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 explored the consequences of hybridization for modal inference. In particular, we learned that the basic hybrid apparatus gives us precisely what we need to extract information from under the scope of diamonds. The key idea is the inferential move from $@_i\Diamond\varphi$ to $@_i\Diamond j$ and $@_j\varphi$.
- I emphasized the close connection of this idea with the first-order rule of Existential Instantiation. The inferential move, from $@_i\Diamond\varphi$ to $@_i\Diamond j$ and $@_j\varphi$, is just the standard translation of the first-order Existential Instantiation rule.
- I also mentioned that pure axioms always yield complete logics; I will come back to this important point today. This is about inference.
- I also used the standard translation to show that pure axioms always define first-order frame conditions. (This was on the notes that we discussed at the end.) This is about expressivity.

Today: Hybrid inference

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

- The $\mathbf{K}_h$ axiom system — a sound and complete axiom system.
- Why $\mathbf{K}_h$ is not the axiomatization we want. There is a better one — and I will explain why.
- The Name and Paste rules, and a Hybrid Lindenbaum Lemma
- A completeness result for The $\mathbf{K}_h + \text{RULES}$ axiom system.
- Technical themes: MCSs that contain other MCSs, Named MCSs, linking Name, Paste, and Tableaus, and Henkin's second idea.
- Conceptual theme: Hybrid logics with pure axioms are like first-order theories.

In today's **tutorial session** **(1 $\times$ 45 minutes)** I will answer questions and we can look at the answers to Homework 2.

Why are general completeness proofs often straightforward in hybrid logic?

- **First:** because the basic hybrid logic enables us to use first-order model building techniques.
- This is fairly easy to imagine for tableau completeness proofs: simply observe that the tableau rules crunch formulas down into expressions of the form $(\neg)@_i\mathrm{p}$, $(\neg)@_ij$ and $(\neg)@_i\Diamond j$. Open branches are thus Robinson diagrams of satisfying models, and completeness can be proved by the Hintikka set method. We will see another first-order model building technique today.
- **Second:** because pure axioms always yield complete logics, we get general completeness theorems.

Today we prove such a general completeness result

- We will discuss two axiom systems for hybrid logic, and we will prove a general strong completeness results for the second one using what is essentially the Henkin method used for first-order logic.
- The model construction we use is related to the canonical model method used in ordinary model logic (which we revised on Monday; remember the Revision Sheet) which builds models out of large collections of Maximal Consistent Sets (MCS) of formulas. However, as we shall see, in hybrid logic models can be built out the information in a single MCS. 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 will show how to 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.

Why look at axiom systems?

- Axioms systems (or Hilbert-style proof systems) are not as practical as tableau or natural deduction, but they have a simple structure. It is easier to prove things about them.
- Also — this course is about inference, not about one kind of system. I wanted us look at two very different systems, tableau and axiomatic systems, because I want you to see what they have in common.
- And today we will again see the same important inferential move: from $@_i\Diamond\varphi$ to $@_i\Diamond j$ and $@_j\varphi$, though it is presented in a different setting.

Reminder: what an MCS is

A set of formulas Σ is maximal consistent iff

- Σ is consistent (i.e. $\Sigma \nvdash \bot$);
- no proper superset of Σ is consistent.

Maximal consistent sets of sentences have many nice properties:

- $\varphi \in \Sigma$ or $\neg\varphi \in \Sigma$;
- $\Sigma \vdash \varphi$ iff $\varphi \in \Sigma$ (Deductive closure);
- $\varphi \in \Sigma$ iff $\neg\varphi \notin \Sigma$;
- $\varphi \wedge \psi \in \Sigma$ iff $\varphi \in \Sigma$ and $\psi \in \Sigma$
- $\varphi \vee \psi \in \Sigma$ iff $\varphi \in \Sigma$ or $\psi \in \Sigma$
- $\varphi \to \psi \in \Sigma$ iff $\varphi \notin \Sigma$ or $\psi \in \Sigma$

Why look at Henkin-style completeness proofs?

- Henkin's first idea — use MCSs to build models — is technically natural. It lets us build models (semantic structures) out of sets that contain syntactic structure.
- MCSs are modally natural. Lots of possible worlds — lots of MCSs! A collection of balloons!
- Moreover, in a hybrid setting, we bring in Henkin's second idea. . .
- Henkin used new first-order constants to help build models. We shall use new nominals to help build models in similar way.

The $\mathbf{K}_h$-axiom system: Step 0

We now present the $\mathbf{K}_h$-axiom system (for a basic modal language with a single $\Box$ and $\Diamond$ pair). As Step 0, we start with **K** as defined in Lecture 1:

Axioms:

$\mathbf{K}_\Box$: $\Box(p \to q) \to (\Box p \to \Box q)$

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

The $\mathbf{K}_h$-axiom system: Step 1

We now present the $\mathbf{K}_h$-axiom system (for a basic modal language with a single $\Box$ and $\Diamond$ pair). As Step 1, we add the following:

Axioms:

$\mathbf{K}_\Box$: $\Box(p \to q) \to (\Box p \to \Box q)$

$\mathbf{K}_@$: $@_i(p \to q) \to (@_i p \to @_i q)$, for all nominals i.

Rules:

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

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

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

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

Sub_{nom}: Uniform substitution of nominals (and nothing else) for nominals

The remaining $\mathbf{K}_h$ axioms

(*intro*)	$i \wedge p \to @_i p$
(*agree*)	$@_j @_i p \leftrightarrow @_i p$
(*back*)	$\Diamond @_i p \to @_i p$
(*self-dual*)	$@_i p \leftrightarrow \neg @_i \neg p$
(*ref*)	$@_i i$
(*sym*)	$@_i j \leftrightarrow @_j i$
(*nom*)	$@_i j \wedge @_j p \to @_i p$

We can prove $\Diamond i \wedge @_i p \to \Diamond p$ (*bridge*) in $\mathbf{K}_h$. In fact, we can prove all valid formulas: $\mathbf{K}_h$ is sound and complete.

Weak soundness and completeness

Soundness: "Only valid formulas are provable"

$$\varphi \text{ is provable } \Rightarrow \varphi \text{ is valid}$$

Completeness: "All valid formulas are provable"

$$\varphi \text{ is valid } \Rightarrow \varphi \text{ is provable}$$

Soundness: $\vdash \varphi \;\Rightarrow\; \models \varphi$

Completeness: $\models \varphi \;\Rightarrow\; \vdash \varphi$

Strong soundness and completeness

Let Γ be a set of formulas.

$\Gamma \vdash \varphi$ means that there is a finite set $\{\gamma_1, \ldots, \gamma_n\} \subseteq \Gamma$ such that $\vdash \gamma_1 \wedge \ldots \wedge \gamma_n \to \varphi$.

$\Gamma \vDash \varphi$ means that for any model $\mathcal{M}$ and any state w, if $\mathcal{M}, w \vDash \Gamma$ then $\mathcal{M}, w \vDash \varphi$.
(Here $\mathcal{M}, w \vDash \Gamma$ means that all formulas in Γ are true at w in $\mathcal{M}$.)

Soundness: $\Gamma \vdash \varphi \;\Rightarrow\; \Gamma \vDash \varphi$.

Completeness: $\Gamma \vDash \varphi \;\Rightarrow\; \Gamma \vdash \varphi$

How can we prove completeness?

We could try and use the canonical model:

- W is the set of all MCSs;
- $\Sigma R \Delta$ iff $\Delta \subseteq \{\varphi \mid \Diamond\varphi \in \Sigma\}$ (or equivalently, $\{\varphi \mid \Box\varphi \in \Sigma\} \subseteq \Delta$).
- $\Sigma \in V(p)$ iff $p \in \Sigma$, for all proposition letters p.
- $\Sigma \in V(i)$ iff $i \in \Sigma$, for all nominals i.

Problem: nominals need to be true at one single state. But there are many MCSs that contain (say) the nominal i. The above definition does not work. We can't use all the MCSs.

We need to be more clever. Let's look more closely at these MCSs.

MCSs inside MCSs. . .

Instead of looking at the set of all MCSs, let's look more closely inside a single MCS. If we do, we discover something interesting: $\mathbf{K}_h$-MCSs have other $\mathbf{K}_h$-MCSs hidden inside them. And as we shall see, this is very useful structure.

Lemma 7.24 (Blue Book) Let Γ be a $\mathbf{K}_h$-MCS. For every nominal i, let Δ_i be $\{\varphi \mid @_i\varphi \in \Gamma\}$. Then:

1. For every nominal i, Δ_i is a $\mathbf{K}_h$-MCS that contains i.
2. For all nominals i and j, if $i \in \Delta_j$, then $\Delta_j = \Delta_i$.
3. For all nominals i and j, $@_i\varphi \in \Delta_j$ iff $@_i\varphi \in \Gamma$.
4. If a nominal $k \in \Gamma$, then $\Gamma = \Delta_k$.

We call such Δ_i the named sets yielded by Γ.

MCSs inside MCSs...

Instead of looking at the set of all MCSs, let's look more closely inside a single MCS. If we do, we discover something interesting: $\mathbf{K}_h$-MCSs have other $\mathbf{K}_h$-MCSs hidden inside them. For a start, any nominal i is in a single MCS.

Lemma 7.24 (Blue Book) Let Γ be a $\mathbf{K}_h$-MCS. For every nominal i, let Δ_i be $\{\varphi \mid @_i\varphi \in \Gamma\}$. Then:

1. For every nominal i, Δ_i is a $\mathbf{K}_h$-MCS that contains i.
2. For all nominals i and j, if $i \in \Delta_j$, then $\Delta_j = \Delta_i$.
3. For all nominals i and j, $@_i\varphi \in \Delta_j$ iff $@_i\varphi \in \Gamma$.
4. If a nominal $k \in \Gamma$, then $\Gamma = \Delta_k$.

We call such Δ_i the named sets yielded by Γ.

MCSs inside MCSs...

Instead of looking at the set of all MCSs, let's look more closely inside a single MCS. If we do, we discover something interesting: $\mathbf{K}_h$-MCSs have other $\mathbf{K}_h$-MCSs hidden inside them. And think about what these two clauses are telling us.

Lemma 7.24 (Blue Book) Let Γ be a $\mathbf{K}_h$-MCS. For every nominal i, let Δ_i be $\{\varphi \mid @_i\varphi \in \Gamma\}$. Then:

1. For every nominal i, Δ_i is a $\mathbf{K}_h$-MCS that contains i.
2. For all nominals i and j, if $i \in \Delta_j$, then $\Delta_j = \Delta_i$.
3. For all nominals i and j, $@_i\varphi \in \Delta_j$ iff $@_i\varphi \in \Gamma$.
4. If a nominal $k \in \Gamma$, then $\Gamma = \Delta_k$.

We call such Δ_i the named sets yielded by Γ.

Δ_i is a $\mathbf{K}_h$-MCS that contains i

First. For every nominal i we have the *ref* axiom $@_i i$, hence $i \in \Delta_i$.

Second. Δ_i is consistent. For assume it is not. Then there are $\delta_1 \ldots \delta_n \in \Delta_i$ such that $\vdash \neg(\delta_1 \wedge \cdots \wedge \delta_n)$. By $@_i$-necessitation, $\vdash @_i\neg(\delta_1 \wedge \cdots \wedge \delta_n)$, hence $@_i\neg(\delta_1 \wedge \cdots \wedge \delta_n)$ is in Γ, and thus by *self-dual* $\neg @_i(\delta_1 \wedge \cdots \wedge \delta_n)$ is in Γ too.
On the other hand, as $\delta_1 \ldots \delta_n \in \Delta_i$, we have $@_i\delta_1 \wedge \cdots \wedge @_i\delta_n \in \Gamma$. As $@_i$ is a normal modality, $@_i(\delta_1 \wedge \cdots \wedge \delta_n) \in \Gamma$ as well, contradicting the consistency of Γ. So Δ_i is consistent.

Third. Δ_i is maximal. For assume it is not. Then there is a formula χ such that neither χ nor $\neg\chi$ is in Δ_i. But then both $\neg @_i\chi$ and $\neg @_i\neg\chi$ belong to Γ, and this is impossible: if $\neg @_i\chi \in \Gamma$, then by *self-duality* $@_i\neg\chi \in \Gamma$ as well. So Δ_i is a $\mathbf{K}_h$-MCS that contains i.

If a nominal $k \in \Gamma$, then $\Gamma = \Delta_k$

Suppose $k \in \Gamma$. That is, suppose that k names Γ.

Let $\varphi \in \Gamma$. Then as $k \in \Gamma$, by *intro* we have $@_k\varphi \in \Gamma$, and hence $\varphi \in \Delta_k$. (Recall that *intro* is $(i \wedge p) \rightarrow @_i p$.)

Conversely, if $\varphi \in \Delta_k$, then $@_k\varphi \in \Gamma$. But as $k \in \Gamma$ we can infer $\varphi \in \Gamma$, (Note that $(@_i p \wedge i) \rightarrow p$ follows from *intro* by propositional logic.)

Choice point: what shall we do. . . ?

This is interesting structure! We can make one $\mathbf{K}_h$ MCS, and it seems that we can see many $\mathbf{K}_h$ MCSs inside!

And these MCSs can be used to prove a completeness result for $\mathbf{K}_h$ MCS.

Every $\mathbf{K}_h$-consistent set of formulas is satisfiable.

That is, $\mathbf{K}_h$ really is the minimal hybrid logic.

But I will not prove this here — because we can prove something more interesting, as I will now explain. . .

An important lemma. . .

This Lemma tells us that if we can build a named model, then we can prove a much more interesting result:

Lemma 7.22 (Blue Book) Let $\mathcal{M} = (W, R, V)$ be a named model and φ a pure formula. Suppose that for all pure instances ψ of φ, $\mathcal{M} \models \psi$. Then $(W, R) \models \varphi$.

That is, for named models and pure formulas the gap between truth in a model and validity in a frame is non-existent.

If we could figure out how to to build a named model, then not only would we have completeness for the minimal logic, we would have completeness whenever we added a pure formula as an extra axiom.

An example

Let us suppose that we know how to build a named model. Why is this Lemma useful...?

Lemma 7.22 (Blue Book) Let $\mathcal{M} = (W, R, V)$ be a named model and φ a pure formula. Suppose that for all pure instances ψ of φ, $\mathcal{M} \models \psi$. Then $(W, R) \models \varphi$.

An example

Let us suppose that we know how to build a named model. Why is this Lemma useful. . . ?

Lemma 7.22 (Blue Book) Let $\mathcal{M} = (W, R, V)$ be a named model and φ a pure formula. Suppose that for all pure instances ψ of φ, $\mathcal{M} \models \psi$. Then $(W, R) \models \varphi$.

Suppose we add the irreflexivity axiom: $i \rightarrow \neg\Diamond i$. Then this axiom will be true in our named model.

An example

Let us suppose that we know how to build a named model. Why is this Lemma useful. . . ?

Lemma 7.22 (Blue Book) Let $\mathcal{M} = (W, R, V)$ be a named model and φ a pure formula. Suppose that for all pure instances ψ of φ, $\mathcal{M} \models \psi$. Then $(W, R) \models \varphi$.

Suppose we add the irreflexivity axiom: $i \to \neg\Diamond i$. Then this axiom will be true in our named model.

But as we can substitute nominals for nominals, so will all its variants $j \to \neg\Diamond j$, $k \to \neg\Diamond k$, $l \to \neg\Diamond l$, . . .

An example

Let us suppose that we know how to build a named model. Why is this Lemma useful. . . ?

Lemma 7.22 (Blue Book) Let $\mathcal{M} = (W, R, V)$ be a named model and φ a pure formula. Suppose that for all pure instances ψ of φ, $\mathcal{M} \models \psi$. Then $(W, R) \models \varphi$.

Suppose we add the irreflexivity axiom: $i \to \neg\Diamond i$. Then this axiom will be true in our named model.

But as we can substitute nominals for nominals, so will all its variants $j \to \neg\Diamond j$, $k \to \neg\Diamond k$, $l \to \neg\Diamond l$, . . .

Furthermore, by @-generalization, we can put $@_i$, $@_j$, $@_k$, $@_l$, . . . in front of any of these.

An example

Let us suppose that we know how to build a named model. Why is this Lemma useful. . . ?

Lemma 7.22 (Blue Book) Let $\mathcal{M} = (W, R, V)$ be a named model and φ a pure formula. Suppose that for all pure instances ψ of φ, $\mathcal{M} \models \psi$. Then $(W, R) \models \varphi$.

Suppose we add the irreflexivity axiom: $i \to \neg\Diamond i$. Then this axiom will be true in our named model.

But as we can substitute nominals for nominals, so will all its variants $j \to \neg\Diamond j$, $k \to \neg\Diamond k$, $l \to \neg\Diamond l$, . . .

Furthermore, by @-generalization, we can put $@_i$, $@_j$, $@_k$, $@_l$, . . . in front of any of these.

As the model is named, this means that all variants are true at all points in the named model — and thus valid on the underlying frame.

Here's our plan...

1. Given a consistent set of sentences Σ, we are going to blow it up to an MCS Σ^* using a Lindenbaum Lemma.

Here's our plan...

1. Given a consistent set of sentences Σ, we are going to blow it up to an MCS Σ^* using a Lindenbaum Lemma.
2. We will use the MCSs within this MCS — the ones we saw in Lemma 7.24 — to build the model.

Here's our plan. . .

1. Given a consistent set of sentences Σ, we are going to blow it up to an MCS Σ^* using a Lindenbaum Lemma.
2. We will use the MCSs within this MCS — the ones we saw in Lemma 7.24 — to build the model.
3. But we want more structure than Lemma 7.24 provides.

Here's our plan...

1. Given a consistent set of sentences Σ, we are going to blow it up to an MCS Σ^* using a Lindenbaum Lemma.
2. We will use the MCSs within this MCS — the ones we saw in Lemma 7.24 — to build the model.
3. But we want more structure than Lemma 7.24 provides.
4. We also want to make sure that Σ^* contains some nominal (that is, Σ^* is named by some nominal). Problem: how will we do this?

Here's our plan...

1. Given a consistent set of sentences Σ, we are going to blow it up to an MCS Σ^* using a Lindenbaum Lemma.
2. We will use the MCSs within this MCS — the ones we saw in Lemma 7.24 — to build the model.
3. But we want more structure than Lemma 7.24 provides.
4. We also want to make sure that Σ^* contains some nominal (that is, Σ^* is named by some nominal). Problem: how will we do this?
5. We also want to make sure that whenever $@_i\Diamond\varphi$ is in Σ^*, then for some nominal j we have that both $@_i\Diamond j$ and $@_j\varphi$ are in Σ^* too. Problem: how will we do this?

The NAME and PASTE rules

How do we solve these two problems? We do so by adding two more rules to $\mathbf{K}_h$. We call this logic $\mathbf{K}_h$+RULES

$$(\textsc{name})\ \frac{\vdash j \to \theta}{\vdash \theta} \qquad\qquad (\textsc{paste})\ \frac{\vdash @_i\Diamond j \wedge @_j\varphi \to \theta}{\vdash @_i\Diamond\varphi \to \theta}$$

In both rules, j is a nominal distinct from i that does not occur in φ or θ. Both rules are sound — they both preserve frame validity. The NAME rule is going to solve our first problem, the PASTE rule our second.

Named and pasted MCSs

We say that a $\mathbf{K}_h + \text{RULES}$-MCS Γ is *named* iff some nominal j belongs to Γ.

We say that a $\mathbf{K}_h + \text{RULES}$-MCS Γ is *pasted* iff $@_i\Diamond\varphi \in \Gamma$ implies that for some nominal j, $@_i\Diamond j \wedge @_j\varphi \in \Gamma$.

Given a consistent set of sentences Σ, we will expand it using a Lindenbaum Lemma to a named and pasted MCS, and then build our model out of that.

Lindenbaum Lemma

Suppose we start with language $\mathcal{L}$. Add a (countably infinite) set of new nominals to the language, and call this enriched language $\mathcal{L}'$. Similar to what is done in Henkin proofs in first-order logic where we add new constants.

Then every $\mathbf{K}_h + \textsc{rules}$-consistent set of formulas in language $\mathcal{L}$ can be extended to a named and pasted $\mathbf{K}_h + \textsc{Rules-mcs}$ in language $\mathcal{L}'$.

How to prove it: step 1

Enumerate the new nominals.

Given a consistent set of $\mathcal{L}$-formulas Σ, define Σ_k to be $\Sigma \cup \{k\}$, where k is the first new nominal in our enumeration. That is, name Σ with a new nominal k.

Σ_k is consistent. For suppose not. Then for some conjunction of formulas θ from Σ, $\vdash k \rightarrow \neg\theta$. But as k is a new nominal, it does not occur in θ; hence, by the NAME rule, $\vdash \neg\theta$. But this contradicts the consistency of Σ, so Σ_k must be consistent.

How to prove it: step 2

We now paste. Enumerate all the formulas of $\mathcal{L}'$, define Σ^0 to be Σ_k, and suppose we have defined Σ^m, where $m \geq 0$.

Let φ_{m+1} be the $(m+1)$-th formula in our enumeration of $\mathcal{L}'$. We define Σ^{m+1} as follows:

If $\Sigma^{m+1} \cup \{\varphi_{m+1}\}$ is inconsistent, then $\Sigma^{m+1} = \Sigma^m$. Otherwise:

1. $\Sigma^{m+1} = \Sigma^m \cup \{\varphi_{m+1}\}$ if φ_{m+1} is not of the form $@_i\Diamond\varphi$. (Here i can be any nominal.)
2. $\Sigma^{m+1} = \Sigma^m \cup \{\varphi_{m+1}\} \cup \{@_i\Diamond j \wedge @_j\varphi\}$, if φ_{m+1} is of the form $@_i\Diamond\varphi$. (Here j is the first new nominal in the nominal enumeration that does not occur in Σ^m or $@_i\Diamond\varphi$.)

How to prove it: step 3

Let $\Sigma^* = \bigcup_{n \geq 0} \Sigma^n$.

Clearly this set is named (by k), maximal, and pasted.

Furthermore, it is consistent, for the only non-trivial aspect of the expansion is that defined by the second item, and the consistency of this step is precisely what the PASTE rule guarantees.

Note the similarity of this argument to the standard completeness proof for first-order logic: in essence, PASTE gives us the deductive power required to use nominals as Henkin constants.

Back to the MCSs inside the MCS

Lemma 7.24 (Blue Book) Let Γ be a $\mathbf{K}_h$-MCS. For every nominal i, let Δ_i be $\{\varphi \mid @_i\varphi \in \Gamma\}$. Then:

1. For every nominal i, Δ_i is a $\mathbf{K}_h$-MCS that contains i.
2. For all nominals i and j, if $i \in \Delta_j$, then $\Delta_j = \Delta_i$.
3. For all nominals i and j, $@_i\varphi \in \Delta_j$ iff $@_i\varphi \in \Gamma$.
4. If a nominal $k \in \Gamma$, then $\Gamma = \Delta_k$.

Defining the model

Let Γ be a named and pasted $\mathbf{K}_h + \textsc{Rules}$-MCS. The named model yielded by Γ is $\mathcal{M}^\Gamma = (W^\Gamma, R^\Gamma, V^\Gamma)$.

- Here W^Γ is the set of all named sets yielded by Γ; that is, the MCSs inside Γ.
- R is the restriction to W^Γ of the usual canonical relation between MCSs (so $uR^\Gamma v$ iff for all formulas φ, $\varphi \in v$ implies $\Diamond\varphi \in u$);
- V^Γ is the usual canonical valuation (so for any atom a, $V^\Gamma(a) = \{w \in W^\Gamma \mid a \in w\}$).

Existence Lemma

Existence Lemma (Lemma 7.27 Blue Book): Let Γ be a named and pasted $\mathbf{K}_h + \text{RULES-MCS}$, and let $\mathcal{M} = (W, R, V)$ be the named model yielded by Γ. Suppose $u \in W$ and $\Diamond\varphi \in u$. Then there is a $v \in W$ such that uRv and $\varphi \in v$.

Proof: As $u \in W$, for some nominal i we have that $u = \Delta_i$. Hence as $\Diamond\varphi \in u$, $@_i\Diamond\varphi \in \Gamma$. But Γ is pasted, so for some nominal j, $@_i\Diamond j \wedge @_j\varphi \in \Gamma$, and so $\Diamond j \in \Delta_i$ and $\varphi \in \Delta_j$. If we could show that $\Delta_i R \Delta_j$, then Δ_j would be a suitable choice of v. So suppose $\psi \in \Delta_j$. This means that $@_j\psi \in \Gamma$. By @-agreement (item (iii) of Lemma 7.24) $@_j\psi \in \Delta_i$. But $\Diamond j \in \Delta_i$. Hence, by *bridge*, $\Diamond i \wedge @_i p \to \Diamond p$, $\Diamond\psi \in \Delta_i$ as required.

Truth Lemma

Truth Lemma (Lemma 7.28 Blue Book): Let $\mathcal{M} = (W, R, V)$ be the named model yielded by a named and pasted $\mathbf{K}_h + \textsc{rules-mcs}$ Γ, and let $u \in W$. Then, for all formulas φ, $\varphi \in u$ iff $\mathcal{M}, u \models \varphi$.

Proof: Induction on the structure of φ. The atomic and boolean cases are clear, and we use the Existence Lemma just proved for the modalities.

For the satisfaction operators, suppose $\mathcal{M}, u \models @_i\psi$. This happens iff $\mathcal{M}, \Delta_i \models \psi$ (for by items (i) and (ii) of Lemma 7.24, Δ_i is the only MCS containing i, and hence, by the the atomic case of the present lemma, the only state in $\mathcal{M}$ where i is true) iff $\psi \in \Delta_i$ (inductive hypothesis) iff $@_i\psi \in \Delta_i$ iff $@_i\psi \in u$ ($@$-agreement).

Completeness

Completeness Theorem (Lemma 7.29 Blue Book):

Every $\mathbf{K}_h + \text{RULES}$-consistent set of formulas is satisfiable in a named model.

Moreover, if Π is a set of pure formulas, and $\mathbf{P}$ is the normal hybrid logic obtained by adding all the formulas in Π as extra axioms to $\mathbf{K}_h + \text{RULES}$, then every $\mathbf{P}$-consistent set of sentences is satisfiable in a named model based on a frame which validates every formula in Π.

$\mathbf{K}_h + \text{RULES}$ does not prove more validities than $\mathbf{K}_h$

But $\mathbf{K}_h + \text{RULES}$ does let us do something that $\mathbf{K}_h$ does not: it lets us infer all the consequences of a set of pure formulas Π.

Like a first-order theory

- In first-order logic there is one completeness theorem.
- This theorem tells us that we can infer all the consequences of any first-order theory Σ.
- This is how it works with $\mathbf{K}_h + \textsc{rules}$. There is one completeness theorem.
- This theorem tells us that we can infer all the consequences of any pure theory Π.
- This is not what happens in ordinary modal logic.
- This reflects what we discussed yesterday: pure validity is first-order, ordinary modal validity is second-order

Linking NAME and PASTE with tableaus

Obviously NAME and PASTE have done a lot of work for us in this proof. They helped us create exactly the kinds of MCSs inside the original MCS that we wanted.

In fact, both rules are closely related to the tableau systems we discussed yesterday.

Let us take a closer look . . .

The PASTE rule is a disguised sequent calculus rule

Here is PASTE:

$$\frac{\vdash @_i \Diamond j \wedge @_j \varphi \to \theta}{\vdash @_i \Diamond \varphi \to \theta}\ .$$

The PASTE rule is a disguised sequent calculus rule

Here is PASTE:

$$\frac{\vdash @_i\Diamond j \wedge @_j\varphi \to \theta}{\vdash @_i\Diamond\varphi \to \theta}\ .$$

Get rid of $\vdash$, and replace the implications by sequent arrows:

$$\frac{@_i\Diamond j \wedge @_j\varphi \longrightarrow \theta}{@_i\Diamond\varphi \longrightarrow \theta}\ .$$

The PASTE rule is a disguised sequent calculus rule

Here is PASTE:

$$\frac{\vdash @_i\Diamond j \wedge @_j\varphi \rightarrow \theta}{\vdash @_i\Diamond\varphi \rightarrow \theta} \quad .$$

Get rid of $\vdash$, and replace the implications by sequent arrows:

$$\frac{@_i\Diamond j \wedge @_j\varphi \longrightarrow \theta}{@_i\Diamond\varphi \longrightarrow \theta} \quad .$$

Splitting the top line formula into two simpler formulas yields:

$$\frac{@_i\Diamond j, @_j\varphi \longrightarrow \theta}{@_i\Diamond\varphi \longrightarrow \theta} \quad .$$

The PASTE rule is a disguised sequent calculus rule

Here is PASTE:

$$\frac{\vdash @_i\Diamond j \wedge @_j\varphi \rightarrow \theta}{\vdash @_i\Diamond\varphi \rightarrow \theta} \ .$$

Get rid of $\vdash$, and replace the implications by sequent arrows:

$$\frac{@_i\Diamond j \wedge @_j\varphi \longrightarrow \theta}{@_i\Diamond\varphi \longrightarrow \theta} \ .$$

Splitting the top line formula into two simpler formulas yields:

$$\frac{@_i\Diamond j, @_j\varphi \longrightarrow \theta}{@_i\Diamond\varphi \longrightarrow \theta} \ .$$

Add a left multiset Γ, and turn θ into a right multiset Θ:

$$\frac{@_i\Diamond j, @_j\varphi, \Gamma \longrightarrow \Theta}{@_i\Diamond\varphi, \Gamma \longrightarrow \Theta} \ .$$

This is a sequent rule. Read it bottom-to-top and it is essentially our most important tableau rule.

What about NAME?

Here is the NAME rule we used:

$$\frac{\vdash j \rightarrow \theta}{\vdash \theta}$$

In fact, this version of the NAME rule also works:

$$\frac{\vdash @_j \theta}{\vdash \theta}$$

This should remind you of something. When we want to prove a formula φ in the tableau system, we prefix it with $\neg @_j$, for some new nominal j, and then start applying rules. Basically, we start trying to build a model at the point named j.

In fact, you can prove that the NAME rule only has to be used once, and can always be used as the final proof step.

Some final comments

We can use this construction for many things: tableau completeness, completeness for first-order hybrid logic, interpolation. . .

We need NAME and PASTE, or something similar.

We build the model out of these (equivalence classes of) "witness nominals" instead (that is: instead of using the MCSs, just use the Robinson Diagram).

Also, it works for more than pure formulas. We can add "pure rules" as well, which give us a lot more results.

"Pure Extensions, Proof Rules, and Hybrid Axiomatics", Patrick Blackburn and Balder ten Cate, Studia Logica, 84(2), pp. 277-322, 2006.