Introduction to Hybrid Logic

Patrick Blackburn

Department of Philosophy, Roskilde University, Denmark

NASSLLI 2012, University of Austin, Texas

What we did yesterday...

What we did yesterday...

- We introduced the $\downarrow$ binder, a device which enables us to create names for nodes on the fly. In essence, $\downarrow$ allows us "to bind a nominal to the present". Yes, x really does mark the spot...

What we did yesterday...

- We introduced the $\downarrow$ binder, a device which enables us to create names for nodes on the fly. In essence, $\downarrow$ allows us "to bind a nominal to the present". Yes, x really does mark the spot...
- We saw that $\downarrow$ gave us (lots of) extra expressivity. One noteworthy feature of this expressivity is the way $\downarrow$ works in tandem with $@$: the $\downarrow$ binder stores, the $@$ operator retrieves. A common way of expressing interesting concepts with $\downarrow$ is first to give a name for the initial node. We then go out exploring with the diamonds, but can always find our way back with the help of $@$.

What we did yesterday...

- We introduced the $\downarrow$ binder, a device which enables us to create names for nodes on the fly. In essence, $\downarrow$ allows us "to bind a nominal to the present". Yes, x really does mark the spot...
- We saw that $\downarrow$ gave us (lots of) extra expressivity. One noteworthy feature of this expressivity is the way $\downarrow$ works in tandem with $@$: the $\downarrow$ binder stores, the $@$ operator retrieves. A common way of expressing interesting concepts with $\downarrow$ is first to give a name for the initial node. We then go out exploring with the diamonds, but can always find our way back with the help of $@$.
- We measured the expressivity. In essence, $\downarrow$ gives us "the logic of locality", or more precisely, the fragment of the correspondence language that is invariant under generated submodels.

What we did yesterday...

- We introduced the $\downarrow$ binder, a device which enables us to create names for nodes on the fly. In essence, $\downarrow$ allows us "to bind a nominal to the present". Yes, x really does mark the spot...
- We saw that $\downarrow$ gave us (lots of) extra expressivity. One noteworthy feature of this expressivity is the way $\downarrow$ works in tandem with @: the $\downarrow$ binder stores, the @ operator retrieves. A common way of expressing interesting concepts with $\downarrow$ is first to give a name for the initial node. We then go out exploring with the diamonds, but can always find our way back with the help of @.
- We measured the expressivity. In essence, $\downarrow$ gives us "the logic of locality", or more precisely, the fragment of the correspondence language that is invariant under generated submodels.
- And this frament is well behaved. Completeness is easy to obtain for any pure axiomatisation, and indeed, interpolation holds for any pure logic.

What we did yesterday...

- We introduced the $\downarrow$ binder, a device which enables us to create names for nodes on the fly. In essence, $\downarrow$ allows us "to bind a nominal to the present". Yes, x really does mark the spot...
- We saw that $\downarrow$ gave us (lots of) extra expressivity. One noteworthy feature of this expressivity is the way $\downarrow$ works in tandem with @: the $\downarrow$ binder stores, the @ operator retrieves. A common way of expressing interesting concepts with $\downarrow$ is first to give a name for the initial node. We then go out exploring with the diamonds, but can always find our way back with the help of @.
- We measured the expressivity. In essence, $\downarrow$ gives us "the logic of locality", or more precisely, the fragment of the correspondence language that is invariant under generated submodels.
- And this frament is well behaved. Completeness is easy to obtain for any pure axiomatisation, and indeed, interpolation holds for any pure logic.
- But we pay a price, losing both the finite model property and decidability.

Today: First-order hybrid logic

In today's lecture we shall:

- Informally discuss first-order modal logic and first-order hybrid logic via lots of examples.
- Define first-order hybrid logic more precisely, focusing on its syntax, semantics, and tableau rules. We'll also look at some of the things we can state in first-order hybrid logic.
- And $\downarrow$ really comes into its own.

First-order modal logic

What happens when modal states are inhabited by kings, queens, and presidents, and we want to talk about them using the full resources of first-order logic? That is, what happens if we upgrade the propositional language underlying our modal/hybrid languages to a full first-order language?

Well, semantically it's pretty easy. Intuitively, a first-order modal model consists of a Kripke model with a first order model (D, I) attached to every state.

The traditional syntactic choice is to blend of modal and first-order syntax to make statements about these models.

We are going to develop a blend of hybrid and first-order syntax which we feel is better suited for this purpose.

There are good reasons for doing this

Intuitive — this choice will enable us to express the sort of distinctions that are needed.

Technical — there are lots of problems in traditional first-order modal logic, and hybridization provides a way round them.

Definite descriptions

Consider the sentence

The queen of Holland will die.

With

- q a constant representing *the queen of Holland*
- D a unary predicate representing *to die*, and
- $\langle \mathrm{F} \rangle$ a diamond meaning *sometime in the future*

we can formalize the sentence as

$$\langle \mathrm{F} \rangle\, D(q).$$

The queen of Holland

It seems that the following argument is valid:

Everybody will die. There is a queen of Holland.
Thus
The queen of Holland will die.

The queen of Holland

It seems that the following argument is valid:

Everybody will die. There is a queen of Holland.
Thus
The queen of Holland will die.

$$\forall x \langle \mathrm{F} \rangle\, D(x),$$

The queen of Holland

It seems that the following argument is valid:

Everybody will die. There is a queen of Holland.
Thus
The queen of Holland will die.

$$\forall x \langle \mathrm{F} \rangle \, D(x), \;\; \exists x (x = q)$$

The queen of Holland

It seems that the following argument is valid:

Everybody will die. There is a queen of Holland.
Thus
The queen of Holland will die.

$$\forall x \langle \mathrm{F} \rangle D(x),\ \ \exists x(x = q) \ \models \ \langle \mathrm{F} \rangle D(q).$$

The queen of Holland

It seems that the following argument is valid:

Everybody will die. There is a queen of Holland.
Thus
The queen of Holland will die.

$$\forall x \langle \mathrm{F} \rangle\, D(x),\ \ \exists x (x = q) \ \models \ \langle \mathrm{F} \rangle\, D(q).$$

On the other hand, we can imagine that in Holland they are very good at anticipating death, and always make their queen abdicate before she dies.

The queen of Holland

It seems that the following argument is valid:

Everybody will die. There is a queen of Holland.
Thus
The queen of Holland will die.

$$\forall x \langle \mathrm{F} \rangle D(x), \;\; \exists x (x = q) \;\; \models \;\; \langle \mathrm{F} \rangle D(q).$$

On the other hand, we can imagine that in Holland they are very good at anticipating death, and always make their queen abdicate before she dies.
But in such a model $D(q)$ is always false. So $\langle \mathrm{F} \rangle D(q)$ is always false.

???

Ambiquity

The problem is this: "The Queen of Holland will die" is ambiguous. It has at least two different readings:

wide scope The present queen of Holland will die.

narrow scope Sometime in the future, the then present queen of Holland dies.

The term "the queen of Holland" contains a hidden reference to the time of evaluation. In English we can make this explicit using phrases like "present" and "then present".

Capturing the ambiguity in standard modal logic

You can capture the wide scope reading in orthodox modal logic. Here's how:

$$\exists x(x = q \wedge \langle \mathrm{F} \rangle\, D(x)).$$

But now compare this with the way we can handle this in hybrid logic . . .

Capturing the ambiguity in hybrid logic

Instead of formalizing *the queen of Holland* as a constant q, we shall formalize it as $@_s q$, for s a state variable.

Capturing the ambiguity in hybrid logic

Instead of formalizing *the queen of Holland* as a constant q, we shall formalize it as $@_s q$, for s a state variable.

Now we can distinguish the two readings in a way that makes the different temporal references they make explicit:

wide scope	$\downarrow s.\langle \mathrm{F} \rangle\, D(@_s q)$	*the now present*

Capturing the ambiguity in hybrid logic

Instead of formalizing *the queen of Holland* as a constant q, we shall formalize it as $@_s q$, for s a state variable.

Now we can distinguish the two readings in a way that makes the different temporal references they make explicit:

wide scope	$\downarrow s.\langle \mathrm{F} \rangle\, D(@_s q)$	*the now present*
narrow scope	$\langle \mathrm{F} \rangle \downarrow s.D(@_s q)$	*the then present*.

Different scopes yield different inferences

On the wide scope reading, the argument is valid:

$$\forall x \langle \mathrm{F} \rangle D(x),\ \exists x(x = q) \models\ \downarrow s. \langle \mathrm{F} \rangle D(@_s q).$$

Different scopes yield different inferences

On the wide scope reading, the argument is valid:

$$\forall x \langle \mathrm{F} \rangle\, D(x),\ \exists x(x = q) \models\ \downarrow s. \langle \mathrm{F} \rangle\, D(@_s q).$$

But on the narrow scope reading it is not:

$$\forall x \langle \mathrm{F} \rangle\, D(x),\ \exists x(x = q) \not\models\ \langle \mathrm{F} \rangle \downarrow s. D(@_s q).$$

The queen of Holland will be hip

This has at least two readings. Two of them are familiar.

The queen of Holland will be hip

This has at least two readings. Two of them are familiar.

1. $\downarrow s.\langle \mathrm{F} \rangle\, Hip(@_s q)$ *The present queen will be hip*

The queen of Holland will be hip

This has at least two readings. Two of them are familiar.

1. $\downarrow s.\langle \mathrm{F} \rangle\, \mathit{Hip}(@_s q)$ *The present queen will be hip*
2. $\langle \mathrm{F} \rangle \downarrow s.\mathit{Hip}(@_s q)$ *Holland will have a hip queen*

The queen of Holland will be hip

This has at least two readings. Two of them are familiar.

1. $\downarrow s.\langle \mathrm{F} \rangle\, Hip(@_s q)$ *The present queen will be hip*
2. $\langle \mathrm{F} \rangle \downarrow s.Hip(@_s q)$ *Holland will have a hip queen*

Wait — hipness is clearly a predicate which changes its meaning over time. So there is another ambiguity here

> *Will she be hip according to our standards of hipness now, or to the hipness criteria in the future?*

The second formula expresses the latter. Can we express the first?

With the help of $\downarrow$ and @ we can ...

Holland will have a hip queen according to todays standards of hipness.

$$\downarrow t.\langle \mathrm{F} \rangle \downarrow s.@_t Hip(@_s q)$$

With the help of $\downarrow$ and @ we can . . .

Holland will have a hip queen according to todays standards of hipness.

$$\downarrow t.\langle \mathrm{F} \rangle \downarrow s.@_t Hip(@_s q)$$

That is, we evaluate the hipness predicate at the present time, with respect to the individual who will then be the Queen of Holland.

More royals

In 1556 Charles V, Holy Roman Emperor, resigned his offices and spent the remainder of his life near the monastery of San Yuste in Spain repairing clocks. One can imagine Charles planning this episode in his life and musing:

Someday the emperor will not be the emperor.

More royals

In 1556 Charles V, Holy Roman Emperor, resigned his offices and spent the remainder of his life near the monastery of San Yuste in Spain repairing clocks. One can imagine Charles planning this episode in his life and musing:

Someday the emperor will not be the emperor.

He would have meant, of course,

$$\downarrow s.\langle \mathrm{F} \rangle \downarrow t.(@_s e \neq @_t e).$$

More royals

In 1556 Charles V, Holy Roman Emperor, resigned his offices and spent the remainder of his life near the monastery of San Yuste in Spain repairing clocks. One can imagine Charles planning this episode in his life and musing:

Someday the emperor will not be the emperor.

He would have meant, of course,

$$\downarrow s.\langle \mathrm{F} \rangle \downarrow t.(@_s e \neq @_t e).$$

and definitely not

$$\downarrow s \downarrow t.\langle \mathrm{F} \rangle (@_s e \neq @_t e).$$

Even more royals, but older ones

For a while it looked like the following statement would soon be true:

The queen mother is the oldest woman in England.

Even more royals, but older ones

For a while it looked like the following statement would soon be true:

The queen mother is the oldest woman in England.

Even if it was true, we don't want to derive from it that this will always be the case in the future. So the following formula should not be valid:

$$\not\models q = o \to [\mathrm{F}]\,(q = o)$$

Even more royals, but older ones

For a while it looked like the following statement would soon be true:

The queen mother is the oldest woman in England.

Even if it was true, we don't want to derive from it that this will always be the case in the future. So the following formula should not be valid:

$$\not\models q = o \to [\mathrm{F}]\,(q = o)$$

But this conflicts with Leibniz's Principle…

Leibniz's Principle

- Individuals are uniquely determined by their properties.

Leibniz's Principle

- Individuals are uniquely determined by their properties.
- Thus, if $q = o$ and $\phi(q)$, then $\phi(o)$.

Leibniz's Principle

- Individuals are uniquely determined by their properties.
- Thus, if $q = o$ and $\phi(q)$, then $\phi(o)$.
- But $q = q$ is a validity.

Leibniz's Principle

- Individuals are uniquely determined by their properties.
- Thus, if $q = o$ and $\phi(q)$, then $\phi(o)$.
- But $q = q$ is a validity. Hence $[\mathrm{F}]\,(q = q)$ is too.

Leibniz's Principle

- Individuals are uniquely determined by their properties.
- Thus, if $q = o$ and $\phi(q)$, then $\phi(o)$.
- But $q = q$ is a validity. Hence $[\mathrm{F}]\,(q = q)$ is too.
- So by Leibniz' Principle:
  $\models q = o \wedge [\mathrm{F}]\,(q = q) \rightarrow [\mathrm{F}]\,(q = o)$.

Leibniz's Principle

- Individuals are uniquely determined by their properties.
- Thus, if $q = o$ and $\phi(q)$, then $\phi(o)$.
- But $q = q$ is a validity. Hence $[\mathrm{F}]\,(q = q)$ is too.
- So by Leibniz' Principle:
  $\models q = o \wedge [\mathrm{F}]\,(q = q) \rightarrow [\mathrm{F}]\,(q = o)$.
- But this is equivalent to:
  $\models [\mathrm{F}]\,(q = q) \rightarrow (q = o \rightarrow [\mathrm{F}]\,(q = o)$.

Leibniz's Principle

- Individuals are uniquely determined by their properties.
- Thus, if $q = o$ and $\phi(q)$, then $\phi(o)$.
- But $q = q$ is a validity. Hence $[\mathrm{F}]\,(q = q)$ is too.
- So by Leibniz' Principle:
  $\models q = o \wedge [\mathrm{F}]\,(q = q) \rightarrow [\mathrm{F}]\,(q = o)$.
- But this is equivalent to:
  $\models [\mathrm{F}]\,(q = q) \rightarrow (q = o \rightarrow [\mathrm{F}]\,(q = o)$.
- Hence by modus ponens: $\models q = o \rightarrow [\mathrm{F}]\,(q = o)$.

It all has to do with *where* the variables are bound

The following *is* valid

> *The queen mother of 1–1–2000 is the eldest woman in England at 1–1–2000. Thus this will always be so.*

It all has to do with *where* the variables are bound

The following *is* valid

> *The queen mother of 1–1–2000 is the eldest woman in England at 1–1–2000.* *Thus* *this will always be so.*

In our formalism:

$$\models @_{1-1-2000}\, q = @_{1-1-2000}\, o \rightarrow [\mathrm{F}]\,(@_{1-1-2000}\, q = @_{1-1-2000}\, o).$$

It all has to do with *where* the variables are bound

The following *is* valid

> *The queen mother of 1–1–2000 is the eldest woman in England at 1–1–2000. Thus this will always be so.*

In our formalism:

$$\models @_{1-1-2000}q = @_{1-1-2000}o \rightarrow [\mathrm{F}](@_{1-1-2000}q = @_{1-1-2000}o).$$

So this is valid: $\downarrow s.(@_s q = @_s o \rightarrow [\mathrm{F}](@_s q = @_s o)$.

It all has to do with *where* the variables are bound

The following *is* valid

> *The queen mother of 1–1–2000 is the eldest woman in England at 1–1–2000. Thus this will always be so.*

In our formalism:

$$\models @_{1-1-2000}q = @_{1-1-2000}o \rightarrow [\mathrm{F}]\,(@_{1-1-2000}q = @_{1-1-2000}o).$$

So this is valid: $\downarrow s.(@_s q = @_s o \rightarrow [\mathrm{F}]\,(@_s q = @_s o)$.

But this not: $@_s q = @_s o \rightarrow [\mathrm{F}]{\downarrow}s.(@_s q = @_s o)$.

It all has to do with *where* the variables are bound

The following *is* valid

> *The queen mother of 1–1–2000 is the eldest woman in England at 1–1–2000. Thus this will always be so.*

In our formalism:

$$\models @_{1-1-2000}q = @_{1-1-2000}o \rightarrow [\mathrm{F}]\,(@_{1-1-2000}q = @_{1-1-2000}o).$$

So this is valid: $\downarrow s.(@_s q = @_s o \rightarrow [\mathrm{F}]\,(@_s q = @_s o)$.

But this not: $@_s q = @_s o \rightarrow [\mathrm{F}]{\downarrow} s.(@_s q = @_s o)$.

Compare with the first order: $\not\models f(x) = g(x) \rightarrow \forall x(f(x) = g(x))$.

$$\downarrow s.(@_s q = @_s o \rightarrow [\mathrm{F}]\, @_s q = @_s o)$$

Here's what (almost!) amounts to a tableau proof:

$\downarrow s.(@_s q = @_s o \to [\mathrm{F}]\, @_s q = @_s o)$

Here's what (almost!) amounts to a tableau proof:

$$1) \quad \neg @_i {\downarrow} s.(@_s q = @_s o \to [\mathrm{F}]\, (@_s q = @_s o))$$

$\downarrow s.(@_s q = @_s o \to [\mathrm{F}]\, @_s q = @_s o)$

Here's what (almost!) amounts to a tableau proof:

$$
\begin{array}{ll}
1) & \neg @_i {\downarrow} s.(@_s q = @_s o \to [\mathrm{F}]\,(@_s q = @_s o)) \\
2) & \neg @_i (@_i q = @_i o \to [\mathrm{F}]\,(@_i q = @_i o))
\end{array}
$$

$\downarrow s.(@_s q = @_s o \to [\mathrm{F}]\, @_s q = @_s o)$

Here's what (almost!) amounts to a tableau proof:

$$
\begin{array}{ll}
1) & \neg @_i {\downarrow} s.(@_s q = @_s o \to [\mathrm{F}]\,(@_s q = @_s o)) \\
2) & \neg @_i (@_i q = @_i o \to [\mathrm{F}]\,(@_i q = @_i o)) \\
3) & @_i (@_i q = @_i o) \\
3') & \neg @_i [\mathrm{F}]\,(@_i q = @_i o)
\end{array}
$$

$\downarrow s.(@_s q = @_s o \rightarrow [\mathrm{F}]\, @_s q = @_s o)$

Here's what (almost!) amounts to a tableau proof:

$$
\begin{array}{ll}
1) & \neg @_i {\downarrow} s.(@_s q = @_s o \rightarrow [\mathrm{F}]\,(@_s q = @_s o)) \\
2) & \neg @_i (@_i q = @_i o \rightarrow [\mathrm{F}]\,(@_i q = @_i o)) \\
3) & @_i (@_i q = @_i o) \\
3') & \neg @_i [\mathrm{F}]\,(@_i q = @_i o) \\
4) & @_i \langle \mathrm{F} \rangle\, j \\
4') & \neg @_j (@_i q = @_i o)
\end{array}
$$

$\downarrow s.(@_s q = @_s o \rightarrow [\mathrm{F}]\, @_s q = @_s o)$

Here's what (almost!) amounts to a tableau proof:

$$
\begin{array}{ll}
1) & \neg @_i \downarrow s.(@_s q = @_s o \rightarrow [\mathrm{F}]\,(@_s q = @_s o)) \\
2) & \neg @_i (@_i q = @_i o \rightarrow [\mathrm{F}]\,(@_i q = @_i o)) \\
3) & @_i (@_i q = @_i o) \\
3') & \neg @_i [\mathrm{F}]\,(@_i q = @_i o) \\
4) & @_i \langle \mathrm{F} \rangle\, j \\
4') & \neg @_j (@_i q = @_i o)
\end{array}
$$

But equality is a rigid predicate: it's meaning does not change over time. Hence 3 and 4' contradict each other. When we specify our tableau system we will have to include rules to make such contradictions explicit.

Wrap up

This concludes our informal discussion. Before turning to the technical development, let's sum up what we have learned:

- If we put an orthodox modal language together with a first order language together, then:
  - Some familiar and desired laws fail: e.g., $\not\models c{=}d \rightarrow [\mathrm{F}]\, c{=}d$.
  - Some readings of natural language sentences cannot be expressed.
- It also seems clear that we need a mechanism to deal with the hidden variables in terms:
  - The $\downarrow$, @ combination is a natural solution (especially if we let ourselves apply @ to terms).

Other solutions are possible

In particular, in "First-Order Modal Logic", by Fitting & Mendelsohn, Kluwer, 1998 (an excellent place to find out more about first-order modal logic) the use of λ abstraction over formulas is explored.

$$< \lambda x.\langle \mathrm{F} \rangle\, D(x) > (q) \qquad vs \qquad \langle \mathrm{F} \rangle < \lambda x.D(x) > (q)$$

$$\downarrow s.\langle \mathrm{F} \rangle\, D(@_s q) \qquad vs \qquad \langle \mathrm{F} \rangle \downarrow s.D(@_s q)$$

Of course, $\downarrow$ and @ are not new machinery bolted on to cope with problems at the first-order level: they were designed to solve problems at the propositional level. That they solve problems at the first-order level is a pleasant bonus, not their raison d'être.

Options for models

When we define our models we must make several design choices:

- **Domains** Constant or varying?
- **Terms** Partial designation or not?
- **Predicates** Range over locally existing or over all possible objects?

Models

A first-order modal model is a structure $\mathfrak{M} = (W, R, D, U, I)$ such that

Models

A first-order modal model is a structure $\mathfrak{M} = (W, R, D, U, I)$ such that

- (W, R) is a modal frame.

Models

A first-order modal model is a structure $\mathfrak{M} = (W, R, D, U, I)$ such that

- (W, R) is a modal frame.
- D assigns to each $w \in W$ a non-empty set, the domain of w.

Models

A first-order modal model is a structure $\mathfrak{M} = (W, R, D, U, I)$ such that

- (W, R) is a modal frame.
- D assigns to each $w \in W$ a non-empty set, the domain of w.
  Varying domains.

Models

A first-order modal model is a structure $\mathfrak{M} = (W, R, D, U, I)$ such that

- (W, R) is a modal frame.
- D assigns to each $w \in W$ a non-empty set, the domain of w. Varying domains.
- U is the union of the range of D; it's called the domain of the model.

Models

A first-order modal model is a structure $\mathfrak{M} = (W, R, D, U, I)$ such that

- (W, R) is a modal frame.
- D assigns to each $w \in W$ a non-empty set, the domain of w. Varying domains.
- U is the union of the range of D; it's called the domain of the model.
- I is an interpretation:

Models

A first-order modal model is a structure $\mathfrak{M} = (W, R, D, U, I)$ such that

- (W, R) is a modal frame.
- D assigns to each $w \in W$ a non-empty set, the domain of w. Varying domains.
- U is the union of the range of D; it's called the domain of the model.
- I is an interpretation:
  - It assigns to each constant symbol c and to each $w \in W$, an element in U.

Models

A first-order modal model is a structure $\mathfrak{M} = (W, R, D, U, I)$ such that

- (W, R) is a modal frame.
- D assigns to each $w \in W$ a non-empty set, the domain of w. Varying domains.
- U is the union of the range of D; it's called the domain of the model.
- I is an interpretation:
    - It assigns to each constant symbol c and to each $w \in W$, an element in U. Terms designate in every world.

Models

A first-order modal model is a structure $\mathfrak{M} = (W, R, D, U, I)$ such that

- (W, R) is a modal frame.
- D assigns to each $w \in W$ a non-empty set, the domain of w. Varying domains.
- U is the union of the range of D; it's called the domain of the model.
- I is an interpretation:
  - It assigns to each constant symbol c and to each $w \in W$, an element in U. Terms designate in every world.
  - It assigns to each n-place relation symbol R and to each $w \in W$, some n-place relation on U.

Models

A first-order modal model is a structure $\mathfrak{M} = (W, R, D, U, I)$ such that

- (W, R) is a modal frame.
- D assigns to each $w \in W$ a non-empty set, the domain of w. Varying domains.
- U is the union of the range of D; it's called the domain of the model.
- I is an interpretation:
  - It assigns to each constant symbol c and to each $w \in W$, an element in U. Terms designate in every world.
  - It assigns to each n-place relation symbol R and to each $w \in W$, some n-place relation on U. Predicates range over possible objects, so "Napoleon is a man" is ok.

Models

A first-order modal model is a structure $\mathfrak{M} = (W, R, D, U, I)$ such that

- (W, R) is a modal frame.
- D assigns to each $w \in W$ a non-empty set, the domain of w. Varying domains.
- U is the union of the range of D; it's called the domain of the model.
- I is an interpretation:
  - It assigns to each constant symbol c and to each $w \in W$, an element in U. Terms designate in every world.
  - It assigns to each n-place relation symbol R and to each $w \in W$, some n-place relation on U. Predicates range over possible objects, so "Napoleon is a man" is ok.
  - It also assigns appropriate values to propositional symbols and nominals, for we can think of these as 0-place relation symbols

Formulas

A Quantified hybrid logic is obtained by combining a standard first-order language with a language of hybrid logic (with $\downarrow$). The only change is in the treatment of constants: the only terms we have are

- Ordinary first order variables
- If w is a nominal or state variable and c a constant, then $@_w c$ is a term.

In orthodox first-order modal logic it is typically assumed that all constants are rigid. We take the reverse approach: all constants are non-rigid, and must be explicitly rigidified using $@$.

Evaluating formulas

The satisfaction definition has the form one would expect: we define

$$\mathfrak{M}, w, g, v \models \phi$$

> *ϕ is true in model $\mathfrak{M}$ at world w under the assignment g to the state variables and the assignment v to the first order variables.*

- For atomic formulas the truth definition is provided by I.
- The quantifiers range only over the locally existing individuals.

Denotations of terms

Let $\mathfrak{M} = (W, R, D, U, I)$ be a model, let g be an assignment to state variables in $\mathfrak{M}$, let v be an assignment to first-order variables in $\mathfrak{M}$. As usual, the denotation of a state variable or a nominal in

$\mathfrak{M}$ is the unique state where it is true (g provides this information for state variables, I provides it for nominals).

Let τ be any term. Then $[g, I, v](\tau)$, the denotation of τ with respect to $\mathfrak{M}$, g, and v is defined as follows:

$$
\begin{aligned}
[g, I, v](x) &= v(x), \\
[g, I, v](@_w c) &= I(c, u), \text{ where } u \text{ is the denotation of } w.
\end{aligned}
$$

Sample satisfaction clauses

Sample satisfaction clauses

1. $\mathfrak{M}, w, g, v \models P\tau_1 \ldots \tau_n$ iff
   $([g, I, v](\tau_1), \ldots, [g, I, v](\tau_n)) \in I_w(P)$,

Sample satisfaction clauses

1. $\mathfrak{M}, w, g, v \models P\tau_1 \dots \tau_n$ iff $([g, I, v](\tau_1), \dots, [g, I, v](\tau_n)) \in I_w(P)$,
2. $\mathfrak{M}, w, g, v \models \tau_1 = \tau_2$ iff $[g, I, v](\tau_1) = [g, I, v](\tau_2)$,

Sample satisfaction clauses

1. $\mathfrak{M}, w, g, v \models P\tau_1 \ldots \tau_n$ iff $([g, I, v](\tau_1), \ldots, [g, I, v](\tau_n)) \in I_w(P)$,
2. $\mathfrak{M}, w, g, v \models \tau_1 = \tau_2$ iff $[g, I, v](\tau_1) = [g, I, v](\tau_2)$,
3. $\mathfrak{M}, w, g, v \models \neg\phi$ iff $\mathfrak{M}, w, g, v \not\models \phi$,

Sample satisfaction clauses

1. $\mathfrak{M}, w, g, v \models P\tau_1 \ldots \tau_n$ iff $([g, I, v](\tau_1), \ldots, [g, I, v](\tau_n)) \in I_w(P)$,
2. $\mathfrak{M}, w, g, v \models \tau_1 = \tau_2$ iff $[g, I, v](\tau_1) = [g, I, v](\tau_2)$,
3. $\mathfrak{M}, w, g, v \models \neg\phi$ iff $\mathfrak{M}, w, g, v \not\models \phi$,
4. $\mathfrak{M}, w, g, v \models \phi \wedge \psi$ iff $\mathfrak{M}, w, g, v \models \phi$ and $\mathfrak{M}, w, g, v \models \psi$,

Sample satisfaction clauses

1. $\mathfrak{M}, w, g, v \models P\tau_1 \ldots \tau_n$ iff $([g, I, v](\tau_1), \ldots, [g, I, v](\tau_n)) \in I_w(P)$,
2. $\mathfrak{M}, w, g, v \models \tau_1 = \tau_2$ iff $[g, I, v](\tau_1) = [g, I, v](\tau_2)$,
3. $\mathfrak{M}, w, g, v \models \neg\phi$ iff $\mathfrak{M}, w, g, v \not\models \phi$,
4. $\mathfrak{M}, w, g, v \models \phi \wedge \psi$ iff $\mathfrak{M}, w, g, v \models \phi$ and $\mathfrak{M}, w, g, v \models \psi$,
5. $\mathfrak{M}, w, g, v \models \Diamond\phi$ iff there exists a $w' \in W$ such that wRw' and $\mathfrak{M}, w', g, v \models \phi$,

Sample satisfaction clauses

1. $\mathfrak{M}, w, g, v \models P\tau_1 \dots \tau_n$ iff $([g, I, v](\tau_1), \dots, [g, I, v](\tau_n)) \in I_w(P)$,
2. $\mathfrak{M}, w, g, v \models \tau_1 = \tau_2$ iff $[g, I, v](\tau_1) = [g, I, v](\tau_2)$,
3. $\mathfrak{M}, w, g, v \models \neg\phi$ iff $\mathfrak{M}, w, g, v \not\models \phi$,
4. $\mathfrak{M}, w, g, v \models \phi \wedge \psi$ iff $\mathfrak{M}, w, g, v \models \phi$ and $\mathfrak{M}, w, g, v \models \psi$,
5. $\mathfrak{M}, w, g, v \models \Diamond\phi$ iff there exists a $w' \in W$ such that wRw' and $\mathfrak{M}, w', g, v \models \phi$,
6. $\mathfrak{M}, w, g, v \models \exists x\phi$ iff there exists an assignment v' different from v at most in that $v'(x) \neq v(x)$, with $v'(x) \in D_w$ and $\mathfrak{M}, w, g, v' \models \phi$.

Example: asserting existence

Claim:

A constant c denotes something that exists at the state t
if and only if
$\mathfrak{M}, t \Vdash \downarrow s.\exists x(x = @_s c)$.

Example: asserting existence

Claim:

A constant c denotes something that exists at the state t
if and only if
$$\mathfrak{M}, t \Vdash {\downarrow} s.\exists x (x = @_s c).$$

Proof:
$\mathfrak{M}, t \Vdash {\downarrow} s.\exists x (x = @_s c)$ iff

Example: asserting existence

Claim:

A constant c denotes something that exists at the state t
if and only if
$\mathfrak{M}, t \Vdash \downarrow s.\exists x(x = @_s c)$.

Proof:
$\mathfrak{M}, t \Vdash \downarrow s.\exists x(x = @_s c)$ iff
$\mathfrak{M}, t \Vdash \exists x(x = @_i c)$ iff

Example: asserting existence

Claim:

A constant c denotes something that exists at the state t
if and only if
$$\mathfrak{M}, t \Vdash \downarrow s.\exists x(x = @_s c).$$

Proof:
$\mathfrak{M}, t \Vdash \downarrow s.\exists x(x = @_s c)$ iff
$\mathfrak{M}, t \Vdash \exists x(x = @_i c)$ iff
there exists a $d \in D(t)$ such that $I(t, c) = d$.

Example: rigid terms

Claim: We can express that a constant is rigid in models in which there is a path between every two worlds.

$$c \text{ designates the same individual in every world in } \mathfrak{M}$$
$$\text{if and only if}$$
$$\mathfrak{M} \models \downarrow s.[\mathrm{F}]\downarrow t.(@_s c = @_t c).$$

If we know a term is rigid, we write just c instead of $@_i c$.

Proof, $\Uparrow$

Proof, $\Uparrow$

Suppose c is not rigid. Then for some i, j, $I(i, c) \neq I(j, c)$ and $R(i, j)$. (Walk along the path that exists between two points that give different denotations to c. You will find two R-related points on this path that give different denotations to c.)

Proof, $\Uparrow$

Suppose c is not rigid. Then for some i, j, $I(i, c) \neq I(j, c)$ and $R(i, j)$. (Walk along the path that exists between two points that give different denotations to c. You will find two R-related points on this path that give different denotations to c.)
Then $\mathfrak{M}, j \Vdash @_i c \neq @_j c$.

Proof, $\Uparrow$

Suppose c is not rigid. Then for some i, j, $I(i, c) \neq I(j, c)$ and $R(i, j)$. (Walk along the path that exists between two points that give different denotations to c. You will find two R-related points on this path that give different denotations to c.)
Then $\mathfrak{M}, j \Vdash @_i c \neq @_j c$.
So, $\mathfrak{M}, j \Vdash \downarrow t. @_i c \neq @_t c$.

Proof, $\Uparrow$

Suppose c is not rigid. Then for some i, j, $I(i, c) \neq I(j, c)$ and $R(i, j)$. (Walk along the path that exists between two points that give different denotations to c. You will find two R-related points on this path that give different denotations to c.)
Then $\mathfrak{M}, j \Vdash @_i c \neq @_j c$.
So, $\mathfrak{M}, j \Vdash \downarrow t.@_i c \neq @_t c$.
So, because $R(i, j)$, $\mathfrak{M}, i \Vdash \langle \mathrm{F} \rangle \downarrow t.(@_i c \neq @_t c)$.

Proof, $\Uparrow$

Suppose c is not rigid. Then for some i, j, $I(i, c) \neq I(j, c)$ and $R(i, j)$. (Walk along the path that exists between two points that give different denotations to c. You will find two R-related points on this path that give different denotations to c.)

Then $\mathfrak{M}, j \Vdash @_i c \neq @_j c$.

So, $\mathfrak{M}, j \Vdash \downarrow t.@_i c \neq @_t c$.

So, because $R(i, j)$, $\mathfrak{M}, i \Vdash \langle \mathrm{F} \rangle \downarrow t.(@_i c \neq @_t c)$.

So, $\mathfrak{M}, i \Vdash \downarrow s.\langle \mathrm{F} \rangle \downarrow t.(@_s c \neq @_t c)$.

Proof, $\Uparrow$

Suppose c is not rigid. Then for some i, j, $I(i,c) \neq I(j,c)$ and $R(i,j)$. (Walk along the path that exists between two points that give different denotations to c. You will find two R-related points on this path that give different denotations to c.)

Then $\mathfrak{M}, j \Vdash @_i c \neq @_j c$.

So, $\mathfrak{M}, j \Vdash \downarrow t.@_i c \neq @_t c$.

So, because $R(i,j)$, $\mathfrak{M}, i \Vdash \langle \mathrm{F} \rangle \downarrow t.(@_i c \neq @_t c)$.

So, $\mathfrak{M}, i \Vdash \downarrow s.\langle \mathrm{F} \rangle \downarrow t.(@_s c \neq @_t c)$.

Thus $\mathfrak{M}, i \nVdash \downarrow s.[\mathrm{F}] \downarrow t.(@_s c = @_t c)$.

Proof, $\Downarrow$

Suppose that for some i, j, $\mathfrak{M}, i \not\Vdash {\downarrow} s.[\mathrm{F}]{\downarrow} t.(@_s c = @_t c)$.

Proof, $\Downarrow$

Suppose that for some i, j, $\mathfrak{M}, i \nVdash {\downarrow} s.[\mathrm{F}]{\downarrow} t.(@_s c = @_t c)$.
Then $\mathfrak{M}, i \Vdash \neg {\downarrow} s.[\mathrm{F}]{\downarrow}.(@_s c = @_t c)$.

Proof, $\Downarrow$

Suppose that for some i, j, $\mathfrak{M}, i \nVdash {\downarrow} s.[\mathrm{F}]{\downarrow} t.(@_s c = @_t c)$.
Then $\mathfrak{M}, i \Vdash \neg {\downarrow} s.[\mathrm{F}]{\downarrow}.(@_s c = @_t c)$.
Then $\mathfrak{M}, i \Vdash \neg[\mathrm{F}]{\downarrow} t.(@_i c = @_t c)$.

Proof, $\Downarrow$

Suppose that for some i, j, $\mathfrak{M}, i \nVdash \downarrow s.[\mathrm{F}]\downarrow t.(@_s c = @_t c)$.
Then $\mathfrak{M}, i \Vdash \neg \downarrow s.[\mathrm{F}]\downarrow .(@_s c = @_t c)$.
Then $\mathfrak{M}, i \Vdash \neg[\mathrm{F}]\downarrow t.(@_i c = @_t c)$.
Then there exists a j such that $R(i, j)$ and
$\mathfrak{M}, j \Vdash \neg \downarrow t.(@_i c = @_t c)$.

Proof, $\Downarrow$

Suppose that for some i, j, $\mathfrak{M}, i \not\Vdash \downarrow s.[\mathrm{F}]\downarrow t.(@_s c = @_t c)$.
Then $\mathfrak{M}, i \Vdash \neg \downarrow s.[\mathrm{F}]\downarrow .(@_s c = @_t c)$.
Then $\mathfrak{M}, i \Vdash \neg[\mathrm{F}]\downarrow t.(@_i c = @_t c)$.
Then there exists a j such that $R(i, j)$ and
$\mathfrak{M}, j \Vdash \neg \downarrow t.(@_i c = @_t c)$.
Then $\mathfrak{M}, j \Vdash \neg(@_i c = @_j c)$.

Proof, $\Downarrow$

Suppose that for some i, j, $\mathfrak{M}, i \nVdash {\downarrow} s.[\mathrm{F}]{\downarrow} t.(@_s c = @_t c)$.
Then $\mathfrak{M}, i \Vdash \neg {\downarrow} s.[\mathrm{F}]{\downarrow}.(@_s c = @_t c)$.
Then $\mathfrak{M}, i \Vdash \neg [\mathrm{F}]{\downarrow} t.(@_i c = @_t c)$.
Then there exists a j such that $R(i, j)$ and
$\mathfrak{M}, j \Vdash \neg {\downarrow} t.(@_i c = @_t c)$.
Then $\mathfrak{M}, j \Vdash \neg(@_i c = @_j c)$.
But this means that c is not rigid.

Tableau Calculus

The calculus consists of all rules for the propositional hybrid logic with downarrow plus rules for the quantifiers and rules for equality.

Tableau Calculus

The calculus consists of all rules for the propositional hybrid logic with downarrow plus rules for the quantifiers and rules for equality. As we have varying domains, the rules for the quantifiers must take existence of individuals into account.

Tableau Calculus

The calculus consists of all rules for the propositional hybrid logic with downarrow plus rules for the quantifiers and rules for equality. As we have varying domains, the rules for the quantifiers must take existence of individuals into account.
As you will see, this gives the quantifier rules the same format as the diamond and box rules.

Quantifier rules

For c a constant, abbreviate $@_i \exists x (x = @_i c)$ by $\mathsf{E}_i(c)$. Read this as "c exists at state i".

Existential rule

$$\frac{@_i \exists x \phi}{\begin{array}{c} \mathsf{E}_i(c) \\ @_i \phi[x \leftarrow @_i c] \end{array}} \qquad \text{for } c \text{ a new constant symbol}$$

Quantifier rules

For c a constant, abbreviate $@_i \exists x(x = @_i c)$ by $\mathrm{E}_i(c)$. Read this as "c exists at state i".

Existential rule

$$\frac{@_i \exists x \phi}{\mathrm{E}_i(c)} \qquad \text{for } c \text{ a new constant symbol}$$

$$@_i \phi[x \leftarrow @_i c]$$

Universal Rule

$$\frac{@_i \forall x \phi \qquad \mathrm{E}_i(c)}{@_i \phi[x \leftarrow @_i c]}$$

Note the analogy of form with the diamond and box rules.

Equality rules

Equality rules

Equality is a rigid predicate.

$$\frac{@_i(@_j c = @_k d)}{@_j c = @_k d} \qquad \frac{\neg @_i(@_j c = @_k d)}{@_j c \neq @_k d}$$

Equality rules

Equality is a rigid predicate.

$$\frac{@_i(@_j c = @_k d)}{@_j c = @_k d} \qquad \frac{\neg @_i(@_j c = @_k d)}{@_j c \neq @_k d}$$

Equality of worlds implies equality of terms

$$\frac{@_i j}{@_i c = @_j c}$$

Equality rules

Equality is a rigid predicate.

$$\frac{@_i(@_j c = @_k d)}{@_j c = @_k d} \qquad \frac{\neg @_i(@_j c = @_k d)}{@_j c \neq @_k d}$$

Equality of worlds implies equality of terms

$$\frac{@_i j}{@_i c = @_j c}$$

Reflexivity

$$\frac{}{@_i c = @_i c}$$

Equality rules

Replacement

$$\frac{@_i c = @_j d \qquad \phi(@_i c)}{\phi(@_j d)}$$

Example derivation: Abraham Lincoln

Adapted from "First-Order Modal Logic", Fitting & Mendelsohn, Kluwer, 1998.

Let p be a constant standing for *the president of the United States*.

Let a be a constant denoting *Abraham Lincoln*.

Let $Tall(x)$ be *x is Tall*.

Example derivation: Abraham Lincoln

Adapted from "First-Order Modal Logic", Fitting & Mendelsohn, Kluwer, 1998.

Let p be a constant standing for *the president of the United States*.

Let a be a constant denoting *Abraham Lincoln*.

Let $Tall(x)$ be x *is Tall*.

It is natural to consider a to be a *rigid* constant, so we'll write a instead of $@_i a$.

Back in about 1850. . .

Back in about 1850 could make the following true statements:

Back in about 1850...

Back in about 1850 could make the following true statements:

- $\langle \mathrm{F} \rangle {\downarrow} s.@_s p = a$. Abraham Lincoln will someday be the president of the USA

Back in about 1850...

Back in about 1850 could make the following true statements:

- $\langle \mathrm{F} \rangle \downarrow s.@_s p = a$. Abraham Lincoln will someday be the president of the USA
- $[\mathrm{F}]\, \mathit{Tall}(a)$. It will always be true of Abraham Lincoln that he is tall.

Back in about 1850...

Back in about 1850 could make the following true statements:

- $\langle \mathrm{F} \rangle \downarrow s.@_s p = a$. Abraham Lincoln will someday be the president of the USA
- $[\mathrm{F}]\, \mathit{Tall}(a)$. It will always be true of Abraham Lincoln that he is tall.
- $\langle \mathrm{F} \rangle \downarrow s.\mathit{Tall}(@_s p)$. It will happen that the president of the United States is tall.

$\models (\langle \mathrm{F} \rangle {\downarrow} s.(@_s p = a) \wedge [\mathrm{F}]\, \mathit{Tall}(a)) \rightarrow \langle \mathrm{F} \rangle {\downarrow} s.\mathit{Tall}(@_s p)$

$\models (\langle F \rangle {\downarrow} s.(@_s p = a) \wedge [F]\, Tall(a)) \rightarrow \langle F \rangle {\downarrow} s.Tall(@_s p)$

1) $@_i \langle F \rangle {\downarrow} s.(@_s p = a)$
1′) $@_i [F]\, Tall(a)$
1″) $\neg @_i \langle F \rangle {\downarrow} s.Tall(@_s p)$

$$\models (\langle \mathrm{F} \rangle {\downarrow} s.(@_s p = a) \wedge [\mathrm{F}]\, \mathit{Tall}(a)) \rightarrow \langle \mathrm{F} \rangle {\downarrow} s.\mathit{Tall}(@_s p)$$

1)	$@_i \langle \mathrm{F} \rangle {\downarrow} s.(@_s p = a)$	
1′)	$@_i [\mathrm{F}]\, \mathit{Tall}(a)$	
1″)	$\neg @_i \langle \mathrm{F} \rangle {\downarrow} s.\mathit{Tall}(@_s p)$	
2)	$@_i \langle \mathrm{F} \rangle\, j$	
2′)	$@_j {\downarrow} s.(@_s p = a)$	Diamond rule on 1

$\models (\langle F\rangle{\downarrow}s.(@_s p = a) \wedge [F]\, Tall(a)) \rightarrow \langle F\rangle{\downarrow}s.Tall(@_s p)$

1)	$@_i\langle F\rangle{\downarrow}s.(@_s p = a)$	
1′)	$@_i[F]\, Tall(a)$	
1″)	$\neg @_i\langle F\rangle{\downarrow}s.Tall(@_s p)$	
2)	$@_i\langle F\rangle j$	
2′)	$@_j{\downarrow}s.(@_s p = a)$	Diamond rule on 1
3)	$@_j(@_j p = a)$	$\downarrow$ rule on 2′

$\models (\langle F \rangle {\downarrow} s.(@_s p = a) \wedge [F]\, Tall(a)) \rightarrow \langle F \rangle {\downarrow} s. Tall(@_s p)$

1)	$@_i \langle F \rangle {\downarrow} s.(@_s p = a)$	
1′)	$@_i [F]\, Tall(a)$	
1″)	$\neg @_i \langle F \rangle {\downarrow} s. Tall(@_s p)$	
2)	$@_i \langle F \rangle j$	
2′)	$@_j {\downarrow} s.(@_s p = a)$	Diamond rule on 1
3)	$@_j (@_j p = a)$	$\downarrow$ rule on 2′
4)	$@_j p = a$	Rigidity rule on 3

$$\models (\langle F\rangle {\downarrow} s.(@_s p = a) \wedge [F]\, Tall(a)) \rightarrow \langle F\rangle {\downarrow} s.Tall(@_s p)$$

1)	$@_i\langle F\rangle {\downarrow} s.(@_s p = a)$	
1′)	$@_i[F]\, Tall(a)$	
1″)	$\neg @_i\langle F\rangle {\downarrow} s.Tall(@_s p)$	
2)	$@_i\langle F\rangle j$	
2′)	$@_j{\downarrow} s.(@_s p = a)$	Diamond rule on 1
3)	$@_j(@_j p = a)$	${\downarrow}$ rule on $2'$
4)	$@_j p = a$	Rigidity rule on 3
5)	$@_j Tall(a)$	Box rule on $1', 2$

$$\models (\langle \mathrm{F} \rangle {\downarrow} s.(@_s p = a) \wedge [\mathrm{F}]\, \mathit{Tall}(a)) \rightarrow \langle \mathrm{F} \rangle {\downarrow} s.\mathit{Tall}(@_s p)$$

1)	$@_i \langle \mathrm{F} \rangle {\downarrow} s.(@_s p = a)$	
1′)	$@_i [\mathrm{F}]\, \mathit{Tall}(a)$	
1″)	$\neg @_i \langle \mathrm{F} \rangle {\downarrow} s.\mathit{Tall}(@_s p)$	
2)	$@_i \langle \mathrm{F} \rangle j$	
2′)	$@_j {\downarrow} s.(@_s p = a)$	Diamond rule on 1
3)	$@_j (@_j p = a)$	$\downarrow$ rule on $2'$
4)	$@_j p = a$	Rigidity rule on 3
5)	$@_j \mathit{Tall}(a)$	Box rule on $1', 2$
6)	$@_j \mathit{Tall}(@_j p)$	Replacement on 4,5

$\models (\langle F\rangle{\downarrow}s.(@_s p = a) \wedge [F]\, Tall(a)) \rightarrow \langle F\rangle{\downarrow}s.Tall(@_s p)$

1)	$@_i\langle F\rangle{\downarrow}s.(@_s p = a)$	
1′)	$@_i[F]\, Tall(a)$	
1″)	$\neg @_i\langle F\rangle{\downarrow}s.Tall(@_s p)$	
2)	$@_i\langle F\rangle j$	
2′)	$@_j{\downarrow}s.(@_s p = a)$	Diamond rule on 1
3)	$@_j(@_j p = a)$	$\downarrow$ rule on $2'$
4)	$@_j p = a$	Rigidity rule on 3
5)	$@_j Tall(a)$	Box rule on $1', 2$
6)	$@_j Tall(@_j p)$	Replacement on 4,5
7)	$\neg @_j{\downarrow}s.Tall(@_s p)$	Box rule on $1'', 2$

$$\models (\langle \mathrm{F} \rangle {\downarrow} s.(@_s p = a) \wedge [\mathrm{F}]\, \mathit{Tall}(a)) \to \langle \mathrm{F} \rangle {\downarrow} s.\mathit{Tall}(@_s p)$$

1)	$@_i \langle \mathrm{F} \rangle {\downarrow} s.(@_s p = a)$	
1′)	$@_i [\mathrm{F}]\, \mathit{Tall}(a)$	
1″)	$\neg @_i \langle \mathrm{F} \rangle {\downarrow} s.\mathit{Tall}(@_s p)$	
2)	$@_i \langle \mathrm{F} \rangle\, j$	
2′)	$@_j {\downarrow} s.(@_s p = a)$	Diamond rule on 1
3)	$@_j(@_j p = a)$	$\downarrow$ rule on $2'$
4)	$@_j p = a$	Rigidity rule on 3
5)	$@_j \mathit{Tall}(a)$	Box rule on $1', 2$
6)	$@_j \mathit{Tall}(@_j p)$	Replacement on 4,5
7)	$\neg @_j {\downarrow} s.\mathit{Tall}(@_s p)$	Box rule on $1'', 2$
8)	$\neg @_j \mathit{Tall}(@_j p)$	$\downarrow$ rule on 7

$$\models (\langle \mathrm{F} \rangle {\downarrow} s.(@_s p = a) \wedge [\mathrm{F}]\, \mathit{Tall}(a)) \rightarrow \langle \mathrm{F} \rangle {\downarrow} s.\mathit{Tall}(@_s p)$$

1)	$@_i \langle \mathrm{F} \rangle {\downarrow} s.(@_s p = a)$	
1′)	$@_i [\mathrm{F}]\, \mathit{Tall}(a)$	
1″)	$\neg @_i \langle \mathrm{F} \rangle {\downarrow} s.\mathit{Tall}(@_s p)$	
2)	$@_i \langle \mathrm{F} \rangle j$	
2′)	$@_j {\downarrow} s.(@_s p = a)$	Diamond rule on 1
3)	$@_j (@_j p = a)$	$\downarrow$ rule on $2'$
4)	$@_j p = a$	Rigidity rule on 3
5)	$@_j \mathit{Tall}(a)$	Box rule on $1', 2$
6)	$@_j \mathit{Tall}(@_j p)$	Replacement on 4,5
7)	$\neg @_j {\downarrow} s.\mathit{Tall}(@_s p)$	Box rule on $1'', 2$
8)	$\neg @_j \mathit{Tall}(@_j p)$	$\downarrow$ rule on 7
	$\perp_{6,8}$	

A dynamic logic example

Adapted from "First-Order Modal Logic", Fitting & Mendelsohn, Kluwer, 1998.

- The Kripke structures model the states of a machine doing a computation.
- Every state in the model corresponds to a state of the machine. The relation in the model corresponds to the performance of a computation step.
- In our example, states give the integer values of program variables, and the computation step is *adding one*
- We will prove that program variables are not rigid. (Our constants are the program variables).

What we will prove:

We denote the program variables by constants c. The value of a program variable c in a computation state s is then $@_s c$.

What we will prove:

We denote the program variables by constants c. The value of a program variable c in a computation state s is then $@_s c$.

We give a tableau proof of the following statement:

If $\quad \downarrow s.\exists x(x = @_s c) \quad$ c is initialized

What we will prove:

We denote the program variables by constants c. The value of a program variable c in a computation state s is then $@_s c$.

We give a tableau proof of the following statement:

If	$\downarrow s.\exists x(x = @_s c)$	c is initialized
and	$\downarrow s.\forall x(@_s x \neq @_s(x+1))$	an eternal truth of math

What we will prove:

We denote the program variables by constants c. The value of a program variable c in a computation state s is then $@_s c$.

We give a tableau proof of the following statement:

If	$\downarrow s.\exists x(x = @_s c)$	c is initialized
and	$\downarrow s.\forall x(@_s x \neq @_s(x+1))$	an eternal truth of math
and	$\downarrow s.\Box \downarrow t.(@_t c = @_s(c+1))$	this is what $\Box$ does, adding one

What we will prove:

We denote the program variables by constants c. The value of a program variable c in a computation state s is then $@_s c$.

We give a tableau proof of the following statement:

If	$\downarrow s.\exists x(x = @_s c)$	c is initialized
and	$\downarrow s.\forall x(@_s x \neq @_s(x+1))$	an eternal truth of math
and	$\downarrow s.\Box \downarrow t.(@_t c = @_s(c+1))$	this is what $\Box$ does, adding one
and	$\Diamond\top$	the program can be executed

What we will prove:

We denote the program variables by constants c. The value of a program variable c in a computation state s is then $@_s c$.

We give a tableau proof of the following statement:

If	$\downarrow s.\exists x(x = @_s c)$	c is initialized
and	$\downarrow s.\forall x(@_s x \neq @_s(x+1))$	an eternal truth of math
and	$\downarrow s.\Box \downarrow t.(@_t c = @_s(c+1))$	this is what $\Box$ does, adding one
and	$\Diamond \top$	the program can be executed
then	$\neg \downarrow s.\Box \downarrow t.(@_s c = @_t c)$	c is not rigid.

The tableau proof

1.	$@_i(\downarrow s.\exists x(x = @_s c))$	Assumption 1
2.	$@_i(\downarrow s.\forall x(@_s x \neq @_s(x+1)))$	Assumption 2
3.	$@_i(\downarrow s.\Box \downarrow t.(@_t c = @_s(c+1)))$	Assumption 3
4.	$@_i(\Diamond \top)$	Assumption 4
5.	$\neg @_i(\neg \downarrow s.\Box \downarrow t.(@_s c = @_t c))$	Negated conclusion

The tableau proof

1.	$@_i(\downarrow s.\exists x(x = @_s c))$	Assumption 1
2.	$@_i(\downarrow s.\forall x(@_s x \neq @_s(x+1)))$	Assumption 2
3.	$@_i(\downarrow s.\Box \downarrow t.(@_t c = @_s(c+1)))$	Assumption 3
4.	$@_i(\Diamond \top)$	Assumption 4
5.	$\neg @_i(\neg \downarrow s.\Box \downarrow t.(@_s c = @_t c))$	Negated conclusion

First we derive that $@_i c \neq @_i(c+1)$:

The tableau proof

1.	$@_i(\downarrow s.\exists x(x = @_s c))$	Assumption 1
2.	$@_i(\downarrow s.\forall x(@_s x \neq @_s(x+1)))$	Assumption 2
3.	$@_i(\downarrow s.\Box \downarrow t.(@_t c = @_s(c+1)))$	Assumption 3
4.	$@_i(\Diamond\top)$	Assumption 4
5.	$\neg @_i(\neg \downarrow s.\Box \downarrow t.(@_s c = @_t c))$	Negated conclusion

First we derive that $@_i c \neq @_i(c+1)$:

6.	$@_i \exists x(x = @_i c)$	$\downarrow$ rule on 1.

The tableau proof

1.	$@_i(\downarrow s.\exists x(x = @_s c))$	Assumption 1
2.	$@_i(\downarrow s.\forall x(@_s x \neq @_s(x+1)))$	Assumption 2
3.	$@_i(\downarrow s.\Box\downarrow t.(@_t c = @_s(c+1)))$	Assumption 3
4.	$@_i(\Diamond\top)$	Assumption 4
5.	$\neg @_i(\neg\downarrow s.\Box\downarrow t.(@_s c = @_t c))$	Negated conclusion

First we derive that $@_i c \neq @_i(c+1)$:

6.	$@_i\exists x(x = @_i c)$	$\downarrow$ rule on 1.
7.	$@_i(\forall x(@_i x \neq @_i(x+1)))$	$\downarrow$ rule on 2.

The tableau proof

1.	$@_i(\downarrow s.\exists x(x = @_s c))$	Assumption 1
2.	$@_i(\downarrow s.\forall x(@_s x \neq @_s(x+1)))$	Assumption 2
3.	$@_i(\downarrow s.\Box \downarrow t.(@_t c = @_s(c+1)))$	Assumption 3
4.	$@_i(\Diamond \top)$	Assumption 4
5.	$\neg @_i(\neg \downarrow s.\Box \downarrow t.(@_s c = @_t c))$	Negated conclusion

First we derive that $@_i c \neq @_i(c+1)$:

6.	$@_i \exists x(x = @_i c)$	$\downarrow$ rule on 1.
7.	$@_i(\forall x(@_i x \neq @_i(x+1)))$	$\downarrow$ rule on 2.
8.	$@_i(@_i c \neq @_i(c+1))$	$\forall$ rule on 6 and 7

The tableau proof

1.	$@_i(\downarrow s.\exists x(x = @_s c))$	Assumption 1
2.	$@_i(\downarrow s.\forall x(@_s x \neq @_s(x+1)))$	Assumption 2
3.	$@_i(\downarrow s.\Box \downarrow t.(@_t c = @_s(c+1)))$	Assumption 3
4.	$@_i(\Diamond\top)$	Assumption 4
5.	$\neg @_i(\neg \downarrow s.\Box \downarrow t.(@_s c = @_t c))$	Negated conclusion

First we derive that $@_i c \neq @_i(c+1)$:

6.	$@_i \exists x(x = @_i c)$	$\downarrow$ rule on 1.
7.	$@_i(\forall x(@_i x \neq @_i(x+1)))$	$\downarrow$ rule on 2.
8.	$@_i(@_i c \neq @_i(c+1))$	$\forall$ rule on 6 and 7
9.	$@_i c \neq @_i(c+1)$	by $\neq$ rigidity on 8

The tableau proof continued

Now we derive that $@_j c = @_i(c+1)$ for some R successor j of i:

3.	$@_i(\downarrow s.\Box \downarrow t.(@_t c = @_s(c+1)))$	Assumption 3
4.	$@_i(\Diamond\top)$	Assumption 4
10.	$@_i\Diamond j$	$\Diamond$ rule on 4.

The tableau proof continued

Now we derive that $@_j c = @_i(c+1)$ for some R successor j of i:

3.	$@_i(\downarrow s.\Box\downarrow t.(@_t c = @_s(c+1)))$	Assumption 3
4.	$@_i(\Diamond\top)$	Assumption 4
10.	$@_i\Diamond j$	$\Diamond$ rule on 4.
11.	$@_i\Box\downarrow t.(@_t c = @_i(c+1))$	$\downarrow$ rule on 3

The tableau proof continued

Now we derive that $@_j c = @_i(c+1)$ for some R successor j of i:

3.	$@_i(\downarrow s.\Box \downarrow t.(@_t c = @_s(c+1)))$	Assumption 3
4.	$@_i(\Diamond\top)$	Assumption 4
10.	$@_i\Diamond j$	$\Diamond$ rule on 4.
11.	$@_i\Box \downarrow t.(@_t c = @_i(c+1))$	$\downarrow$rule on 3
12.	$@_j \downarrow t.(@_t c = @_i(c+1))$	$\Box$ rule on 10,11

The tableau proof continued

Now we derive that $@_j c = @_i(c+1)$ for some R successor j of i:

3.	$@_i(\downarrow s.\Box\downarrow t.(@_t c = @_s(c+1)))$	Assumption 3
4.	$@_i(\Diamond\top)$	Assumption 4
10.	$@_i\Diamond j$	$\Diamond$ rule on 4.
11.	$@_i\Box\downarrow t.(@_t c = @_i(c+1))$	$\downarrow$rule on 3
12.	$@_j\downarrow t.(@_t c = @_i(c+1))$	$\Box$ rule on 10,11
13.	$@_j(@_j c = @_i(c+1))$	$\downarrow$rule on 12

The tableau proof continued

Now we derive that $@_j c = @_i(c+1)$ for some R successor j of i:

3.	$@_i(\downarrow s.\Box \downarrow t.(@_t c = @_s(c+1)))$	Assumption 3
4.	$@_i(\Diamond \top)$	Assumption 4
10.	$@_i \Diamond j$	$\Diamond$ rule on 4.
11.	$@_i \Box \downarrow t.(@_t c = @_i(c+1))$	$\downarrow$rule on 3
12.	$@_j \downarrow t.(@_t c = @_i(c+1))$	$\Box$ rule on 10,11
13.	$@_j(@_j c = @_i(c+1))$	$\downarrow$rule on 12
14.	$@_j c = @_i(c+1)$	by rigidity on 13

Derive the contradiction

We derive the contradiction using the negated conclusion:

5.	$\neg @_i(\neg \downarrow s.\Box \downarrow t.(@_s c = @_t c))$	Negated conclusion
9.	$@_i c \neq @_i(c+1)$	
10.	$@_i \Diamond j$	
14.	$@_j c = @_i(c+1)$	

Derive the contradiction

We derive the contradiction using the negated conclusion:

5.	$\neg @_i(\neg \downarrow s.\Box \downarrow t.(@_s c = @_t c))$	Negated conclusion
9.	$@_i c \neq @_i(c+1)$	
10.	$@_i \Diamond j$	
14.	$@_j c = @_i(c+1)$	
15.	$@_i(\downarrow s.\Box \downarrow t.(@_s c = @_t c))$	$\neg$ rule on 5

Derive the contradiction

We derive the contradiction using the negated conclusion:

5.	$\neg @_i(\neg \downarrow s.\Box \downarrow t.(@_s c = @_t c))$	Negated conclusion
9.	$@_i c \neq @_i(c+1)$	
10.	$@_i \Diamond j$	
14.	$@_j c = @_i(c+1)$	
15.	$@_i(\downarrow s.\Box \downarrow t.(@_s c = @_t c))$	$\neg$ rule on 5
16.	$@_i(\Box \downarrow t.(@_i c = @_t c))$	$\downarrow$rule on15

Derive the contradiction

We derive the contradiction using the negated conclusion:

5.	$\neg @_i(\neg \downarrow s.\Box \downarrow t.(@_s c = @_t c))$	Negated conclusion
9.	$@_i c \neq @_i(c+1)$	
10.	$@_i \Diamond j$	
14.	$@_j c = @_i(c+1)$	
15.	$@_i(\downarrow s.\Box \downarrow t.(@_s c = @_t c))$	$\neg$ rule on 5
16.	$@_i(\Box \downarrow t.(@_i c = @_t c))$	$\downarrow$rule on15
17.	$@_j(\downarrow t.(@_i c = @_t c))$	$\Box$ rule on 10, 16

Derive the contradiction

We derive the contradiction using the negated conclusion:

5.	$\neg @_i(\neg \downarrow s.\Box \downarrow t.(@_s c = @_t c))$	Negated conclusion
9.	$@_i c \neq @_i(c+1)$	
10.	$@_i \Diamond j$	
14.	$@_j c = @_i(c+1)$	
15.	$@_i(\downarrow s.\Box \downarrow t.(@_s c = @_t c))$	$\neg$ rule on 5
16.	$@_i(\Box \downarrow t.(@_i c = @_t c))$	$\downarrow$rule on15
17.	$@_j(\downarrow t.(@_i c = @_t c))$	$\Box$ rule on 10, 16
18.	$@_j(@_i c = @_j c)$	$\downarrow$rule on 17

Derive the contradiction

We derive the contradiction using the negated conclusion:

5.	$\neg @_i(\neg \downarrow s.\Box \downarrow t.(@_s c = @_t c))$	Negated conclusion
9.	$@_i c \neq @_i(c+1)$	
10.	$@_i \Diamond j$	
14.	$@_j c = @_i(c+1)$	
15.	$@_i(\downarrow s.\Box \downarrow t.(@_s c = @_t c))$	$\neg$ rule on 5
16.	$@_i(\Box \downarrow t.(@_i c = @_t c))$	$\downarrow$rule on15
17.	$@_j(\downarrow t.(@_i c = @_t c))$	$\Box$ rule on 10, 16
18.	$@_j(@_i c = @_j c)$	$\downarrow$rule on 17
19.	$@_i c = @_j c$	rigidity rule on 18

Derive the contradiction

We derive the contradiction using the negated conclusion:

5.	$\neg @_i(\neg \downarrow s.\Box \downarrow t.(@_s c = @_t c))$	Negated conclusion
9.	$@_i c \neq @_i(c+1)$	
10.	$@_i \Diamond j$	
14.	$@_j c = @_i(c+1)$	
15.	$@_i(\downarrow s.\Box \downarrow t.(@_s c = @_t c))$	$\neg$ rule on 5
16.	$@_i(\Box \downarrow t.(@_i c = @_t c))$	$\downarrow$ rule on15
17.	$@_j(\downarrow t.(@_i c = @_t c))$	$\Box$ rule on 10, 16
18.	$@_j(@_i c = @_j c)$	$\downarrow$ rule on 17
19.	$@_i c = @_j c$	rigidity rule on 18
20.	$@_i c = @_i(c+1)$	replacement on 19, 14

Expressing domain conditions by pure axioms

It is easy to show that

$$\begin{gathered}\mathfrak{M} \text{ has expanding domains} \\ \text{if and only if} \\ \mathfrak{M} \models \downarrow s.(\exists x(x = @_s c) \rightarrow [\mathrm{F}]\, \exists x(x = @_s c)).\end{gathered}$$

And it is easy to find a pure axiom for contracting domains. The conjunction of these two axioms characterizes constant domains. Adding these axioms to our tableau system gives us complete systems for expanding, contracting, and constant domains.
But we can be smarter than this and add rules instead...

Implementing constraints on the domains

	condition	*tableau rule*
Expanding	$Rxy \wedge c \in D(x) \Rightarrow c \in D(y)$	$\dfrac{@_i\langle \mathrm{F}\rangle j \quad \mathsf{E}_i(c)}{\mathsf{E}_j(c)}$
Contracting	$Rxy \wedge c \in D(y) \Rightarrow c \in D(x)$	$\dfrac{@_i\langle \mathrm{F}\rangle j \quad \mathsf{E}_j(c)}{\mathsf{E}_i(c)}$
Constant	$c \in D(x) \Rightarrow c \in D(y)$	$\dfrac{\mathsf{E}_i(c)}{\mathsf{E}_j(c)}$

Results

- The tableau system is complete for the class of all models.

Results

- The tableau system is complete for the class of all models.
- All completeness results for specific frame classes are inherited from propositional hybrid logic with $\downarrow$. That is, any extension via pure formulas is automatically complete. This result covers all the philosophically "standard" orthodox first-order modal logics.

Results

- The tableau system is complete for the class of all models.
- All completeness results for specific frame classes are inherited from propositional hybrid logic with $\downarrow$. That is, any extension via pure formulas is automatically complete. This result covers all the philosophically "standard" orthodox first-order modal logics.
- Adding the appropriate domain rule yields completeness for such domains for all these logics. A uniform modal construction works for all these logics. (No such general modal construction is known in orthodox first-order modal logic).

Results

- The tableau system is complete for the class of all models.
- All completeness results for specific frame classes are inherited from propositional hybrid logic with $\downarrow$. That is, any extension via pure formulas is automatically complete. This result covers all the philosophically "standard" orthodox first-order modal logics.
- Adding the appropriate domain rule yields completeness for such domains for all these logics. A uniform modal construction works for all these logics. (No such general modal construction is known in orthodox first-order modal logic).
- Moreover, all these logic have interpolation, and the tableau system can be used to compute interpolants. (Kit Fine showed that interpolation almost never holds in orthodox first-order modal logic).

Summing up . . .

- Many natural language expressions contain an implicit reference to situation where they are evaluated. Because $\downarrow$ and @ allow evaluation points to be named and manipulated, hybrid logic handles a number of traditional "problems" smoothly.
- Hybridizing orthodox modal logic is a straightforward extension of what we saw in the propositional case. But one new idea is natural: use @ as term rigidifier, not just as a modality.
- Orthodox first-order modal logic inherits all propositional hybrid logics general completeness and interpolation results.