Expressivity and Inference in Hybrid Logic

Patrick Blackburn

Section of Philosophy and Science Studies, IKH, Roskilde University, Denmark

The Second Tsinghua Logic Summer School

June 27 – July 3, 2022, Beijing, China

What we did yesterday...

- We talked about boxes and diamonds, and showed that they were like little animals, or automata, exploring graphs from the inside. Modal logics are local, internal.
- We mentioned decidability, the standard translation, the first-order correspondence language, frame definability, bisimulations, and the van Benthem Correspondence Theorem, and concluded that modal logic was elegant. . .
- But orthodox modal logic can't refer to the worlds, times, intervals, states, or whatever it is that the nodes in Kripke models are taken to be!
- We solved this by introducing nominals—propositional symbols i, j, k constrained in their interpretation to be true at a unique world in any model. We also introduced the @-operators.
- We showed that our new hybrid tools both dealt with the expressive shortcomings and had inherited modal elegance. And, by lifting the van Benthem Characterization Theorem to cover them, we concluded that we were still doing, good-old modal logic....

Today: Hybrid inference

In today's **lecture (2 $\times$ 45 minutes)** I will present:

- Why inference in orthodox modal logic can be difficult.
- How nominals and @ operators solve this problem.
- Remarks on other proof styles (e.g. natural deduction)
- Hybrid tableaux and how to extend them using pure formulas
- Technical themes: Hybrid tabeleau, the standard translation, and pure formulas
- Conceptual theme: Hybrid inference is simple because hybrid logic lets us use ideas from first-order logic

In today's **tutorial session (1 $\times$ 45 minutes)** I will contrast frame definability for pure formulas and formulas with propositional variables. Pure formulas alway define first-order classes of frames. Answers to Homework 1.

Different graphs, 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 in all models: it's part of the basic, universally applicable, logic.
- But $\Diamond\Diamond p \rightarrow \Diamond p$ is only valid on transitive models. 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.

Simplicity shows in modal axiomatics

Here's the **K**-axiom system (for a basic modal language with a single $\Box$ and $\Diamond$ pair): simply add the following axioms and rules (the **normality** axioms/rules) to propositional logic:

Axioms: $\Box(p \to q) \to (\Box p \to \Box q)$

Rules:

MP: If $\vdash \varphi \to \psi$ and $\vdash \varphi$ then $\vdash \psi$

$\text{Gen}_\Box$: If $\vdash \varphi$ then $\vdash \Box\varphi$

Sub_{prop}: Uniform substitution of formulas for propositional variables

Soundness and completeness:
φ is **K**-provable iff φ is true at every world in every model.

Modal deduction should be general

- Modal logicians like to develop 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).
- As we noted in the first lecture, there is a basic axiomatic system (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 ease 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 φ? 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 inference

- Hybrid inference is based on a simple observation: it's easy to get at the information under the scope of diamonds — for there is a 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 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.

Tasty and sweet

Let's start with a simple example. . .

> *If everything I eat is tasty, and something I eat is sweet, then something I eat is both tasty and sweet.*

We can represent it as follows:

$$[\text{EAT}]\,\text{tasty} \wedge \langle\text{EAT}\rangle\,\text{sweet} \rightarrow \langle\text{EAT}\rangle\,(\text{tasty} \wedge \text{sweet})$$

This is a valid statement, and its validity is easy to establish informally. . .

Informal argument by contradiction

- Suppose "If everything I eat is tasty, and something I eat is sweet, then something I eat is sweet and tasty" is not true.

Informal argument by contradiction

- Suppose "If everything I eat is tasty, and something I eat is sweet, then something I eat is sweet and tasty" is not true.
- Then everything I eat is tasty, and something I eat is sweet. However nothing I eat is both sweet and tasty.

Informal argument by contradiction

- Suppose "If everything I eat is tasty, and something I eat is sweet, then something I eat is sweet and tasty" is not true.
- Then everything I eat is tasty, and something I eat is sweet. However nothing I eat is both sweet and tasty.
- So there is something that I eat—let's call it Jello—that is sweet.

Informal argument by contradiction

- Suppose "If everything I eat is tasty, and something I eat is sweet, then something I eat is sweet and tasty" is not true.
- Then everything I eat is tasty, and something I eat is sweet. However nothing I eat is both sweet and tasty.
- So there is something that I eat—let's call it Jello—that is sweet.
- But as Jello is something I eat, it will be tasty as well as sweet (for everything I eat is tasty).

Informal argument by contradiction

- Suppose "If everything I eat is tasty, and something I eat is sweet, then something I eat is sweet and tasty" is not true.
- Then everything I eat is tasty, and something I eat is sweet. However nothing I eat is both sweet and tasty.
- So there is something that I eat—let's call it Jello—that is sweet.
- But as Jello is something I eat, it will be tasty as well as sweet (for everything I eat is tasty).
- But Jello can't be both tasty and sweet (for nothing I eat is both tasty and sweet). Contradiction!
- So the original statement was true after all.

$$[\mathrm{EAT}]\,\mathrm{tasty} \wedge \langle \mathrm{EAT} \rangle\, \mathrm{sweet} \rightarrow \langle \mathrm{EAT} \rangle\, (\mathrm{tasty} \wedge \mathrm{sweet})$$

$[\mathrm{EAT}]\,\mathrm{tasty} \land \langle \mathrm{EAT} \rangle\,\mathrm{sweet} \to \langle \mathrm{EAT} \rangle\,(\mathrm{tasty} \land \mathrm{sweet})$

1 $\neg @_i([\mathrm{EAT}]\,\mathrm{tasty} \land \langle \mathrm{EAT} \rangle\,\mathrm{sweet} \to \langle \mathrm{EAT} \rangle\,(\mathrm{tasty} \land \mathrm{sweet}))$

$[\mathrm{EAT}]\,\mathrm{tasty} \wedge \langle \mathrm{EAT} \rangle\,\mathrm{sweet} \rightarrow \langle \mathrm{EAT} \rangle\,(\mathrm{tasty} \wedge \mathrm{sweet})$

1	$\neg @_i([\mathrm{EAT}]\,\mathrm{tasty} \wedge \langle \mathrm{EAT} \rangle\,\mathrm{sweet} \rightarrow \langle \mathrm{EAT} \rangle\,(\mathrm{tasty} \wedge \mathrm{sweet}))$
2	$@_i([\mathrm{EAT}]\,\mathrm{tasty} \wedge \langle \mathrm{EAT} \rangle\,\mathrm{sweet})$
$2'$	$\neg @_i \langle \mathrm{EAT} \rangle\,(\mathrm{tasty} \wedge \mathrm{sweet})$

$[\mathrm{EAT}]\,\mathrm{tasty} \wedge \langle \mathrm{EAT} \rangle\,\mathrm{sweet} \rightarrow \langle \mathrm{EAT} \rangle\,(\mathrm{tasty} \wedge \mathrm{sweet})$

1	$\neg @_i([\mathrm{EAT}]\,\mathrm{tasty} \wedge \langle \mathrm{EAT} \rangle\,\mathrm{sweet} \rightarrow \langle \mathrm{EAT} \rangle\,(\mathrm{tasty} \wedge \mathrm{sweet}))$
2	$@_i([\mathrm{EAT}]\,\mathrm{tasty} \wedge \langle \mathrm{EAT} \rangle\,\mathrm{sweet})$
2′	$\neg @_i \langle \mathrm{EAT} \rangle\,(\mathrm{tasty} \wedge \mathrm{sweet})$
3	$@_i [\mathrm{EAT}]\,\mathrm{tasty}$
3′	$@_i \langle \mathrm{EAT} \rangle\,\mathrm{sweet}$

$[\mathrm{EAT}]\,\mathrm{tasty} \land \langle\mathrm{EAT}\rangle\,\mathrm{sweet} \rightarrow \langle\mathrm{EAT}\rangle\,(\mathrm{tasty} \land \mathrm{sweet})$

1	$\neg @_i([\mathrm{EAT}]\,\mathrm{tasty} \land \langle\mathrm{EAT}\rangle\,\mathrm{sweet} \rightarrow \langle\mathrm{EAT}\rangle\,(\mathrm{tasty} \land \mathrm{sweet}))$
2	$@_i([\mathrm{EAT}]\,\mathrm{tasty} \land \langle\mathrm{EAT}\rangle\,\mathrm{sweet})$
2′	$\neg @_i\langle\mathrm{EAT}\rangle\,(\mathrm{tasty} \land \mathrm{sweet})$
3	$@_i[\mathrm{EAT}]\,\mathrm{tasty}$
3′	$@_i\langle\mathrm{EAT}\rangle\,\mathrm{sweet}$
4	$@_i\langle\mathrm{EAT}\rangle\,\mathit{jello}$
4′	$@_{\mathit{jello}}\mathrm{sweet}$

$[\text{EAT}]\,\text{tasty} \wedge \langle\text{EAT}\rangle\,\text{sweet} \rightarrow \langle\text{EAT}\rangle\,(\text{tasty} \wedge \text{sweet})$

1	$\neg @_i([\text{EAT}]\,\text{tasty} \wedge \langle\text{EAT}\rangle\,\text{sweet} \rightarrow \langle\text{EAT}\rangle\,(\text{tasty} \wedge \text{sweet}))$
2	$@_i([\text{EAT}]\,\text{tasty} \wedge \langle\text{EAT}\rangle\,\text{sweet})$
2′	$\neg @_i\langle\text{EAT}\rangle\,(\text{tasty} \wedge \text{sweet})$
3	$@_i[\text{EAT}]\,\text{tasty}$
3′	$@_i\langle\text{EAT}\rangle\,\text{sweet}$
4	$@_i\langle\text{EAT}\rangle\,\mathit{jello}$
4′	$@_{\mathit{jello}}\text{sweet}$
5	$@_{\mathit{jello}}\text{tasty}$

$[\mathrm{EAT}]\,\mathrm{tasty} \wedge \langle\mathrm{EAT}\rangle\,\mathrm{sweet} \rightarrow \langle\mathrm{EAT}\rangle\,(\mathrm{tasty} \wedge \mathrm{sweet})$

1	$\neg @_i([\mathrm{EAT}]\,\mathrm{tasty} \wedge \langle\mathrm{EAT}\rangle\,\mathrm{sweet} \rightarrow \langle\mathrm{EAT}\rangle\,(\mathrm{tasty} \wedge \mathrm{sweet}))$
2	$@_i([\mathrm{EAT}]\,\mathrm{tasty} \wedge \langle\mathrm{EAT}\rangle\,\mathrm{sweet})$
2′	$\neg @_i\langle\mathrm{EAT}\rangle\,(\mathrm{tasty} \wedge \mathrm{sweet})$
3	$@_i[\mathrm{EAT}]\,\mathrm{tasty}$
3′	$@_i\langle\mathrm{EAT}\rangle\,\mathrm{sweet}$
4	$@_i\langle\mathrm{EAT}\rangle\,\mathit{jello}$
4′	$@_{\mathit{jello}}\mathrm{sweet}$
5	$@_{\mathit{jello}}\mathrm{tasty}$
6	$\neg @_{\mathit{jello}}(\mathrm{tasty} \wedge \mathrm{sweet})$

$[\mathrm{EAT}]\,\mathrm{tasty} \wedge \langle\mathrm{EAT}\rangle\,\mathrm{sweet} \rightarrow \langle\mathrm{EAT}\rangle\,(\mathrm{tasty} \wedge \mathrm{sweet})$

1		$\neg @_i([\mathrm{EAT}]\,\mathrm{tasty} \wedge \langle\mathrm{EAT}\rangle\,\mathrm{sweet} \rightarrow \langle\mathrm{EAT}\rangle\,(\mathrm{tasty} \wedge \mathrm{sweet}))$	
2		$@_i([\mathrm{EAT}]\,\mathrm{tasty} \wedge \langle\mathrm{EAT}\rangle\,\mathrm{sweet})$	
2′		$\neg @_i\langle\mathrm{EAT}\rangle\,(\mathrm{tasty} \wedge \mathrm{sweet})$	
3		$@_i[\mathrm{EAT}]\,\mathrm{tasty}$	
3′		$@_i\langle\mathrm{EAT}\rangle\,\mathrm{sweet}$	
4		$@_i\langle\mathrm{EAT}\rangle\,jello$	
4′		$@_{jello}\mathrm{sweet}$	
5		$@_{jello}\mathrm{tasty}$	
6		$\neg @_{jello}(\mathrm{tasty} \wedge \mathrm{sweet})$	
7	$\neg @_{jello}\mathrm{tasty}$		$\neg @_{jello}\mathrm{sweet}$
	$\bot_{5,7}$		$\bot_{4',7}$

Internalizing Labelled Deduction

$\neg$ rules $\qquad \dfrac{@_i \neg\varphi}{\neg @_i \varphi} \qquad \dfrac{\neg @_i \neg\varphi}{@_i \varphi}$

Internalizing Labelled Deduction

¬ rules

$$\frac{@_i\neg\varphi}{\neg @_i\varphi} \qquad \frac{\neg @_i\neg\varphi}{@_i\varphi}$$

∧ rules

$$\frac{@_i(\varphi \wedge \psi)}{\begin{array}{c}@_i\varphi \\ @_i\psi\end{array}} \qquad \frac{\neg @_i(\varphi \wedge \psi)}{\neg @_i\varphi \mid \neg @_i\psi}$$

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 following rules j is a nominal new to the branch where the rule is being applied.

◇ rules

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

Extracting information from modal contexts

In the following rules j is a nominal new to the branch where the rule is being applied.

$\Diamond$ rules

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

$\Box$ rules

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

Link with first-order deduction (Fast Version)

Hybrid Logic	First Order Logic
$@_i \Diamond \varphi$	

Link with first-order deduction (Fast Version)

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

Link with first-order deduction (Fast Version)

Hybrid Logic	First Order Logic
$@_i \Diamond \varphi$	$\exists y(Riy \wedge ST_y(\varphi))$
$@_i \Diamond j$ $@_j \varphi$	

Link with first-order deduction (Fast Version)

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

Link with first-order deduction (Fast Version)

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

Link with first-order deduction (Slow 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.
- 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.

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:

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

$$\frac{@_j i \quad @_k i}{@_j k} \qquad \frac{@_i \Diamond j \quad @_j k}{@_i \Diamond k}$$

$(i \wedge p) \to @_i p$

1 $\neg @_j((i \wedge p) \to @_i p))$

$(i \wedge \mathrm{p}) \rightarrow @_i\mathrm{p}$

1	$\neg @_j((i \wedge \mathrm{p}) \rightarrow @_i\mathrm{p}))$	
2	$@_j(i \wedge \mathrm{p})$	.
2′	$\neg @_j @_i \mathrm{p}$	Propositional rule on 1

$(i \wedge p) \rightarrow @_i p$

1	$\neg @_j((i \wedge p) \rightarrow @_i p))$	
2	$@_j(i \wedge p)$	.
2′	$\neg @_j @_i p$	Propositional rule on 1
3	$@_j i$	
3′	$@_j p$	Propositional rule on 2

$(i \wedge p) \rightarrow @_i p$

1	$\neg @_j((i \wedge p) \rightarrow @_i p))$	
2	$@_j(i \wedge p)$	.
2′	$\neg @_j @_i p$	Propositional rule on 1
3	$@_j i$	
3′	$@_j p$	Propositional rule on 2
4	$\neg @_i p$	@ rule on 2'

$(i \wedge \mathrm{p}) \rightarrow @_i \mathrm{p}$

1	$\neg @_j((i \wedge \mathrm{p}) \rightarrow @_i \mathrm{p}))$	
2	$@_j(i \wedge \mathrm{p})$	.
$2'$	$\neg @_j @_i \mathrm{p}$	Propositional rule on 1
3	$@_j i$	
$3'$	$@_j \mathrm{p}$	Propositional rule on 2
4	$\neg @_i \mathrm{p}$	@ rule on 2'
5	$@_i j$	Equality rule on 3

$(i \wedge \mathrm{p}) \rightarrow @_i\mathrm{p}$

1	$\neg @_j((i \wedge \mathrm{p}) \rightarrow @_i\mathrm{p}))$	
2	$@_j(i \wedge \mathrm{p})$	.
2′	$\neg @_j @_i\mathrm{p}$	Propositional rule on 1
3	$@_j i$	
3′	$@_j\mathrm{p}$	Propositional rule on 2
4	$\neg @_i\mathrm{p}$	@ rule on 2'
5	$@_i j$	Equality rule on 3
6	$@_i\mathrm{p}$	Equality rule on 5 and 3'

$(i \wedge p) \rightarrow @_i p$

1	$\neg @_j((i \wedge p) \rightarrow @_i p))$	
2	$@_j(i \wedge p)$	.
2′	$\neg @_j @_i p$	Propositional rule on 1
3	$@_j i$	
3′	$@_j p$	Propositional rule on 2
4	$\neg @_i p$	@ rule on 2’
5	$@_i j$	Equality rule on 3
6	$@_i p$	Equality rule on 5 and 3’
	$\perp_{4,6}$	

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 can easily be 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

- Suppose $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice} \rightarrow \text{nice}$ is false.

Informal Argument

- Suppose $\langle \textsc{neighbour} \rangle\, [\textsc{neighbour}]\, \text{nice} \to \text{nice}$ is false.
- Then $\langle \textsc{neighbour} \rangle\, [\textsc{neighbour}]\, \text{nice}$ is true, but nice is false.

Informal Argument

- Suppose $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice} \rightarrow \text{nice}$ is false.
- Then $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice}$ is true, but nice is false.
- Then some neighbour (let's call him Julio) who only has nice neighbours (that is, $[\text{NEIGHBOUR}]\, \text{nice}$ is true of Julio).

Informal Argument

- Suppose $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice} \to \text{nice}$ is false.
- Then $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice}$ is true, but nice is false.
- Then some neighbour (let's call him Julio) who only has nice neighbours (that is, $[\text{NEIGHBOUR}]\, \text{nice}$ is true of Julio).
- But neighbourhood is a symmetric relation, hence you are one of Julio's neighbours.

Informal Argument

- Suppose $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice} \to \text{nice}$ is false.
- Then $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice}$ is true, but nice is false.
- Then some neighbour (let's call him Julio) who only has nice neighbours (that is, $[\text{NEIGHBOUR}]\, \text{nice}$ is true of Julio).
- But neighbourhood is a symmetric relation, hence you are one of Julio's neighbours.
- But all Julio's neighbours are nice — so you must be nice too. Contradiction!

Informal Argument

- Suppose $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice} \rightarrow \text{nice}$ is false.
- Then $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice}$ is true, but nice is false.
- Then some neighbour (let's call him Julio) who only has nice neighbours (that is, $[\text{NEIGHBOUR}]\, \text{nice}$ is true of Julio).
- But neighbourhood is a symmetric relation, hence you are one of Julio's neighbours.
- But all Julio's neighbours are nice — so you must be nice too. Contradiction!
- So $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice} \rightarrow \text{nice}$ must true of you after all.

Informal Argument

- Suppose $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice} \rightarrow \text{nice}$ is false.
- Then $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice}$ is true, but nice is false.
- Then some neighbour (let's call him Julio) who only has nice neighbours (that is, $[\text{NEIGHBOUR}]\, \text{nice}$ is true of Julio).
- But neighbourhood is a symmetric relation, hence you are one of Julio's neighbours.
- But all Julio's neighbours are nice — so you must be nice too. Contradiction!
- So $\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice} \rightarrow \text{nice}$ must true of you after all.

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

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

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

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

1	$\neg @_i(\langle \text{NEIGHBOUR} \rangle [\text{NEIGHBOUR}] \text{nice} \rightarrow \text{nice})$	
2	$@_i \langle \text{NEIGHBOUR} \rangle [\text{NEIGHBOUR}] \text{nice}$	
2′	$\neg @_i \text{nice}$	Propositional rule on 1

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

1	$\neg @_i(\langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice} \rightarrow \text{nice})$	
2	$@_i \langle \text{NEIGHBOUR} \rangle\, [\text{NEIGHBOUR}]\, \text{nice}$	
2′	$\neg @_i \text{nice}$	Propositional rule on 1
3	$@_i \langle \text{NEIGHBOUR} \rangle\, julio$	
3′	$@_{julio} [\text{NEIGHBOUR}]\, \text{nice}$	$\Diamond$ rule on 2

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

1	$\neg @_i (\langle \text{NEIGHBOUR} \rangle \, [\text{NEIGHBOUR}] \, \text{nice} \rightarrow \text{nice})$	
2	$@_i \langle \text{NEIGHBOUR} \rangle \, [\text{NEIGHBOUR}] \, \text{nice}$	
2′	$\neg @_i \text{nice}$	Propositional rule on 1
3	$@_i \langle \text{NEIGHBOUR} \rangle \, julio$	
3′	$@_{julio} [\text{NEIGHBOUR}] \, \text{nice}$	$\Diamond$ 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:

$$\frac{@_i\langle\mathrm{NEIGHBOUR}\rangle\, j}{@_j\langle\mathrm{NEIGHBOUR}\rangle\, i}$$

(Here i and j are 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.

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

1	$@_i(\langle \text{NEIGHBOUR} \rangle \, [\text{NEIGHBOUR}] \, \text{nice} \to \text{nice})$	
2	$@_i \langle \text{NEIGHBOUR} \rangle \, [\text{NEIGHBOUR}] \, \text{nice}$	
2′	$\neg @_i \text{nice}$	Propositional rule on 1
3	$@_i \langle \text{NEIGHBOUR} \rangle \, \textit{julio}$	
3′	$@_{\textit{julio}} [\text{NEIGHBOUR}] \, \text{nice}$	$\Diamond$ rule on 2

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

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	$@_i \langle \text{NEIGHBOUR} \rangle \textit{julio}$	
3′	$@_{julio} [\text{NEIGHBOUR}] \text{nice}$	$\Diamond$ rule on 2
4	$@_{julio} \langle \text{NEIGHBOUR} \rangle i$	Symmetry rule on 3

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

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	$@_i\langle \text{NEIGHBOUR} \rangle\, \textit{julio}$	
3′	$@_{julio}[\text{NEIGHBOUR}]\, \text{nice}$	$\diamond$ rule on 2
4	$@_{julio}\langle \text{NEIGHBOUR} \rangle\, i$	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	$@_i \langle \text{NEIGHBOUR} \rangle \, julio$	
3′	$@_{julio} [\text{NEIGHBOUR}] \text{nice}$	$\Diamond$ rule on 2
4	$@_{julio} \langle \text{NEIGHBOUR} \rangle \, i$	Symmetry rule on 3
5	$@_i \text{nice}$	$\Box$ rule on 3’ and 4
	$\bot_{2',5}$	

Loop-free time

Consider the following statement:

> *If time i is earlier that time j, then time j is not earlier than time i.*

We can represent the statement as follows (where F is the Priorean diamond meaning “sometime-in-the-future”):

$$@_i F j \to \neg @_j F i$$

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

But can we prove $Fj \rightarrow \neg @_j Fi$ using our existing tableau system? Let's try…

1 $\neg @_k(@_i Fj \rightarrow \neg @_j Fi)$

But can we prove $Fj \to \neg @_j Fi$ using our existing tableau system? Let's try...

1	$\neg @_k (@_i Fj \to \neg @_j Fi)$	
2	$@_k @_i Fj$	
2′	$\neg @_k \neg @_j Fi$	Propositional rule on 1

But can we prove $Fj \rightarrow \neg @_j Fi$ using our existing tableau system? Let's try...

1	$\neg @_k(@_i Fj \rightarrow \neg @_j Fi)$	
2	$@_k @_i Fj$	
2′	$\neg @_k \neg @_j Fi$	Propositional rule on 1
3	$@_i Fj$	@ rule on 2

But can we prove $Fj \to \neg @_j Fi$ using our existing tableau system? Let's try...

1	$\neg @_k(@_i Fj \to \neg @_j Fi)$	
2	$@_k @_i Fj$	
2′	$\neg @_k \neg @_j Fi$	Propositional rule on 1
3	$@_i Fj$	@ rule on 2
4	$@_j Fi$	¬@¬ rule on 2'

But can we prove $Fj \rightarrow \neg @_j Fi$ using our existing tableau system? Let's try...

1	$\neg @_k (@_i Fj \rightarrow \neg @_j Fi)$	
2	$@_k @_i Fj$	
$2'$	$\neg @_k \neg @_j Fi$	Propositional rule on 1
3	$@_i Fj$	$@$ rule on 2
4	$@_j Fi$	$\neg @ \neg$ 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:

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

Here i, j and k can be any nominals on the tableau 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.

$@_i Fj \to \neg @_j Fi$

1	$\neg @_k(@_i Fj \to \neg @_j Fi)$	
2	$@_k @_i Fj$	
2′	$\neg @_k \neg @_j Fi)$	Propositional rule on 1
3	$@_i Fj$	@ rule on 2
4	$@_j Fi$	$\neg @ \neg$ rule on 2’

$@_i Fj \to \neg @_j Fi$

1	$\neg @_k(@_i Fj \to \neg @_j Fi)$	
2	$@_k @_i Fj$	
2′	$\neg @_k \neg @_j Fi)$	Propositional rule on 1
3	$@_i Fj$	$@$ rule on 2
4	$@_j Fi$	$\neg @ \neg$ rule on 2’
5	$@_i Fi$	Transitivity rule on 3 and 4

$@_i Fj \rightarrow \neg @_j Fi$

1	$\neg @_k (@_i Fj \rightarrow \neg @_j Fi)$	
2	$@_k @_i Fj$	
$2'$	$\neg @_k \neg @_j Fi)$	Propositional rule on 1
3	$@_i Fj$	$@$ rule on 2
4	$@_j Fi$	$\neg @ \neg$ rule on 2'
5	$@_i Fi$	Transitivity rule on 3 and 4
6	$\neg @_i Fi$	Irreflexivity rule

$@_i F j \to \neg @_j F i$

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

Pure formulas and inference

- It's time to be more precise about what completeness results are possible here.
- This brings us back to pure formulas.
- Recall that 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).

Frame definability

A formula defines a class of frames if it is valid on precisely the frames belonging to that class. Recall that we can define many classes of frames using pure formulas, including some that we can't define in ordinary modal logic:

$@_i \Diamond i$	*Reflexivity*
$@_i \neg\Diamond i$	*Irreflexivity*
$@_i \Diamond j \rightarrow @_j \neg\Diamond i$	*Asymmetry*

From pure formulas to tableau rules

Let φ be a pure formula, built out of nominals $i_1, \ldots, i_n$. Then the simplest 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 φ and any nominal i not occurring in φ, φ 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.

Three useful references

That concludes our informal introduction to tableau systems for hybrid logic. The following three papers will help you explore the systems we have just discussed in more detail:

- "Representation, reasoning, and relational structures: a hybrid logic manifesto", Patrick Blackburn, Logic Journal of the IGPL, 8(3), pp. 339-365, 2000. Very close in style to these lectures — but takes things a little further, treating some material we discuss in later lectures as well.
- "Internalizing labelled deduction", Patrick Blackburn, *Journal of Logic and Computation*, 10(1), pp. 137–168, 2000. Focusses on deductive issues, both conceptual and technical, and proves a number of tableau completeness results.
- "Termination for hybrid tableaus", Thomas Bolander and Patrick Blackburn, *Journal of Logic and Computation*,17(3), pp.517-554, 2007. This paper shows how to guarantee that these tableaux systems halt (terminate) thereby showing that they provide a complete algorithm for hybrid validity (and satisfiability).

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?
- Seligman style systems.
- Is any of this stuff implementable?

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.
- This is fairly easy to imagine for tableau completeness proofs: simply 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.
- This is essentially what lies behind the completeness proofs in the paper "Internalizing Labelled Deduction" mentioned on the previous slide.

But it also works for axiom systems

- Tomorrow we discuss axiom systems for hybrid logic, and we can prove completeness results using what is essentially the Henkin method used for first-order logic.
- That is: in sharp contrast to the canonical model method used in orthodox model logic (which we saw yesterday) which builds models out of a (typically uncountably infinite) collection of Maximal Consistent Sets (MCS) of formulas, in hybrid logic models can be built out of a *single* MCS by taking equivalence classes of nominals. Two useful references here are:
  - "Hybrid Logic", Chapter 7, Section 3 of Modal Logic, Patrick Blackburn, Maarten de Rijke and Yde Venema, Cambridge University Press, 2001.
  - "Pure Extensions, Proof Rules, and Hybrid Axiomatics", Patrick Blackburn and Balder ten Cate, Studia Logica, 84(2), pp. 277-322, 2006.
- In essence: we can build special MCSs in which each formula of the form $\Diamond\varphi$ is "witnessed" by a new nominal i which names the world where φ holds. We then build the model out of (equivalence classes of) these "witness nominals".

Named models are important

- Moreover, the models we build in this Henkin style approach are named. (A named model is a model in which every point is named by some nominal.)
- A simple model theoretic argument shows that if all pure instances of a pure formula φ are true at all states in a named model, then the underlying frame validates φ.
- So — as we shall see tomorrow — we have automatic completeness for axiomatic systems too: any extended logic obtained by adding pure axioms is guaranteed to be complete with respect to the frame class the pure axioms define.
- The two references on the previous slide discuss this in detail.

Can we adapt these ideas to other proof styles?

Yes. The key insight is that the combination of nominals and @ allows 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, Braüner), resolution calculi (Areces), and display calculi (Demri and Goré).

Here's a partial list of work on hybrid deduction in other proof styles. . .

Other proof styles

- "Resolution in modal, description and hybrid logic", Carlos Areces, Maarten de Rijke, and Hans de Nivelle, *Journal of Logic and Computation*, 11(5), pp. 717-736, 2001.
- "Ordered Resolution with Selection for $\mathcal{H}(@)$", Carlos Areces and Daniel Gorín, *International Conference on Logic for Programming Artificial Intelligence and Reasoning*. Springer, 2005.
- *Hybrid logic and its proof-theory*, Torben Braüner, Springer, 2010. ISBN-10: 9400700024
- "Display calculi for nominal tense logics", Stéphane Demri and Rajeev Goré, *Journal of Logic and Computation*, 12(6), pp. 993-1016, 2002.
- "The logic of correct description", Jerry Seligman, In M. de Rijke, editor, *Advances in Intensional Logic*, pages 107–135. Kluwer, 1997.
- "Internalization: The case of hybrid logics", Jerry Seligman, *Journal of logic and computation*, 11(5), pp. 671–689, 2001.

Let's take a quick look at (one of) the way(s) Torben Braüner handles natural deduction for hybrid logic in his book...

Some basic natural deduction rules

$$\dfrac{\begin{array}{c}[@_i\varphi] \\ \vdots \\ @_i\psi\end{array}}{@_i(\varphi \to \psi)}\ (\to I) \qquad \dfrac{@_i(\varphi \to \psi) \quad @_i\varphi}{@_i\psi}\ (\to E)$$

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

The familiar introduction and elimination rules for $\to$ (prefixed by @-operators), and the introduction and elimination rules for @.

Natural deduction rules for modalities

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

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

On the left, the crucial rule: the one which lets us access the information under the scope of a diamond. Suppose we assume that i has some successor, which we'll call j (that is: assume $@_i\Diamond j$). Suppose this lets us conclude $@_j\varphi$. Then we are free to discharge our assumption and conclude $@_i\Box\varphi$. **Why? Because j is an arbitrary successor of i.**

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

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

Seligman-style systems

The tableau systems used today do inference "under the scope" of $@_i$ or $\neg @_i$: everything on a tableau is preceded by one of these "labels".

And in the Torben Braüner natural deduction system that we saw, everything is under the scope of "under the scope" of $@_i$.

However, Jeremy Seligman has developed several elegant systems in which such labelling is not needed.

- "The logic of correct description", J. Seligman, In M. de Rijke, editor, *Advances in Intensional Logic*, pages 107–135. Kluwer, 1997.
- "Internalization: The case of hybrid logics", J. Seligman, *Journal of logic and computation*, 11(5), pp. 671–689, 2001.
- "Completeness and termination for a Seligman-style tableau system", P. Blackburn, T. Bolander, T. Braü ner, K. F. Jørgensen (2017). Journal of Logic and Computation, 27(1), 81-107.

Is any of this stuff implementable?

The tableau material certainly is, but we need to be careful. For example, the tableau equality rules discussed today are nice for hand calculation, but naive computationally. Among other changes, **HTab** system implements more sophisticated equality rules:

"Htab: a terminating tableaux system for hybrid logic," Carlos Areces and Guillaume Hoffmann, *Electronic Notes in Theoretical Computer Science*, 231, pp. 3-19, 2009.

"Lightweight hybrid tableaux", Guillaume Hoffmann, *Journal of Applied Logic* 8(4), pp. 397-408, 2010.

The **Spartacus** prover, based on rather different ideas, is also impressive:

"Spartacus: A tableau prover for hybrid logic," Daniel Götzmann, Mark Kaminski and Gert Smolka. *Electronic Notes in Theoretical Computer Science*, 262, 127-139, 2010.

"Terminating tableau systems for hybrid logic with difference and converse", Mark Kaminski and Gert Smolka. *Journal of Logic, Language and Information*, 18(4), pp. 437-464, 2009.

And even more. . .

Handling good equality reasoning efficiently is crucial to obtaining high-performing tableaux implementations for hybrid logic. The following papers explore this theme both theoretically, and with an experimental comparison between the **Herod** and **Pilate** provers.

"An efficient approach to nominal equalities in hybrid logic tableaux", Serenella Cerrito and Marta Cialdea Mayer, *Journal of Applied Non-classical Logics* 20(1-2), pp. 39-61, 2010.

"Nominal Substitution at Work with the Global and Converse Modalities", Serenella Cerrito and Marta Cialdea Mayer, *Advances in Modal Logic 8*, pp. 57-74, 2010.

"Herod and Pilate: two tableau provers for basic hybrid logic", Marta Cialdea Mayer and Serenella Cerrito, *International Joint Conference on Automated Reasoning*, Springer, 2010.

For a general approach using an "unrestricted blocking" tableau rule see:
"Decision procedures for some strong hybrid logics", Andrzej Indrzejczak and Michał Zawidzki, Logic and Logical Philosophy 22(4), pp. 389-409, 2013.

A resolution theorem prover for hybrid logic

Resolution provers rule in first-order logic, but they are not common in modal and description logic, where tableau methods reign. However, as the **HyLoRes** prover demonstrated, resolution works well in hybrid logic: @ and nominals allow us to pull resolvents out of the scope of modalities, and paramodulation can be used to handle the equality reasoning.

"HyLoRes 1.0: Direct Resolution for Hybrid Logics", Carlos Areces and Juan Heguiabehere, *Proceedings of International Conference on Automated Deduction, CADE-18*, Springer, pp. 156-160, 2002.

Summing up ...

- Orthodox modal logic demands proof methods that are applicable to a wide range of frame classes. 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 devise usable proof systems (like tableau and natural deduction) basically because they make it easy to access information under the scope of a diamond.
- Indeed, nominals and @ make it possible to prove completeness for hybrid Hilbert systems using (essentially) the Henkin method for first-order logic: building a model from a single "witnessed" MCS.
- These proof methods generalize to a wide range of frame classes, as completeness is automatic for pure formulas.
- Sophisticated implementations exist.