first-order deduction (Studio Version)

- The hybrid rule from $@_i\Diamond\varphi$ conclude $@_i\Diamond j$ and $@_j\varphi$ is essentially the first-order rule of Existential Elimination (from $\exists x\varphi$ conclude $\varphi[x \leftarrow j]$).
- Recall that (via the Standard Translation) we know that $\Diamond\varphi$ is shorthand for $\exists y(Riy \wedge \textsc{st}_y(\varphi))$.
- Applying Existential Elimination to this yields $Rij \wedge \textsc{st}_j(\varphi)$. But this is just $@_i\Diamond j \wedge @_j\varphi$, the output of the tableau rule.
- In short, nominals give us exactly the grip we need on the bound variables hidden by modal notation. They give us the benefits of first-order techniques in a decidable logic.
- Incidentally, in Lecture 3 we'll be seeing yet another perspective on this rule.

# Link with first-order deduction (Live Version)

| Hybrid Logic | First Order Logic |
| --- | --- |
| $@_i \Diamond \phi$ | |

# Link with first-order deduction (Live Version)

| Hybrid Logic | First Order Logic |
| --- | --- |
| $@_i \Diamond \phi$ | $\exists y(Riy \wedge ST_y(\phi))$ |

## Link with first-order deduction (Live Version)

| Hybrid Logic | First Order Logic |
| --- | --- |
| $@_i \Diamond \phi$ | $\exists y(Riy \wedge ST_y(\phi))$ |
| $@_i \Diamond j$<br>$@_j \phi$ | |

# Link with first-order deduction (Live Version)

| Hybrid Logic | First Order Logic |
|---|---|
| $@_i\Diamond\phi$ | $\exists y(Riy \wedge ST_y(\phi))$ |
| | $Rij \wedge ST_j(\phi)$ |
| $@_i\Diamond j$ | |
| $@_j\phi$ | |

# Link with first-order deduction (Live Version)

| Hybrid Logic | First Order Logic |
|---|---|
| $@_i \Diamond \phi$ | $\exists y(Riy \wedge ST_y(\phi))$ |
| | $Rij \wedge ST_j(\phi)$ |
| $@_i \Diamond j$ | $Rij$ |
| $@_j \phi$ | $ST_j(\phi)$ |

# Equality rules

But more rules are needed. Why? Nothing we have said so far gets to grips with fact that nominals have an intrinsic logic. Nominals give us a modal theory of equality, and we need to get to deal with this. Here's one way of doing this:

$$\frac{(i \text{ occurs on branch})}{@_i i} \qquad \frac{@_i j \quad @_i \varphi}{@_j \varphi} \qquad \frac{@_i \Diamond j \quad @_j k}{@_i \Diamond k}$$

$$(\Diamond \mathrm{p} \wedge \Diamond \neg \mathrm{p}) \rightarrow (\Box(\mathrm{q} \rightarrow i) \rightarrow \Diamond \neg \mathrm{q})$$

$$(\Diamond \mathrm{p} \wedge \Diamond \neg \mathrm{p}) \rightarrow (\Box(\mathrm{q} \rightarrow i) \rightarrow \Diamond \neg \mathrm{q})$$

1 $\neg @_i((\Diamond \mathrm{p} \wedge \Diamond \neg \mathrm{p}) \rightarrow (\Box(\mathrm{q} \rightarrow i) \rightarrow \Diamond \neg \mathrm{q}))$

$$(\Diamond p \land \Diamond \neg p) \rightarrow (\Box(q \rightarrow i) \rightarrow \Diamond \neg q)$$

| | | |
|---|---|---|
| 1 | $\neg @_i((\Diamond p \land \Diamond \neg p) \rightarrow (\Box(q \rightarrow i) \rightarrow \Diamond \neg q))$ | |
| 2 | $@_i(\Diamond p \land \Diamond \neg p)$ | |
| 2′ | $\neg @_i(\Box(q \rightarrow i) \rightarrow \Diamond \neg q)$ | Propositional rule on 1 |

$$(\Diamond p \wedge \Diamond \neg p) \rightarrow (\Box(q \rightarrow i) \rightarrow \Diamond \neg q)$$

| | | |
|---|---|---|
| 1 | $\neg @_i((\Diamond p \wedge \Diamond \neg p) \rightarrow (\Box(q \rightarrow i) \rightarrow \Diamond \neg q))$ | |
| 2 | $@_i(\Diamond p \wedge \Diamond \neg p)$ | |
| 2′ | $\neg @_i(\Box(q \rightarrow i) \rightarrow \Diamond \neg q)$ | Propositional rule on 1 |
| 3 | $@_i \Diamond p$ | |
| 3′ | $@_i \Diamond \neg p$ | Propositional rule on 2 |

$$(\Diamond p \land \Diamond\neg p) \to (\Box(q \to i) \to \Diamond\neg q)$$

| | | |
|---|---|---|
| 1 | $\neg @_i((\Diamond p \land \Diamond\neg p) \to (\Box(q \to i) \to \Diamond\neg q))$ | |
| 2 | $@_i(\Diamond p \land \Diamond\neg p)$ | |
| 2′ | $\neg @_i(\Box(q \to i) \to \Diamond\neg q)$ | Propositional rule on 1 |
| 3 | $@_i \Diamond p$ | |
| 3′ | $@_i \Diamond\neg p$ | Propositional rule on 2 |
| 4 | $@_i \Diamond j$ | |
| 4′ | $@_j p$ | $\Diamond$ rule on 3 |

## $(\Diamond p \wedge \Diamond\neg p) \to (\Box(q \to i) \to \Diamond\neg q)$

| | | |
|---|---|---|
| 1 | $\neg @_i((\Diamond p \wedge \Diamond\neg p) \to (\Box(q \to i) \to \Diamond\neg q))$ | |
| 2 | $@_i(\Diamond p \wedge \Diamond\neg p)$ | |
| 2′ | $\neg @_i(\Box(q \to i) \to \Diamond\neg q)$ | Propositional rule on 1 |
| 3 | $@_i\Diamond p$ | |
| 3′ | $@_i\Diamond\neg p$ | Propositional rule on 2 |
| 4 | $@_i\Diamond j$ | |
| 4′ | $@_j p$ | $\Diamond$ rule on 3 |
| 5 | $@_i\Diamond k$ | |
| 5′ | $@_k\neg p$ | $\Diamond$ rule on 3’ |

## $(\Diamond p \wedge \Diamond\neg p) \to (\Box(q \to i) \to \Diamond\neg q)$

| | | |
|---|---|---|
| 1 | $\neg @_i((\Diamond p \wedge \Diamond\neg p) \to (\Box(q \to i) \to \Diamond\neg q))$ | |
| 2 | $@_i(\Diamond p \wedge \Diamond\neg p)$ | |
| $2'$ | $\neg @_i(\Box(q \to i) \to \Diamond\neg q)$ | Propositional rule on 1 |
| 3 | $@_i\Diamond p$ | |
| $3'$ | $@_i\Diamond\neg p$ | Propositional rule on 2 |
| 4 | $@_i\Diamond j$ | |
| $4'$ | $@_j p$ | $\Diamond$ rule on 3 |
| 5 | $@_i\Diamond k$ | |
| $5'$ | $@_k\neg p$ | $\Diamond$ rule on 3' |
| 6 | $@_i\Box(q \to i)$ | |
| $6'$ | $\neg @_i\Diamond\neg q$ | Propositional rule on 2' |

# The proof continued...

| | |
|---|---|
| 4 | $@_i \Diamond j$ |
| 4′ | $@_j \mathrm{p}$ |
| 5 | $@_i \Diamond k$ |
| 5′ | $@_k \neg \mathrm{p}$ |
| 6 | $@_i \Box(\mathrm{q} \rightarrow i)$ |
| 6′ | $\neg @_i \Diamond \neg \mathrm{q}$ |

# The proof continued...

| | | |
|---|---|---|
| 4 | $@_i\Diamond j$ | |
| 4′ | $@_j p$ | |
| 5 | $@_i\Diamond k$ | |
| 5′ | $@_k\neg p$ | |
| 6 | $@_i\Box(q \rightarrow i)$ | |
| 6′ | $\neg @_i\Diamond\neg q$ | |
| 7 | $@_j q$ | $\neg\Diamond$ rule on 4 and 6', then $\neg @$ rule |

# The proof continued…

| | | |
|---|---|---|
| 4 | $@_i\Diamond j$ | |
| 4′ | $@_j p$ | |
| 5 | $@_i\Diamond k$ | |
| 5′ | $@_k\neg p$ | |
| 6 | $@_i\Box(q \to i)$ | |
| 6′ | $\neg @_i\Diamond\neg q$ | |
| 7 | $@_j q$ | $\neg\Diamond$ rule on 4 and 6', then $\neg @$ rule |
| 8 | $@_j(q \to i)$ | $\Box$ rule on 4 and 6 |

# The proof continued...

| | | | | |
|---|---|---|---|---|
| 4 | | $@_i \Diamond j$ | | |
| 4′ | | $@_j p$ | | |
| 5 | | $@_i \Diamond k$ | | |
| 5′ | | $@_k \neg p$ | | |
| 6 | | $@_i \Box (q \rightarrow i)$ | | |
| 6′ | | $\neg @_i \Diamond \neg q$ | | |
| 7 | | $@_j q$ | | $\neg\Diamond$ rule on 4 and 6', then $\neg @$ rule |
| 8 | | $@_j (q \rightarrow i)$ | | $\Box$ rule on 4 and 6 |
| 9 | $\neg @_j q$ | | $@_j i$ | Propositional rule on 7 and 8 |
| | $\perp_{7,9}$ | | | |

# The proof continued. . .

| | | | | |
|---|---|---|---|---|
| 4 | | $@_i\Diamond j$ | | |
| 4′ | | $@_j \mathrm{p}$ | | |
| 5 | | $@_i\Diamond k$ | | |
| 5′ | | $@_k\neg \mathrm{p}$ | | |
| 6 | | $@_i\Box(\mathrm{q}\rightarrow i)$ | | |
| 6′ | | $\neg @_i\Diamond\neg \mathrm{q}$ | | |
| 7 | | $@_j \mathrm{q}$ | | $\neg\Diamond$ rule on 4 and 6', then $\neg@$ rule |
| 8 | | $@_j(\mathrm{q}\rightarrow i)$ | | $\Box$ rule on 4 and 6 |
| 9 | $\neg @_j \mathrm{q}$ | | $@_j i$ | Propositional rule on 7 and 8 |
| | $\perp_{7,9}$ | | | |

# The proof continued. . .

| | |
|---|---|
| 4 | $@_i\Diamond j$ |
| 4′ | $@_j \mathrm{p}$ |
| 5 | $@_i\Diamond k$ |
| 5′ | $@_k\neg \mathrm{p}$ |
| 6 | $@_i\Box(\mathrm{q} \rightarrow i)$ |
| 6′ | $\neg @_i\Diamond\neg \mathrm{q}$ |
| 9 | $@_j i$ |

# The proof continued...

| | | |
|---|---|---|
| 4 | $@_i\Diamond j$ | |
| 4′ | $@_j$p | |
| 5 | $@_i\Diamond k$ | |
| 5′ | $@_k\neg$p | |
| 6 | $@_i\Box(\mathrm{q} \rightarrow i)$ | |
| 6′ | $\neg @_i\Diamond\neg$q | |
| 9 | $@_j i$ | |
| 10 | $@_k$q | $\neg\Diamond$ rule on 5 and 6', then $\neg@$ rule |

# The proof continued...

| | | |
|---|---|---|
| 4 | $@_i \Diamond j$ | |
| 4′ | $@_j \mathrm{p}$ | |
| 5 | $@_i \Diamond k$ | |
| 5′ | $@_k \neg \mathrm{p}$ | |
| 6 | $@_i \Box(\mathrm{q} \rightarrow i)$ | |
| 6′ | $\neg @_i \Diamond \neg \mathrm{q}$ | |
| 9 | $@_j i$ | |
| 10 | $@_k \mathrm{q}$ | $\neg\Diamond$ rule on 5 and 6’, then $\neg @$ rule |
| 11 | $@_k(\mathrm{q} \rightarrow i)$ | $\Box$ rule on 5 and 6 |

# The proof continued...

| | | |
|---|---|---|
| 4 | $@_i \Diamond j$ | |
| 4′ | $@_j \mathrm{p}$ | |
| 5 | $@_i \Diamond k$ | |
| 5′ | $@_k \neg \mathrm{p}$ | |
| 6 | $@_i \Box(\mathrm{q} \rightarrow i)$ | |
| 6′ | $\neg @_i \Diamond \neg \mathrm{q}$ | |
| 9 | $@_j i$ | |
| 10 | $@_k \mathrm{q}$ | $\neg\Diamond$ rule on 5 and 6’, then $\neg @$ rule |
| 11 | $@_k(\mathrm{q} \rightarrow i)$ | $\Box$ rule on 5 and 6 |
| 12 | $@_k i$ | Modus Ponens on 10 and 11 |

# The proof continued...

| | | |
|---|---|---|
| 4 | $@_i\Diamond j$ | |
| 4′ | $@_j \mathrm{p}$ | |
| 5 | $@_i\Diamond k$ | |
| 5′ | $@_k\neg \mathrm{p}$ | |
| 6 | $@_i\Box(\mathrm{q} \rightarrow i)$ | |
| 6′ | $\neg @_i\Diamond\neg \mathrm{q}$ | |
| 9 | $@_j i$ | |
| 10 | $@_k \mathrm{q}$ | $\neg\Diamond$ rule on 5 and 6', then $\neg @$ rule |
| 11 | $@_k(\mathrm{q} \rightarrow i)$ | $\Box$ rule on 5 and 6 |
| 12 | $@_k i$ | Modus Ponens on 10 and 11 |
| 13 | $@_i \mathrm{p}$ | Nom on 4' and 9 |

# The proof continued...

| | | |
|---|---|---|
| 4 | $@_i \Diamond j$ | |
| 4′ | $@_j \mathrm{p}$ | |
| 5 | $@_i \Diamond k$ | |
| 5′ | $@_k \neg \mathrm{p}$ | |
| 6 | $@_i \Box(\mathrm{q} \rightarrow i)$ | |
| 6′ | $\neg @_i \Diamond \neg \mathrm{q}$ | |
| 9 | $@_j i$ | |
| 10 | $@_k \mathrm{q}$ | $\neg\Diamond$ rule on 5 and 6', then $\neg @$ rule |
| 11 | $@_k(\mathrm{q} \rightarrow i)$ | $\Box$ rule on 5 and 6 |
| 12 | $@_k i$ | Modus Ponens on 10 and 11 |
| 13 | $@_i \mathrm{p}$ | Nom on 4' and 9 |
| 14 | $@_i \neg \mathrm{p}$ | Nom on 5' and 12 |

# The proof continued...

| | | |
|---|---|---|
| 4 | $@_i\Diamond j$ | |
| 4′ | $@_j p$ | |
| 5 | $@_i\Diamond k$ | |
| 5′ | $@_k\neg p$ | |
| 6 | $@_i\Box(q \rightarrow i)$ | |
| 6′ | $\neg @_i\Diamond\neg q$ | |
| 9 | $@_j i$ | |
| 10 | $@_k q$ | $\neg\Diamond$ rule on 5 and 6', then $\neg @$ rule |
| 11 | $@_k(q \rightarrow i)$ | $\Box$ rule on 5 and 6 |
| 12 | $@_k i$ | Modus Ponens on 10 and 11 |
| 13 | $@_i p$ | Nom on 4' and 9 |
| 14 | $@_i\neg p$ | Nom on 5' and 12 |
| 15 | $\neg @_i p$ | $\neg$ rule on 14 — contradiction! |

# Reasoning over other classes of models

- Our tableau system deals (correctly and completely) with reasoning over arbitrary models, that is, models where we have made no special assumptions about the underlying relations. For some applications this is sufficient.
- But (as we said at the start of the lecture) in many applications we are interested in models where the relations interpreting the modalities have special properties, such as symmetry, transitivity, irreflexivity, density, discreteness, antisymmetry, determinism, and so on. We need to find a way of coping with such frame conditions in hybrid logic.
- Our basic tableau system cannot handle such requirements — but it can be easily extended to cope with them, thus meeting the traditional modal goal of generality. We'll look at two examples.

# Nice neighbours

Consider the following statement:

> *If you have a neighbour who only has nice neighbours, then you are nice.*

We can represent it as follows:

$$\langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice} \rightarrow \text{nice}$$

This is true no matter how the adjective "nice" is interpreted. Its truth hinges on the fact that neighbourhood is a symmetric relation.

# Informal Argument

# Informal Argument

- Suppose $\langle \textsc{neighbour} \rangle [\textsc{neighbour}]\, \text{nice} \rightarrow \text{nice}$ is false of you.

# Informal Argument

- Suppose ⟨NEIGHBOUR⟩ [NEIGHBOUR] nice → nice is false of you.
- Then ⟨NEIGHBOUR⟩ [NEIGHBOUR] nice is true of you, but nice is false of you (that is, you are not nice).

# Informal Argument

- Suppose ⟨NEIGHBOUR⟩ [NEIGHBOUR] nice → nice is false of you.
- Then ⟨NEIGHBOUR⟩ [NEIGHBOUR] nice is true of you, but nice is false of you (that is, you are not nice).
- Then you have a neighbour (let's call him Joe) who only has nice neighbours (that is, [NEIGHBOUR] nice is true of Joe).

# Informal Argument

- Suppose $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice} \rightarrow \text{nice}$ is false of you.
- Then $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice}$ is true of you, but nice is false of you (that is, you are not nice).
- Then you have a neighbour (let's call him Joe) who only has nice neighbours (that is, $[\text{NEIGHBOUR}]\, \text{nice}$ is true of Joe).
- But neighbourhood is a symmetric relation, hence you are one of Joe's neighbours.

# Informal Argument

- Suppose $\langle \textsc{neighbour} \rangle [\textsc{neighbour}]\, \text{nice} \rightarrow \text{nice}$ is false of you.
- Then $\langle \textsc{neighbour} \rangle [\textsc{neighbour}]\, \text{nice}$ is true of you, but nice is false of you (that is, you are not nice).
- Then you have a neighbour (let's call him Joe) who only has nice neighbours (that is, $[\textsc{neighbour}]\, \text{nice}$ is true of Joe).
- But neighbourhood is a symmetric relation, hence you are one of Joe's neighbours.
- But all Joe's neighbours are nice — so you must be nice too. Contradiction!
- So $\langle \textsc{neighbour} \rangle [\textsc{neighbour}]\, \text{nice} \rightarrow \text{nice}$ must true of you after all.

# Informal Argument

- Suppose $\langle \textsc{neighbour} \rangle [\textsc{neighbour}]\,\text{nice} \rightarrow \text{nice}$ is false of you.
- Then $\langle \textsc{neighbour} \rangle [\textsc{neighbour}]\,\text{nice}$ is true of you, but nice is false of you (that is, you are not nice).
- Then you have a neighbour (let's call him Joe) who only has nice neighbours (that is, $[\textsc{neighbour}]\,\text{nice}$ is true of Joe).
- But neighbourhood is a symmetric relation, hence you are one of Joe's neighbours.
- But all Joe's neighbours are nice — so you must be nice too. Contradiction!
- So $\langle \textsc{neighbour} \rangle [\textsc{neighbour}]\,\text{nice} \rightarrow \text{nice}$ must true of you after all.

But can we mimic this argument using our existing tableau system? Let's try...

$\langle$NEIGHBOUR$\rangle$ [NEIGHBOUR] nice $\rightarrow$ nice

# $\langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice} \rightarrow \text{nice}$

1 $@_i(\langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice} \rightarrow \text{nice})$

# $\langle \textsc{neighbour} \rangle\, [\textsc{neighbour}]\, \text{nice} \rightarrow \text{nice}$

| | | |
|---|---|---|
| 1 | $@_i(\langle \textsc{neighbour} \rangle\, [\textsc{neighbour}]\, \text{nice} \rightarrow \text{nice})$ | |
| 2 | $@_i\langle \textsc{neighbour} \rangle\, [\textsc{neighbour}]\, \text{nice}$ | |
| 2′ | $\neg @_i\text{nice}$ | Propositional rule on 1 |

# $\langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice} \rightarrow \text{nice}$

| | | |
|---|---|---|
| 1 | $@_i(\langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice} \rightarrow \text{nice})$ | |
| 2 | $@_i \langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice}$ | |
| 2′ | $\neg @_i \text{nice}$ | Propositional rule on 1 |
| 3 | $@_i \langle \textsc{neighbour} \rangle \, j$ | |
| 3′ | $@_j [\textsc{neighbour}] \, \text{nice}$ | $\Diamond$ rule on 2 |

# $\langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice} \rightarrow \text{nice}$

| | | |
|---|---|---|
| 1 | $@_i(\langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice} \rightarrow \text{nice})$ | |
| 2 | $@_i \langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice}$ | |
| 2′ | $\neg @_i \text{nice}$ | Propositional rule on 1 |
| 3 | $@_i \langle \textsc{neighbour} \rangle \, j$ | |
| 3′ | $@_j [\textsc{neighbour}] \, \text{nice}$ | $\Diamond$ rule on 2 |

Now we are blocked. There is no way to close this branch.

# But there is an easy solution

Add the following rule when working with symmetric relations:

$$\frac{@_i \langle \textsc{neighbour} \rangle\, j}{@_j \langle \textsc{neighbour} \rangle\, i}$$

(Here $i$ and $j$ can be any nominals on the branch we are working on).
This rule is a direct expression of symmetry, and with its help we can finish off our proof.

## $\langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice} \rightarrow \text{nice}$

| | | |
|---|---|---|
| 1 | $@_i(\langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice} \rightarrow \text{nice})$ | |
| 2 | $@_i \langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice}$ | |
| 2′ | $\neg @_i \text{nice}$ | Propositional rule on 1 |
| 3 | $@_i \langle \textsc{neighbour} \rangle \, j$ | |
| 3′ | $@_j [\textsc{neighbour}] \, \text{nice}$ | $\Diamond$ rule on 2 |

# $\langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice} \rightarrow \text{nice}$

| | | |
|---|---|---|
| 1 | $@_i(\langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice} \rightarrow \text{nice})$ | |
| 2 | $@_i \langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice}$ | |
| 2′ | $\neg @_i \text{nice}$ | Propositional rule on 1 |
| 3 | $@_i \langle \textsc{neighbour} \rangle \, j$ | |
| 3′ | $@_j [\textsc{neighbour}] \, \text{nice}$ | $\Diamond$ rule on 2 |
| 4 | $@_j \langle \textsc{neighbour} \rangle \, i$ | Symmetry rule on 3 |

# $\langle \textsc{neighbour} \rangle\, [\textsc{neighbour}]\, \text{nice} \rightarrow \text{nice}$

| | | |
|---|---|---|
| 1 | $@_i(\langle \textsc{neighbour} \rangle\, [\textsc{neighbour}]\, \text{nice} \rightarrow \text{nice})$ | |
| 2 | $@_i \langle \textsc{neighbour} \rangle\, [\textsc{neighbour}]\, \text{nice}$ | |
| $2'$ | $\neg @_i \text{nice}$ | Propositional rule on 1 |
| 3 | $@_i \langle \textsc{neighbour} \rangle\, j$ | |
| $3'$ | $@_j [\textsc{neighbour}]\, \text{nice}$ | $\Diamond$ rule on 2 |
| 4 | $@_j \langle \textsc{neighbour} \rangle\, i$ | Symmetry rule on 3 |
| 5 | $@_i \text{nice}$ | $\Box$ rule on 3' and 4 |

# $\langle \text{NEIGHBOUR} \rangle [\text{NEIGHBOUR}]\, \text{nice} \rightarrow \text{nice}$

| | | |
|---|---|---|
| 1 | $@_i(\langle \text{NEIGHBOUR} \rangle [\text{NEIGHBOUR}]\, \text{nice} \rightarrow \text{nice})$ | |
| 2 | $@_i \langle \text{NEIGHBOUR} \rangle [\text{NEIGHBOUR}]\, \text{nice}$ | |
| $2'$ | $\neg @_i \text{nice}$ | Propositional rule on 1 |
| 3 | $@_i \langle \text{NEIGHBOUR} \rangle\, j$ | |
| $3'$ | $@_j [\text{NEIGHBOUR}]\, \text{nice}$ | $\Diamond$ rule on 2 |
| 4 | $@_j \langle \text{NEIGHBOUR} \rangle\, i$ | Symmetry rule on 3 |
| 5 | $@_i \text{nice}$ | $\Box$ rule on 3' and 4 |
| | $\bot_{2',5}$ | |

# $\langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice} \rightarrow \text{nice}$

| | | |
|---|---|---|
| 1 | $@_i(\langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice} \rightarrow \text{nice})$ | |
| 2 | $@_i \langle \textsc{neighbour} \rangle \, [\textsc{neighbour}] \, \text{nice}$ | |
| $2'$ | $\neg @_i \text{nice}$ | Propositional rule on 1 |
| 3 | $@_i \langle \textsc{neighbour} \rangle \, j$ | |
| $3'$ | $@_j [\textsc{neighbour}] \, \text{nice}$ | $\Diamond$ rule on 2 |
| 4 | $@_j \langle \textsc{neighbour} \rangle \, i$ | Symmetry rule on 3 |
| 5 | $@_i \text{nice}$ | $\Box$ rule on 3' and 4 |
| | $\perp_{2',5}$ | |

# Loop-free time

Consider the following statement:

> *If time i precedes time j, then time j does not precede time i.*

We can represent the statement as follows (where $\langle \mathrm{F} \rangle$ is a diamond meaning "sometime-in-the-future"):

$$@_i \langle \mathrm{F} \rangle\, j \rightarrow \neg @_j \langle \mathrm{F} \rangle\, i$$

If you accept that temporal precedence is both transitive and irreflexive (the usual assumption) then this is a valid statement.

# Informal Argument

# Informal Argument

- Suppose that “if i precedes time j, then time j does not precede time i” is false.

# Informal Argument

- Suppose that “if i precedes time j, then time j does not precede time i” is false.
- Then time i precedes time j, but time j precedes time i too.

# Informal Argument

- Suppose that "if i precedes time j, then time j does not precede time i" is false.
- Then time i precedes time j, but time j precedes time i too.
- But temporal precedence is transitive, so time i precedes time i.

# Informal Argument

- Suppose that "if i precedes time j, then time j does not precede time i" is false.
- Then time i precedes time j, but time j precedes time i too.
- But temporal precedence is transitive, so time i precedes time i.
- But temporal precedence is irreflexive, so time i cannot precede time i.

# Informal Argument

- Suppose that "if i precedes time j, then time j does not precede time i" is false.
- Then time i precedes time j, but time j precedes time i too.
- But temporal precedence is transitive, so time i precedes time i.
- But temporal precedence is irreflexive, so time i cannot precede time i.
- From this contradiction we conclude that our original statement was true after all.

But can we prove $@_i \langle \mathrm{F} \rangle j \rightarrow \neg @_j \langle \mathrm{F} \rangle i$ using our existing tableau system? Let's try…

## But can we prove $@_i \langle \mathrm{F} \rangle j \rightarrow \neg @_j \langle \mathrm{F} \rangle i$ using our existing tableau system? Let's try…

1 $\neg @_k (@_i \langle \mathrm{F} \rangle j \rightarrow \neg @_j \langle \mathrm{F} \rangle i)$

## But can we prove $@_i \langle \mathrm{F} \rangle\, j \rightarrow \neg @_j \langle \mathrm{F} \rangle\, i$ using our existing tableau system? Let's try…

| | | |
|---|---|---|
| 1 | $\neg @_k(@_i \langle \mathrm{F} \rangle\, j \rightarrow \neg @_j \langle \mathrm{F} \rangle\, i)$ | |
| 2 | $@_k @_i \langle \mathrm{F} \rangle\, j$ | |
| 2′ | $\neg @_k \neg @_j \langle \mathrm{F} \rangle\, i)$ | Propositional rule on 1 |

# But can we prove $@_i \langle \mathrm{F} \rangle j \rightarrow \neg @_j \langle \mathrm{F} \rangle i$ using our existing tableau system? Let's try...

| | | |
|---|---|---|
| 1 | $\neg @_k (@_i \langle \mathrm{F} \rangle j \rightarrow \neg @_j \langle \mathrm{F} \rangle i)$ | |
| 2 | $@_k @_i \langle \mathrm{F} \rangle j$ | |
| 2′ | $\neg @_k \neg @_j \langle \mathrm{F} \rangle i)$ | Propositional rule on 1 |
| 3 | $@_i \langle \mathrm{F} \rangle j$ | @ rule on 2 |

# But can we prove $@_i\langle \mathrm{F}\rangle j \rightarrow \neg @_j\langle \mathrm{F}\rangle i$ using our existing tableau system? Let's try...

| | | |
|---|---|---|
| 1 | $\neg @_k(@_i\langle \mathrm{F}\rangle j \rightarrow \neg @_j\langle \mathrm{F}\rangle i)$ | |
| 2 | $@_k @_i\langle \mathrm{F}\rangle j$ | |
| 2′ | $\neg @_k \neg @_j\langle \mathrm{F}\rangle i)$ | Propositional rule on 1 |
| 3 | $@_i\langle \mathrm{F}\rangle j$ | @ rule on 2 |
| 4 | $@_j\langle \mathrm{F}\rangle i$ | $\neg @ \neg$ rule on 2’ |

## But can we prove $@_i \langle F \rangle j \rightarrow \neg @_j \langle F \rangle i$ using our existing tableau system? Let's try…

| | | |
|---|---|---|
| 1 | $\neg @_k (@_i \langle F \rangle j \rightarrow \neg @_j \langle F \rangle i)$ | |
| 2 | $@_k @_i \langle F \rangle j$ | |
| $2'$ | $\neg @_k \neg @_j \langle F \rangle i)$ | Propositional rule on 1 |
| 3 | $@_i \langle F \rangle j$ | @ rule on 2 |
| 4 | $@_j \langle F \rangle i$ | $\neg @ \neg$ rule on 2' |

Now we are blocked. There is no way to close this branch.

# But there is an easy solution

Add the following rules when working with irreflexive and transitive relations:

$$\frac{}{@_i \neg \langle F \rangle\, i} \qquad \frac{@_i \langle F \rangle\, j \quad @_j \langle F \rangle\, k}{@_i \langle F \rangle\, k}$$

(Here $i$, $j$ and $k$ can be any nominals on the branch we are working on).
These rules are a direct expression of irreflexivity and transitivity, and with their help we can finish off our proof.

## $@_i \langle F \rangle j \rightarrow \neg @_j \langle F \rangle i$

| | | |
|---|---|---|
| 1 | $\neg @_k (@_i \langle F \rangle j \rightarrow \neg @_j \langle F \rangle i)$ | |
| 2 | $@_k @_i \langle F \rangle j$ | |
| $2'$ | $\neg @_k \neg @_j \langle F \rangle i)$ | Propositional rule on 1 |
| 3 | $@_i \langle F \rangle j$ | @ rule on 2 |
| 4 | $@_j \langle F \rangle i$ | ¬@¬ rule on 2' |

## $@_i \langle F \rangle j \rightarrow \neg @_j \langle F \rangle i$

| | | |
|---|---|---|
| 1 | $\neg @_k (@_i \langle F \rangle j \rightarrow \neg @_j \langle F \rangle i)$ | |
| 2 | $@_k @_i \langle F \rangle j$ | |
| $2'$ | $\neg @_k \neg @_j \langle F \rangle i)$ | Propositional rule on 1 |
| 3 | $@_i \langle F \rangle j$ | @ rule on 2 |
| 4 | $@_j \langle F \rangle i$ | ¬@¬ rule on 2' |
| 5 | $@_i \langle F \rangle i$ | Transitivity rule on 3 and 4 |

## $@_i \langle F \rangle j \rightarrow \neg @_j \langle F \rangle i$

| | | |
|---|---|---|
| 1 | $\neg @_k (@_i \langle F \rangle j \rightarrow \neg @_j \langle F \rangle i)$ | |
| 2 | $@_k @_i \langle F \rangle j$ | |
| $2'$ | $\neg @_k \neg @_j \langle F \rangle i)$ | Propositional rule on 1 |
| 3 | $@_i \langle F \rangle j$ | @ rule on 2 |
| 4 | $@_j \langle F \rangle i$ | ¬@¬ rule on 2' |
| 5 | $@_i \langle F \rangle i$ | Transitivity rule on 3 and 4 |
| 6 | $\neg @_i \langle F \rangle i$ | Irreflexivity rule |

# $@_i \langle F \rangle j \rightarrow \neg @_j \langle F \rangle i$

| | | |
|---|---|---|
| 1 | $\neg @_k (@_i \langle F \rangle j \rightarrow \neg @_j \langle F \rangle i)$ | |
| 2 | $@_k @_i \langle F \rangle j$ | |
| $2'$ | $\neg @_k \neg @_j \langle F \rangle i)$ | Propositional rule on 1 |
| 3 | $@_i \langle F \rangle j$ | @ rule on 2 |
| 4 | $@_j \langle F \rangle i$ | ¬@¬ rule on 2' |
| 5 | $@_i \langle F \rangle i$ | Transitivity rule on 3 and 4 |
| 6 | $\neg @_i \langle F \rangle i$ | Irreflexivity rule |
| | $\bot_{5,6}$ | |

# Pure formulas

- It's time to be more precise about what completeness results are possible here.
- To do this we need to think about pure formulas.
- A formula of the basic hybrid language is pure if it contains no propositional variables. That is, the only atoms in pure formulas are nominals (and $\top$ and $\bot$ if we have them in the language).
- We'll first discuss what we can say about frames using pure formulas, and then we'll state a general result about how they can help us in hybrid deduction.

# Frame definability (I)

A formula defines a class of frames if it is valid on precisely the frames belonging to that class class. We can define many important classes of frames using pure formulas:

| | |
|---|---|
| $@_i \Diamond i$ | *Reflexivity* |
| $@_i \Diamond j \rightarrow @_j \Diamond i$ | *Symmetry* |
| $@_i \Diamond j \wedge @_j \Diamond k \rightarrow @_i \Diamond k$ | *Transitivity* |

# Frame definability (II)

These previous three examples are also definable using orthodox modal language. But pure formulas can also define frame classes which are not definable in orthodox modal logic:

| | |
|---|---|
| $@_i \neg \Diamond i$ | *Irreflexivity* |
| $@_i \Diamond j \rightarrow @_j \neg \Diamond i$ | *Asymmetry* |
| $@_i \Box (\Diamond i \rightarrow i)$ | *Antisymmetry* |
| $@_j \Diamond i \vee @_j i \vee @_i \Diamond j$ | *Trichotomy* |

# From formulas to tableau rules

Let $@_i\varphi$ be a pure formula, built out of nominals $i, i_1, \ldots, i_n$. Then the simplest (though not always the smartest!) way of turning this formula into a tableau rule is as follows:

$$\frac{(j, j_1, \ldots, j_n \text{ on branch})}{@_i\varphi[i \leftarrow j, i_1 \leftarrow j_1, \ldots, i_n \leftarrow j_n]}$$

This rule simply says: for any branch $B$ of the tableau you are building, you are free to instantiate $@_i\varphi$ with nominals occurring on $B$ and add the resulting formula to the end of $B$.

# Frame definability and deduction match for pure formulas

**Completeness Theorem** Suppose you extend the basic tableau system with the tableau rules for the pure formulas $@_j\varphi, \ldots, @_k\psi$ (that is, the rules of the form just described). Then the resulting system is (sound and) complete with respect to the class of frames defined by these formulas.

That is, the frame-defining and deductive powers of pure formulas match perfectly for pure formulas.

Two comments should be made about this result…

# We can use any pure formula

At first glance, it seems that this completeness result only covers pure formulas of the form $@_i\varphi$. But many interesting pure formulas are not of this form. For example symmetry: $@_i\Diamond j \rightarrow @_j\Diamond i$.

Note, however, that for any pure formula $\varphi$, and any nominals $i$, $\varphi$ and $@_i\varphi$ define exactly the same class of frames.

For example symmetry can be defined by $@_k(@_i\Diamond j \rightarrow @_j\Diamond i)$.

So our completeness theorem is fully general: it covers all classes of frames definable by a pure formulas.

# But we can often be smarter

Suppose we want a complete system for symmetry. We could do this by adding the rule suggested by the previous system:

$$\frac{}{@_k(@_i\Diamond j \rightarrow @_j\Diamond i)}.$$

But in the nice neighbours example we used the following rule instead:

$$\frac{@_i\Diamond j}{@_j\Diamond i}$$

This rule is smarter: it saves us having to use tableau rules to get rid of the outermost $@_k$, and then break down the implication.

# Slightly more generally

Given a pure formula of the form

$$(@_i\varphi_1 \wedge \cdots \wedge @_j\varphi_n) \rightarrow (@_k\varphi_{n+1} \vee \cdots \vee @_l\varphi_{n+m})$$

we can turn it into the tableau rule

$$\frac{@_i\varphi_1, \ldots, @_j\varphi_n}{@_k\varphi_{n+1} \mid \cdots \mid @_l\varphi_{n+m}}$$

without losing completeness.

# Further themes in hybrid deduction

To conclude, let's briefly address the following questions:

- Why are general completeness proof easy to come by in hybrid logic?
- Can we really adapt these ideas to other proof styles?
- Is any of this stuff implementable?

# Why are general completeness proofs so easy to come by in hybrid logic?

# Why are general completeness proofs so easy to come by in hybrid logic?

- Essentially because the basic hybrid logic enables us to use first-order techniques to build models.

# Why are general completeness proofs so easy to come by in hybrid logic?

- Essentially because the basic hybrid logic enables us to use first-order techniques to build models.
- For example, when proving completeness for hybrid Hilbert systems, it's not necessary to use modal-style canonical models — you can build what are basically first-order Henkin models. We'll be talking about this in Lecture 3.

# Why are general completeness proofs so easy to come by in hybrid logic?

- Essentially because the basic hybrid logic enables us to use first-order techniques to build models.
- For example, when proving completeness for hybrid Hilbert systems, it's not necessary to use modal-style canonical models — you can build what are basically first-order Henkin models. We'll be talking about this in Lecture 3.
- And for tableau completeness proofs, observe that the tableau rules crunch formulas down into expressions of the form $(\neg)@_i\mathrm{p}$, $(\neg)@_i j$ and $(\neg)@_i\Diamond j$. Open branches are thus Robinson diagrams of satisfying models.

# Named models are important

- Moreover, the models we build in this way are named. (A named model is a model in which every point is named by some nominal.)
- A model theoretic argument (discussed in Lecture 3) shows that if all instances of a pure formula $\varphi$ are true at all states in a named model, then the underlying frame validates $\phi$. This gives us completeness for any extension by pure axioms.

# Can we really adapt these ideas to other proof styles?

Yes. The key insight is that the combination of nominals and @ allow us to extract information from behind the scope of diamonds.

This idea has been successfully applied to define general sequent calculi (Seligman), natural deduction systems (Seligman, Brauner), and resolution calculi (Areces). It's also been applied with partial success to define display calculi (Demri and Goré).

Let's take a quick look at the way Torben Brauner handles natural deduction in hybrid logic.

# Some basic natural deduction rules

$$\dfrac{\begin{matrix}[@_i\varphi] \\ \vdots \\ @_i\psi\end{matrix}}{@_i(\varphi \rightarrow \psi)}\;(\rightarrow I)$$

$$\dfrac{@_i(\varphi \rightarrow \psi) \qquad @_i\varphi}{@_i\psi}\;(\rightarrow E)$$

$$\dfrac{@_i\varphi}{@_k@_i\varphi}\;(@I)$$

$$\dfrac{@_k@_i\varphi}{@_i\varphi}\;(@E)$$

# Natural deduction rules for modalities

$$\dfrac{\begin{array}{c}[@_i\Diamond j]\\ \vdots \\ @_j\varphi\end{array}}{@_i\Box\varphi}\ (\Box I)^*$$

$$\dfrac{@_i\Box\varphi \qquad @_i\Diamond k}{@_k\varphi}\ (\Box E)$$

$*$ $j$ does not occur in $@_i\Box\varphi$ or in any undischarged assumptions other than the specified occurrences of $@_i\Diamond j$.

# An example: $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$

$$
\dfrac{
  \dfrac{
    \dfrac{[@_i\Box(\varphi \to \psi)]^3 \quad [@_i\Diamond j]^1}{@_j(\varphi \to \psi)}\,(\Box E)
    \qquad
    \dfrac{[@_i\Box\varphi]^2 \quad [@_i\Diamond j]^1}{@_j\varphi}\,(\Box E)
  }{
    \dfrac{\dfrac{@_j\psi}{@_i\Box\psi}\,(\Box I)^1}{@_i(\Box\varphi \to \Box\psi)}\,(\to I)^2
  }\,(\to E)
}{@_i(\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi))}\,(\to I)^3
$$

# Is any of this stuff implementable?

Yes — but we need to be careful. For example, the equality rules discussed today are nice for hand calculation, but naive computationally.

The **HTab** system (Areces and Hoffmann) implements more sophisticated rules (due to Bolander and Blackburn) which guarantee termination. The system is optimised in several ways, and although a recent system, is already a competitive prover.

# And then there's resolution

A significant development is the adaptation of the resolution method for hybrid logic (Areces) and the implementation of the **HyLoRes** prover (Areces, Gorín, and Heguiabehere).

Strictly speaking, the method is resolution, plus a little paramodulation to handle the equality reasoning. The hybrid resolution rule is significantly simpler than other known approaches to modal resolution — @ and nominals allow us to pull resolvents out of the scope of modalities.

Many first-order resolution optimization techniques transfer to hybrid logic, and Areces and Gorin are currently incorporating such improvements into **HyLoRes**.

**HTab** and **HyLoRes** from the core of the new **InToHyLo** (inference Tools for Hybrid Logic System).

# Summing up ...

- Orthodox modal logic demands proof methods that are applicable to a wide range to of logics. But because it is hard to extract information from under the scope of diamonds it has been forced to rely on Hilbert-systems, thereby sacrificing ease-of-use.
- The new tools offered by the basic hybrid language (nominals and @) enable us to define usable proof systems, such as tableau and natural deduction, basically because they make it easy to pull information out of modal scope.
- These proof methods can be generalized to a wide range of logics (completeness is automatic for pure formulas). Mature implementations now exist.