First-Order Modal Logic: Predicate Abstracts and Definite Descriptions

Melvin Fitting
City University of New York

Poland
April 10, 2024

Last October a new edition of the
Fitting-Mendelsohn book came out.

First-Order Modal Logic
Second Edition
Melvin Fitting and Richard Mendelsohn
Springer 2023

I was asked to say something today about
the treatment of definite descriptions in this edition.

It is, in fact, completely different than it was
in the first edition.

Papers by both
Andrzej Indrzejczaj and Eugenio Orlandelli
were clearly an influence,
though our treatment is not exactly
the same as either of theirs.

I will sketch the approach of our second edition.
First syntax details,
then semantics,
and finally tableau proof systems.

The tableau presentation will begin at the beginning,
and work its way to more complicated things.

The book has soundness and completeness proofs,
but they are more than I can discuss today.

Syntax

The propositional basis is the usual.
Logical connectives: $\wedge, \vee, \neg, \supset$,
Modal operators: $\Box, \Diamond$.

Quantification adds $\forall, \exists$.

Equality, $=$, plays its special role.

All this is straightforward and
I'll skip the usual details.

We do have constant and function symbols.
But they will be *non-rigid*.

This entails that, like quantifiers,
they have *scopes*.

Here is the syntactic machinery for this.

Let Φ be a formula.

Let Φ be a formula.

1. $\langle \lambda x.\Phi \rangle$ is a *predicate abstract*; the free variable occurrences of $\langle \lambda x.\Phi \rangle$ are those of Φ except for occurrences of x.

Let Φ be a formula.

1. $\langle \lambda x.\Phi \rangle$ is a *predicate abstract*; the free variable occurrences of $\langle \lambda x.\Phi \rangle$ are those of Φ except for occurrences of x.

2. A constant symbol c is an *intensional term*. If f is an n-ary function symbol and x_1, ..., x_n are variables, $f(x_1, \ldots, x_n)$ is an *intensional term*.

Let Φ be a formula.

1. $\langle \lambda x.\Phi \rangle$ is a *predicate abstract*; the free variable occurrences of $\langle \lambda x.\Phi \rangle$ are those of Φ except for occurrences of x.

2. A constant symbol c is an *intensional term*. If f is an n-ary function symbol and x_1, ..., x_n are variables, $f(x_1, \ldots, x_n)$ is an *intensional term*.

3. If t is an intensional term, then $\langle \lambda x.\Phi \rangle(t)$ is a formula; the free variable occurrences of this formula are those of the predicate abstract $\langle \lambda x.\Phi \rangle$, together with all variable occurrences in t.

For example, we do not syntactically allow $\Diamond P(f(c))$.

A proper version of it using predicate abstraction could be any of the following.

$$\langle \lambda x. \langle \lambda y. \Diamond P(y) \rangle (f(x)) \rangle (c)$$
$$\langle \lambda x. \Diamond \langle \lambda y. P(y) \rangle (f(x)) \rangle (c)$$
$$\Diamond \langle \lambda x. \langle \lambda y. P(y) \rangle (f(x)) \rangle (c)$$

All behave differently semantically.
As you will see.

We may also have definite descriptions.
For this we add the following syntax.

4. If Ψ is a formula and y is a variable, $\imath y.\Psi$ is an intensional term, called a *definite description*. Its free variable occurrences are those of Ψ except for occurrences of y.

That's it for syntax.

Semantics (constant domain)

The Fitting/Mendelsohn book discusses both constant and varying domain semantics and tableau systems.

I'll only discuss a constant domain version today.

There are two reasons for this.

The first is the matter of time.
Varying domains are more complicated,
and constant domains are illustrative enough.

But second, constant domains can be enhanced
with an existence predicate
and so can simulate varying domains quite well.

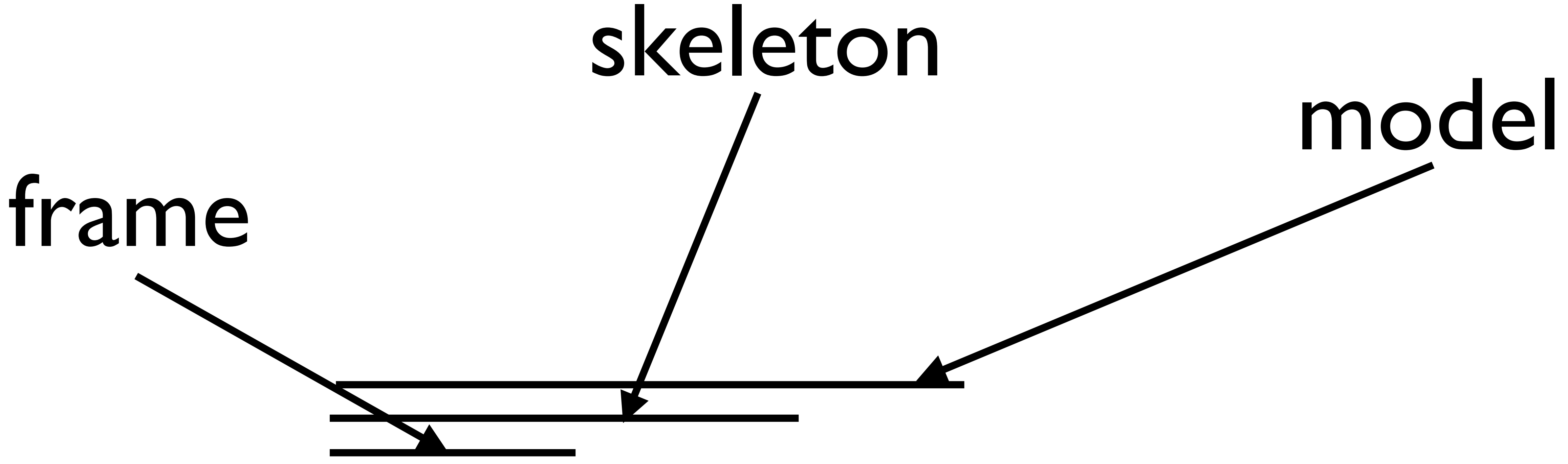
$$\langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$$

possible worlds

accessibility relation

domain (a non-empty set)

interpretation



$$\langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$$

$$\langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$$

No special conditions for now. The logic K.

Non-empty set.

More to be said.

$\mathcal{I}$ assigns to each n-place relation symbol R
and to each possible world $\Gamma \in \mathcal{G}$,
some n-place relation on $\mathcal{D}$.

$\mathcal{I}$ assigns to each intensional constant symbol c,
and to *some* members $\Gamma \in \mathcal{G}$,
a member of the domain of $\mathcal{F}$.
If $\mathcal{I}(c, \Gamma)$ is defined, we say that c *designates* at Γ.

$\mathcal{I}$ assigns to each n-ary function symbol f,
and to *some* $\Gamma \in \mathcal{G}$,
an n-ary *partial* function on the domain of the frame.
If $\mathcal{I}(f, \Gamma)$ is defined, we say that f *designates* at Γ.
If f designates at Γ and $\langle d_1, \ldots, d_n \rangle$ is in the domain of $\mathcal{I}(f, \Gamma)$,
we say f is *specified* on $\langle d_1, \ldots, d_n \rangle$ at Γ.

Note:
we will always assume the equality symbol ‘$=$’
is interpreted by the equality relation
on $\mathcal{D}$.

Next, a definition of truth in a model.

Note: the model conditions can be modified a lot.

Domains can be varying, monotonic, anti-monotonic.

Constant and function symbols can be required to be total, instead of partial.

All combinations of these have corresponding tableau systems.

We'll only discuss the one version here.

$\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ is a constant domain first-order modal model.

A *valuation* in $\mathcal{M}$ is a mapping v from free variables to $\mathcal{D}$.

Valuations v and w are x-*variants* if they agree on all variables except possibly x.

Let $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ be a constant domain model, $\Gamma \in \mathcal{G}$, and v a valuation in $\mathcal{M}$:

$\mathcal{M}, \Gamma \Vdash_v \Phi$ symbolizes that formula Φ is true at world Γ using valuation v.

1. If R is an n-place relation symbol, $\mathcal{M}, \Gamma \Vdash_v R(x_1, \ldots, x_n)$ provided $\langle v(x_1), \ldots, v(x_n) \rangle \in \mathcal{I}(R, \Gamma)$.

2. $\mathcal{M}, \Gamma \Vdash_v \neg\Phi \iff \mathcal{M}, \Gamma \nVdash_v \Phi$.

3. $\mathcal{M}, \Gamma \Vdash_v (\Phi \wedge \Psi) \iff \mathcal{M}, \Gamma \Vdash_v \Phi$ and $\mathcal{M}, \Gamma \Vdash_v \Psi$.

4. $\mathcal{M}, \Gamma \Vdash_v (\Phi \vee \Psi) \iff \mathcal{M}, \Gamma \Vdash_v \Phi$ or $\mathcal{M}, \Gamma \Vdash_v \Psi$.

5. $\mathcal{M}, \Gamma \Vdash_v (\Phi \supset \Psi) \iff \mathcal{M}, \Gamma \nVdash_v \Phi$ or $\mathcal{M}, \Gamma \Vdash_v \Psi$.

6. $\mathcal{M}, \Gamma \Vdash_v \Box\Phi \iff$ for every $\Delta \in \mathcal{G}$, if $\Gamma\mathcal{R}\Delta$ then $\mathcal{M}, \Delta \Vdash_v \Phi$.

7. $\mathcal{M}, \Gamma \Vdash_v \Diamond\Phi \iff$ for some $\Delta \in \mathcal{G}$, $\Gamma\mathcal{R}\Delta$ and $\mathcal{M}, \Delta \Vdash_v \Phi$.

8. $\mathcal{M}, \Gamma \Vdash_v (\forall x)\Phi \iff$ for every x-variant w of v in $\mathcal{M}$, $\mathcal{M}, \Gamma \Vdash_w \Phi$.

9. $\mathcal{M}, \Gamma \Vdash_v (\exists x)\Phi \iff$ for some x-variant w of v in $\mathcal{M}$, $\mathcal{M}, \Gamma \Vdash_w \Phi$.

This is all as usual.

Next we come to the predicate abstract machinery.

10. For an intensional constant symbol c, if c designates at Γ in $\mathcal{G}$ then

$$\mathcal{M}, \Gamma \Vdash_v \langle \lambda x.\Phi \rangle(c) \iff \mathcal{M}, \Gamma \Vdash_w \Phi$$

where w is the x-variant of v such that $w(x) = \mathcal{I}(c, \Gamma)$,

10. For an intensional constant symbol c, if c designates at Γ in $\mathcal{G}$ then

$$\mathcal{M}, \Gamma \Vdash_v \langle \lambda x.\Phi \rangle(c) \Longleftrightarrow \mathcal{M}, \Gamma \Vdash_w \Phi$$

where w is the x-variant of v such that $w(x) = \mathcal{I}(c, \Gamma)$, and otherwise,

$$\mathcal{M}, \Gamma \nVdash_v \langle \lambda x.\Phi \rangle(c).$$

11. For an intensional n-place function symbol f, if f designates at $\Gamma \in \mathcal{G}$ and f is specified on $\langle v(x_1), \ldots, v(x_n) \rangle$ at Γ then

$$\mathcal{M}, \Gamma \Vdash_v \langle \lambda y.\Phi \rangle(f(x_1, \ldots, x_n)) \Longleftrightarrow \mathcal{M}, \Gamma \Vdash_w \Phi$$

where w is the y-variant of v such that $w(y) = \mathcal{I}(f, \Gamma)(v(x_1), \ldots, v(x_n))$, Here's an example.

$$\Gamma \longleftrightarrow \Delta$$

Model domain is $\{0, 1, 2, 3, \ldots\}$

E is the evens at both worlds.

P is empty at Γ and is the primes at Δ.

f is doubling at Γ and is successor at Δ.

g is undefined at Γ.

$\mathcal{I}(g, \Delta)(n) = n + 1$ if $n \neq 3$, undefined at 3.

$\mathcal{I}(c, \Gamma) = 3$.

$\mathcal{I}(c, \Delta)$ undefined.

$$\Gamma \longleftrightarrow \Delta$$

Model domain is $\{0, 1, 2, 3, \ldots\}$
E is the evens at both worlds.
P is empty at Γ and is the primes at Δ.
f is doubling at Γ and is successor at Δ.
g is undefined at Γ.
$\mathcal{I}(g, \Delta)(n) = n + 1$ if $n \neq 3$, undefined at 3.
$\mathcal{I}(c, \Gamma) = 3$.
$\mathcal{I}(c, \Delta)$ undefined.

$$\mathcal{M}, \Gamma \Vdash_v \langle \lambda x. \langle \lambda y. \Diamond P(y) \rangle (f(x)) \rangle (c)$$

$$\iff \mathcal{M}, \Gamma \Vdash_w \langle \lambda y. \Diamond P(y) \rangle (f(x))$$

where w is like v except that

$$w(x) = \mathcal{I}(c, \Gamma) = 3$$

$$\iff \mathcal{M}, \Gamma \Vdash_u \Diamond P(y)$$

where u is like w except that

$$u(y) = \mathcal{I}(f, \Gamma)(w(x))$$

$$= 2w(x) = 6$$

$$\Gamma \longleftrightarrow \Delta$$

Model domain is $\{0, 1, 2, 3, \ldots\}$
E is the evens at both worlds.
P is empty at Γ and is the primes at Δ.
f is doubling at Γ and is successor at Δ.
g is undefined at Γ.
$\mathcal{I}(g, \Delta)(n) = n + 1$ if $n \neq 3$, undefined at 3.
$\mathcal{I}(c, \Gamma) = 3$.
$\mathcal{I}(c, \Delta)$ undefined.

$\mathcal{M}, \Gamma \Vdash_v \langle \lambda x.\langle \lambda y.\Diamond P(y)\rangle(f(x))\rangle(c)$

$\Longleftrightarrow \mathcal{M}, \Gamma \Vdash_w \langle \lambda y.\Diamond P(y)\rangle(f(x))$

where w is like v except that

$w(x) = \mathcal{I}(c, \Gamma) = 3$

$\Longleftrightarrow \mathcal{M}, \Gamma \Vdash_u \Diamond P(y)$

where u is like w except that

$u(y) = \mathcal{I}(f, \Gamma)(w(x))$

$= 2w(x) = 6$

$\Longleftrightarrow \mathcal{M}, \Delta \Vdash_u P(y)$

$$\Gamma \longleftrightarrow \Delta$$

Model domain is $\{0, 1, 2, 3, \ldots\}$
E is the evens at both worlds.
P is empty at Γ and is the primes at Δ.
f is doubling at Γ and is successor at Δ.
g is undefined at Γ.
$\mathcal{I}(g, \Delta)(n) = n + 1$ if $n \neq 3$, undefined at 3.
$\mathcal{I}(c, \Gamma) = 3$.
$\mathcal{I}(c, \Delta)$ undefined.

$$\mathcal{M}, \Gamma \Vdash_v \langle \lambda x. \Diamond \langle \lambda y. P(y) \rangle (g(x)) \rangle (c)$$

$$\Longleftrightarrow \mathcal{M}, \Gamma \Vdash_w \Diamond \langle \lambda y. P(y) \rangle (g(x))$$

where w is like v except that

$w(x) = \mathcal{I}(c, \Gamma) = 3$

$$\Longleftrightarrow \mathcal{M}, \Delta \Vdash_w \langle \lambda y. P(y) \rangle (g(x))$$

but g is not specified on $w(x) = 3$, so this is false.

$$\Gamma \longleftrightarrow \Delta$$

Model domain is $\{0, 1, 2, 3, \ldots\}$
E is the evens at both worlds.
P is empty at Γ and is the primes at Δ.
f is doubling at Γ and is successor at Δ.
g is undefined at Γ.
$\mathcal{I}(g, \Delta)(n) = n + 1$ if $n \neq 3$, undefined at 3.
$\mathcal{I}(c, \Gamma) = 3$.
$\mathcal{I}(c, \Delta)$ undefined.

$$\mathcal{M}, \Gamma \Vdash_v \langle \lambda x. \langle \lambda y. \langle \lambda z. \Diamond P(z) \rangle (f(y)) \rangle (g(x)) \rangle (c)$$

$$\Longleftrightarrow \mathcal{M}, \Gamma \Vdash_w \langle \lambda y. \langle \lambda z. \Diamond P(z) \rangle (f(y)) \rangle (g(x))$$

where w is like v except that

$w(x) = \mathcal{I}(c, \Gamma) = 3$

but g does not designate at Γ, so this is false.

Tableaus, As Far As We've Gone (constant domain)

We use *prefixed* tableaus.
These can handle the 'standard' modal logics,
K, T, S4, S5, etc.

They cannot handle things like S4.3.

Other proof systems would do, and may be more general.
I'm kind of partial to hybrid logic,
but I'll skip all that here.

A *prefix* is a finite sequence of positive integers. A *prefixed formula* is an expression of the form $\sigma\, X$, where σ is a prefix and X is a formula.

Informally a prefix σ names a possible world in some modal model, and $\sigma\, X$ tells us that X is true at the world that σ names.

A prefix $\sigma.n$ should name a world that is accessible from the one that σ names.

Other conditions on prefixes depend on which modal logic we are considering, but this is enough to get started. It gives us *K*.

The tableau idea.

Tableaus are refutation systems.

A proof amounts to showing that assuming a formula could be false somewhere leads to a contradiction.

Here this is, more formally.

A tableau for X starts with $1 \neg X$.

It grows, as a tree, using branch extension rules, given shortly.

A branch is *closed* if it contains both σA and $\sigma \neg A$,

A tableau is *closed* if all branches are.

A proof of X is a closed tableau starting with $1 \neg X$.

Propositional Rules

Conjunctive Rules:

$$\frac{\sigma\, X \wedge Y}{\begin{array}{c}\sigma\, X \\ \sigma\, Y\end{array}} \qquad \frac{\sigma\, \neg(X \vee Y)}{\begin{array}{c}\sigma\, \neg X \\ \sigma\, \neg Y\end{array}} \qquad \frac{\sigma\, \neg(X \supset Y)}{\begin{array}{c}\sigma\, X \\ \sigma\, \neg Y\end{array}}$$

Disjunctive Rules:

$$\frac{\sigma\, X \vee Y}{\sigma\, X \mid \sigma\, Y} \qquad \frac{\sigma\, \neg(X \wedge Y)}{\sigma\, \neg X \mid \sigma\, \neg Y} \qquad \frac{\sigma\, X \supset Y}{\sigma\, \neg X \mid \sigma\, Y}$$

Negation Rule:

$$\frac{\sigma\, \neg\neg X}{\sigma\, X}$$

Quantifier Rules

A parameter is a free variable that is never bound.

(We use a new and distinct set for these.)

Universal Rules Where p is any parameter whatsoever.

$$\frac{\sigma\ (\forall x)\Phi(x)}{\sigma\ \Phi(p)} \qquad \frac{\sigma\ \neg(\exists x)\Phi(x)}{\sigma\ \neg\Phi(p)}$$

Existential Rules Where p is a parameter that is *new to the tableau branch*.

$$\frac{\sigma\ (\exists x)\Phi(x)}{\sigma\ \Phi(p)} \qquad \frac{\sigma\ \neg(\forall x)\Phi(x)}{\sigma\ \neg\Phi(p)}$$

Equality Rules

Reflexivity Rule For a parameter p and a prefix σ already on the branch:

$$\frac{}{\sigma\ (p = p)}$$

Atomic Substitutivity Rule For an atomic formula $\Phi(x)$:

$$\frac{\begin{array}{c}\sigma\ (p = q)\\ \tau\ \Phi(p)\end{array}}{\tau\ \Phi(q)}$$

Here $\Phi(q)$ is $\Phi(p)$ with *some* p occurrences replaced with q.

Note: valuations of variables is rigid.

Note: "Atomic" can be dropped.

Modal Rules (for K)

Prefixed Possibility Rules If the prefix $\sigma.n$ is new to the branch,

$$\frac{\sigma\ \Diamond X}{\sigma.n\ X} \qquad \frac{\sigma\ \neg\Box X}{\sigma.n\ \neg X}$$

Prefixed Necessity Rules If the prefix $\sigma.n$ is *not* new to the branch,

$$\frac{\sigma\ \Box X}{\sigma.n\ X} \qquad \frac{\sigma\ \neg\Diamond X}{\sigma.n\ \neg X}$$

Constant and Function Symbols

Our formal language was enlarged for tableaus. *Parameters* were added. Think of these as stand-ins for members of a model domain.

Now, one more enlargement. If c is a constant symbol, and σ is a prefix, then tableaus may also contain c^{σ}.

Think of c^{σ} as a stand-in for what c designates at σ.

Similarly for function symbols.

Object Terms

Variables (including parameters) are object terms.

If c is a constant symbol, c^σ is an object term.

If f is an n-place function symbol and t_1, ..., t_n are object terms, then $f^\sigma(t_1, \ldots, t_n)$ is an object term.

An object term is *pseudo-closed* if it contains no free variables other than parameters.

Equality Rules Again

Our earlier substitutivity rule for equality needs extension.

Atomic Substitutivity Rule For an atomic formula $\Phi(x)$:

$$\frac{\begin{array}{c}\sigma\,(p = q)\\ \tau\,\Phi(p)\end{array}}{\tau\,\Phi(q)}$$

Where p and q are any pseudo-closed object terms.

The Reflexivity Rule stays the same.
Only for parameters.

Existence Abstract

By $\mathbf{E}$ we mean the predicate abstract $\langle \lambda x.(\exists y)(y = x) \rangle$.
Then for an object term t, $\mathbf{E}(t)$ is $\langle \lambda x.(\exists y)(y = x) \rangle(t)$.

It, perhaps, looks useless here.
In constant domains, loosely, everything exists everywhere.

But it allows us to equate object terms with parameters.

This serves to simplify the rule statements.

Generation on a Branch

For a non-rigid constant symbol c we say:

1. c^σ is *positively generated* on a tableau branch if the branch contains $\sigma\,\langle\lambda x.\Phi(x)\rangle(c)$ for some $\Phi(x)$;

2. c^σ is *negatively generated* on a tableau branch if the branch contains $\sigma\,\neg\langle\lambda x.\Phi(x)\rangle(c)$ for some $\Phi(x)$.

Generation on a Branch

For an n-place function symbol f we say:

1. $f^\sigma(t_1, \ldots, t_n)$ is *positively generated* on a tableau branch if the branch contains $\sigma\,\langle \lambda x.\Phi(x)\rangle(f(t_1, \ldots, t_n))$;

2. $f^\sigma(t_1, \ldots, t_n)$ is *negatively generated* on a tableau branch if the branch contains $\sigma\,\neg\langle \lambda x.\Phi(x)\rangle(f(t_1, \ldots, t_n))$.

Predicate Abstraction Rules

For a constant symbol c.

$$\frac{\sigma\ \langle\lambda x.\Phi(x)\rangle(c)}{\sigma\ \Phi(c^{\sigma})} \qquad \frac{\sigma\ \neg\langle\lambda x.\Phi(x)\rangle(c)}{\sigma\ \neg\Phi(c^{\sigma})}$$

provided c^{σ} is positively generated on the tableau branch

Predicate Abstraction Rules

For an n place f, and pseudo-closed object terms $t_1, \ldots, t_n$.

$$\frac{\sigma\ \langle \lambda x.\Phi(x)\rangle(f(t_1,\ldots,t_n))}{\sigma\ \Phi(f^\sigma(t_1,\ldots,t_n))}$$

$$\frac{\sigma\ \neg\langle \lambda x.\Phi(x)\rangle(f(t_1,\ldots,t_n))}{\sigma\ \neg\Phi(f^\sigma(t_1,\ldots,t_n))}$$

provided $f^\sigma(t_1,\ldots,t_n)$ is positively generated on the tableau branch

Existence Rules

And finally, *Existence Rules* For any constant symbol c, n-place function symbol , and pseudo-closed object terms t_1, . . . , t_n:

$$\frac{\sigma\,\langle\lambda x.\Phi(x)\rangle(c)}{\sigma\,\mathbf{E}(c)} \qquad \frac{\sigma\,\langle\lambda x.\Phi(x)\rangle(f(t_1,\ldots,t_n))}{\sigma\,\mathbf{E}(f(t_1,\ldots,t_n))}$$

We don’t have definite descriptions yet.
But we do have non-rigid terms that might not designate.
And we have predicate abstraction.
This is enough to represent most of Russell’s examples.

There are two aspects to Russell's treatment.

1. Determining what is being designated, if anything is.
2. Determining how this behaves in a context.

Understanding the behavior in a context
is really an issue of scope distinctions.

What we have so far is quite enough for this.

Here are a few formal examples.

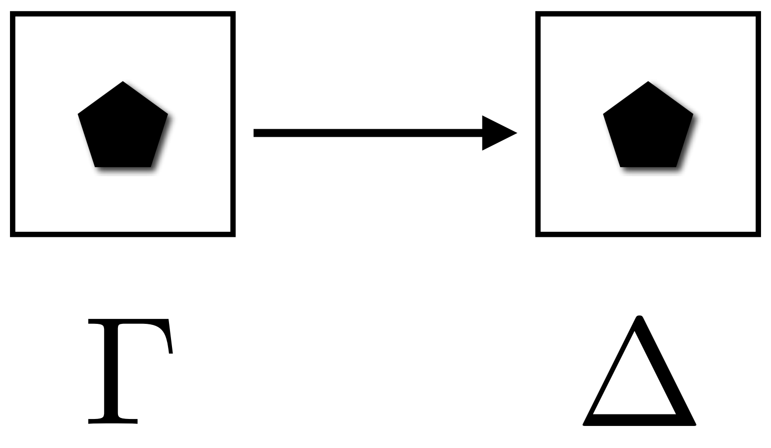
$\neg\langle\lambda x.P(x)\rangle(c) \supset \langle\lambda x.\neg P(x)\rangle(c)$ is not valid.

Consider a model with a possible world at which c does not designate.

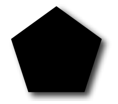
$\langle\lambda x.\neg P(x)\rangle(c) \supset \neg\langle\lambda x.P(x)\rangle(c)$ is valid.

$$
\begin{array}{lll}
1 & \neg[\langle\lambda x.\neg P(x)\rangle(c) \supset \neg\langle\lambda x.P(x)\rangle(c)] & 1. \\
1 & \langle\lambda x.\neg P(x)\rangle(c) & 2. \\
1 & \neg\neg\langle\lambda x.P(x)\rangle(c) & 3. \\
1 & \langle\lambda x.P(x)\rangle(c) & 4. \\
1 & \neg P(c^1) & 5. \\
1 & P(c^1) & 6.
\end{array}
$$

$\Diamond\langle\lambda x.P(x)\rangle(c) \supset \langle\lambda x.\Diamond P(x)\rangle(c)$ is not valid.



c does not designate at Γ

c does designate at Δ
and it designates 
which is something for which P
is true at Δ.

$\langle \lambda x.\Diamond P(x)\rangle(c) \supset \Diamond\langle \lambda x.P(x)\rangle(c)$ is also not valid.

Exercise.

More Complicated Example

Using the rules we've given so far,
here is a tableau proof
of the following.

$$\Diamond\langle\lambda x.P(x)\rangle(c) \supset (\exists x)\Diamond P(x)$$

$$
\begin{array}{lll}
1 & \neg[\Diamond\langle\lambda x.P(x)\rangle(c) \supset (\exists x)\Diamond P(x)] & 1. \\
1 & \quad \Diamond\langle\lambda x.P(x)\rangle(c) & 2. \\
1 & \neg(\exists x)\Diamond P(x) & 3. \\
1.1 & \quad \langle\lambda x.P(x)\rangle(c) & 4. \\
1.1 & \quad P(c^{1.1}) & 5. \\
1.1 & \quad \mathbf{E}(c) & 6. \\
1.1 & \quad \langle\lambda x.(\exists y)(y = x)\rangle(c) & 7. \\
1.1 & \quad (\exists y)(y = c^{1.1}) & 8. \\
1.1 & \quad p = c^{1.1} & 9. \\
1 & \neg\Diamond P(p) & 10. \\
1.1 & \neg P(p) & 11. \\
1.1 & \quad P(p) & 12.
\end{array}
$$

Definite Descriptions Semantics

We extend the definition of the interpretation function $\mathcal{I}$ to definite descriptions.

We say $\mathcal{I}(\imath x.\Phi, \Gamma)$ is defined in model $\mathcal{M}$ under valuation v if there is exactly one x-variant w of v such that $\mathcal{M}, \Gamma \Vdash_w \Phi$.

If this is the case then $\imath x.\Phi$ *designates* at Γ under v, and $\mathcal{I}(\imath x.\Phi, \Gamma) = w(x)$.

For a definite description $\imath x.\Phi$:

1. If $\imath x.\Phi$ designates at Γ in $\mathcal{G}$ under valuation v then

$$\mathcal{M}, \Gamma \Vdash_v \langle \lambda y.\Psi \rangle(\imath x.\Phi) \Longleftrightarrow \mathcal{M}, \Gamma \Vdash_w \Psi$$

   where w is the y-variant of v such that $w(y) = \mathcal{I}(\imath x.\Phi, \Gamma)$.

2. If $\imath x.\Phi$ fails to designate at Γ under v then $\mathcal{M}, \Gamma \nVdash_v \langle \lambda y.\Phi \rangle(\imath x.\Phi)$.

This is a semantic counterpart of Russell's famous definition.

It applies using varying domain semantics too.

But it is in the constant domain (possibilist) setting
that, we feel, it behaves most naturally.

Here is an essentially modal example.

Let $\mathbf{D}$ be the predicate abstract $\langle \lambda x . x = x \rangle$.

Then $\mathbf{D}(t)$ is true at
exactly those worlds where t designates.

The example is the validity of:
$\mathbf{D}(\imath x . \Diamond P(x)) \supset \Diamond \mathbf{D}(\imath x . P(x))$.

Informally, if *the possible* P designates,
then it is possible that *the* P designates.

The example is the validity of: $\mathbf{D}(\imath x.\Diamond P(x)) \supset \Diamond\mathbf{D}(\imath x.P(x))$.

And here is an informally presented argument for its validity.

Suppose $\mathbf{D}(\imath x.\Diamond P(x))$ is true at world Γ.
Then $\imath x.\Diamond P(x)$ designates at Γ.
There is exactly one a such that $\Diamond P(a)$ is true at Γ.
At some accessible Δ, $P(a)$ is true.
If $P(b)$ were true at Δ for some $b \neq a$,
then $\Diamond P(b)$ would be true at Γ.
But no.
So there is exactly one a with $P(a)$ true at Δ.
So $\imath x.P(x)$ designates at world Δ.
$\mathbf{D}(\imath x.P(x))$ is true at Δ.
$\Diamond\mathbf{D}(\imath x.P(x))$ is true at Γ.

Definite Description Tableau Rules

The first thing is to extend earlier rules to allow definite descriptions.

If $\imath x.\Phi$ is a definite description and σ is a prefix, then $[\imath x.\Phi]^\sigma$ is an object term. As with other object terms, it is *pseudo-closed* if it contains no free variables other than parameters.

Our earlier equality rules now apply to definite descriptions too.

Positive generation is likewise extended.
$[\imath x.\Psi]^\sigma$ is *positively generated* on a tableau branch
if the branch contains $\sigma\,\langle\lambda x.\Phi(x)\rangle(\imath x.\Psi)$ for some $\Phi(x)$;

Our earlier predicate abstraction and equality rules
are likewise extended.

$$\frac{\sigma\ \langle \lambda x.\Phi(x)\rangle(\imath y.\Psi(y))}{\sigma\ \Phi([\imath y.\Psi(y)]^{\sigma})}$$

$$\frac{\sigma\ \neg\langle \lambda x.\Phi(x)\rangle(\imath y.\Psi(y))}{\sigma\ \neg\Phi([\imath y.\Psi(y)]^{\sigma})}$$

provided $[\imath y.\Psi(y)]^{\sigma}$ is positively generated on the tableau branch

$$\frac{\sigma\ \langle \lambda x.\Phi(x)\rangle(\imath y.\Psi(y))}{\sigma\ \mathbf{E}(\imath y.\Psi(y)}$$

This gives us the following *derived* rules.

$$\frac{\sigma\ \langle \lambda x.\Phi(x)\rangle(c)}{\sigma\, p = c^{\sigma}} \qquad \frac{\sigma\ \langle \lambda x.\Phi(x)\rangle(f(t_1,\ldots,t_n))}{\sigma\, p = f^{\sigma}(t_1,\ldots,t_n)} \qquad \frac{\sigma\ \langle \lambda x.\Phi(x)\rangle(\imath y.\Psi(y))}{\sigma\, p = [\imath y.\Psi(y)]^{\sigma}}$$

Where parameter p is new to the branch.

All this is straightforward.
Now, here are the rules that
really say we are discussing *definite descriptions*.

First the positive rules.

$$\frac{\sigma\ \langle \lambda x.\Phi(x)\rangle(\imath y.\Psi(y))}{\sigma\ \Psi([\imath y.\Psi(y)]^{\sigma})}$$

What is designated by *the* Ψ must satisfy Ψ.

First the positive rules.

$$\frac{\sigma \langle \lambda x.\Phi(x)\rangle(\imath y.\Psi(y))}{\sigma \neg\Psi(p) \mid \sigma \langle \lambda x.x = p\rangle(\imath y.\Psi(y))}$$

for any parameter p

If *the* Ψ designates then
for any parameter p
either p doesn't satisfy Ψ, or
it does and so must be *the* Ψ.

Next the Negative Rule.

$$\frac{\sigma\ \neg\langle\lambda x.\Phi(x)\rangle(\imath y.\Psi(y))}{\sigma\ \neg\Psi(p) \quad \Bigg| \quad \begin{array}{l}\sigma\ \Psi(q)\\ \sigma\ \neg(p=q)\end{array} \quad \Bigg| \quad \sigma\ \langle\lambda x.x=p\rangle(\imath y.\Psi(y))}$$

Given the rule premise either:
p doesn't make Ψ true,
or it does but so does some other q,
or p is the only thing making Ψ true
and so p is what $\imath y.\Psi$ designates.

A Tableau Example

$$(\forall z)\langle\lambda x.x = z\rangle(\imath x.x = z)$$

In words: *for each* z*, the* z *has the property of being* z.

$1\ \neg(\forall z)\langle \lambda x.x = z\rangle(\iota x.x = z)$ 1.

$1\ \neg\langle \lambda x.x = p\rangle(\iota x.x = p)$ 2.

Left branch:

$1\ \neg(p = p)$ 3.

$1\ (p = p)$ 7.

Middle branch:

$1\ (q = p)$ 4.

$1\ \neg(p = q)$ 5.

$1\ \neg(p = p)$ 8.

$1\ (p = p)$ 9.

Right branch:

$1\ \langle \lambda x.x = p\rangle(\iota x.x = p)$ 6.

$1\ [\iota x.x = p]^1 = p$ 10.

$1\ \neg([\iota x.x = p]^1 = p)$ 11.

All branches are closed.

You might try proving the following.

$$[\mathbf{E}(\imath x.T(x)) \wedge (\forall x)\Diamond\neg T(x)] \supset \langle \lambda x.\Diamond\neg\langle \lambda y.x = y\rangle(\imath z.T(z))\rangle(\imath z.T(z))$$

It asserts,
under reasonable conditions,
that someday the tallest person in the world
might not be the tallest person in the world.

If you have trouble, a proof is given as
an example in our book.

Or, as an easier problem,
give a tableau proof of
$\mathbf{D}(\imath x.\Diamond P(x)) \supset \Diamond\mathbf{D}(\imath x.P(x))$.

We saw this earlier, semantically.

Notes to me:
our book gives four versions for each modal logic treated;
what are they;
I only discussed one;
Why this one?

Thank You