# Logic 2: Modal Logic

## Lecture 18

Wolfgang Schwarz

University of Edinburgh

# Trees for first-order predicate logic

# Trees for first-order predicate logic

Target: $\forall x \neg Fx \rightarrow \neg \exists x(Fx \wedge Gx)$

1. $\neg(\forall x \neg Fx \rightarrow \neg \exists x(Fx \wedge Gx))$ (Ass.)

# Trees for first-order predicate logic

Target: $\forall x \neg Fx \rightarrow \neg \exists x(Fx \wedge Gx)$

| | | |
|---|---|---|
| 1. | $\neg(\forall x \neg Fx \rightarrow \neg \exists x(Fx \wedge Gx))$ | (Ass.) |
| 2. | $\forall x \neg Fx$ | (1) |
| 3. | $\neg\neg \exists x(Fx \wedge Gx)$ | (1) |

# Trees for first-order predicate logic

Target: $\forall x \neg Fx \rightarrow \neg \exists x(Fx \wedge Gx)$

| | | |
|---|---|---|
| 1. | $\neg(\forall x \neg Fx \rightarrow \neg \exists x(Fx \wedge Gx))$ | (Ass.) |
| 2. | $\forall x \neg Fx$ | (1) |
| 3. | $\neg\neg \exists x(Fx \wedge Gx)$ | (1) |
| 4. | $\exists x(Fx \wedge Gx)$ | (3) |

# Trees for first-order predicate logic

Target: $\forall x \neg Fx \rightarrow \neg \exists x(Fx \land Gx)$

| | | |
|---|---|---|
| 1. | $\neg(\forall x \neg Fx \rightarrow \neg \exists x(Fx \land Gx))$ | (Ass.) |
| 2. | $\forall x \neg Fx$ | (1) |
| 3. | $\neg\neg \exists x(Fx \land Gx)$ | (1) |
| 4. | $\exists x(Fx \land Gx)$ | (3) |
| 5. | $Fa \land Ga$ | (4) |

# Trees for first-order predicate logic

Target: $\forall x \neg Fx \rightarrow \neg \exists x(Fx \wedge Gx)$

| | | |
|---|---|---|
| 1. | $\neg(\forall x \neg Fx \rightarrow \neg \exists x(Fx \wedge Gx))$ | (Ass.) |
| 2. | $\forall x \neg Fx$ | (1) |
| 3. | $\neg\neg \exists x(Fx \wedge Gx)$ | (1) |
| 4. | $\exists x(Fx \wedge Gx)$ | (3) |
| 5. | $Fa \wedge Ga$ | (4) |
| 6. | $Fa$ | (5) |
| 7. | $Ga$ | (5) |

# Trees for first-order predicate logic

Target: $\forall x \neg Fx \rightarrow \neg \exists x(Fx \wedge Gx)$

| | | |
|---|---|---|
| 1. | $\neg(\forall x \neg Fx \rightarrow \neg \exists x(Fx \wedge Gx))$ | (Ass.) |
| 2. | $\forall x \neg Fx$ | (1) |
| 3. | $\neg\neg \exists x(Fx \wedge Gx)$ | (1) |
| 4. | $\exists x(Fx \wedge Gx)$ | (3) |
| 5. | $Fa \wedge Ga$ | (4) |
| 6. | $Fa$ | (5) |
| 7. | $Ga$ | (5) |
| 8. | $\neg Fa$ | (2) |
| | x | |

# Trees for first-order predicate logic

| $\forall xA$ | $\exists xA$ | $\neg\forall xA$ | $\neg\exists xA$ |
| --- | --- | --- | --- |
| ⋮ | ⋮ | ⋮ | ⋮ |
| $A[c/x]$ | $A[c/x]$ | $\neg A[c/x]$ | $\neg A[c/x]$ |
| ↑ | ↑ | ↑ | ↑ |
| old or first | new | new | old or first |

Self-Identity:

⋮

$c = c$

↑

old

Leibniz' Law:

$b = c$

$A$

⋮

$A[c//b]$

# Trees for first-order predicate logic

Target: $\forall x \forall y((Rxy \wedge x = y) \rightarrow Rxx)$

1. $\neg \forall x \forall y((Rxy \wedge x = y) \rightarrow Rxx)$ (Ass.)

# Trees for first-order predicate logic

Target: $\forall x \forall y((Rxy \wedge x = y) \rightarrow Rxx)$

1. $\neg \forall x \forall y((Rxy \wedge x = y) \rightarrow Rxx)$ (Ass.)
2. $\neg \forall y((Ray \wedge a = y) \rightarrow Raa)$ (1)

## Trees for first-order predicate logic

Target: $\forall x \forall y((Rxy \land x = y) \to Rxx)$

| | | |
|---|---|---|
| 1. | $\neg \forall x \forall y((Rxy \land x = y) \to Rxx)$ | (Ass.) |
| 2. | $\neg \forall y((Ray \land a = y) \to Raa)$ | (1) |
| 3. | $\neg((Rab \land a = b) \to Raa)$ | (2) |

# Trees for first-order predicate logic

Target: $\forall x \forall y((Rxy \wedge x=y) \rightarrow Rxx)$

| | | |
|---|---|---|
| 1. | $\neg\forall x \forall y((Rxy \wedge x=y) \rightarrow Rxx)$ | (Ass.) |
| 2. | $\neg\forall y((Ray \wedge a=y) \rightarrow Raa)$ | (1) |
| 3. | $\neg((Rab \wedge a=b) \rightarrow Raa)$ | (2) |
| 4. | $Rab \wedge a=b$ | (3) |
| 5. | $\neg Raa$ | (3) |

# Trees for first-order predicate logic

Target: $\forall x \forall y((Rxy \wedge x=y) \rightarrow Rxx)$

| | | |
|---|---|---|
| 1. | $\neg\forall x \forall y((Rxy \wedge x=y) \rightarrow Rxx)$ | (Ass.) |
| 2. | $\neg\forall y((Ray \wedge a=y) \rightarrow Raa)$ | (1) |
| 3. | $\neg((Rab \wedge a=b) \rightarrow Raa)$ | (2) |
| 4. | $Rab \wedge a=b$ | (3) |
| 5. | $\neg Raa$ | (3) |
| 6. | $Rab$ | (4) |
| 7. | $a=b$ | (4) |

# Trees for first-order predicate logic

Target: $\forall x \forall y((Rxy \land x = y) \to Rxx)$

| | | |
|---|---|---|
| 1. | $\neg\forall x \forall y((Rxy \land x = y) \to Rxx)$ | (Ass.) |
| 2. | $\neg\forall y((Ray \land a = y) \to Raa)$ | (1) |
| 3. | $\neg((Rab \land a = b) \to Raa)$ | (2) |
| 4. | $Rab \land a = b$ | (3) |
| 5. | $\neg Raa$ | (3) |
| 6. | $Rab$ | (4) |
| 7. | $a = b$ | (4) |
| 8. | $Raa$ | (6,7,LL) |
| | x | |

# Semantics of predicate logic

The non-logical vocabulary of $\mathfrak{L}_P$ are the names and the predicates.

Intuitively:

- *Fa* is true (in a given scenario) iff the individual picked out by '*a*' has the property expressed by '*F*'.
- *Rab* is true iff the individual picked out by '*a*' stands to the individual picked out by '*b*' in the relation expressed by '*R*'.

# Semantics of predicate logic

To settle the truth-values of $\mathfrak{L}_P$-sentences in a scenario, we might specify

(a) which individuals are picked out by the names,
(b) which properties and relations are expressed by the predicates, and
(c) which individuals have which properties and stand in which relations to one another.

We can be more economical.

Instead of (b) and (c), we simply specify which predicates apply to which individuals.

# Semantics of predicate logic

$V(a)$ = Alice
$V(F)$ = { Alice, Bob }

$V(F)$ is the set of individuals to which $F$ applies.

$Fa$ is true because $V(a)$ is a member of $V(F)$.

# Semantics of predicate logic

$V(a)$ = Alice
$V(b)$ = Bob
$V(F)$ = { Alice, Bob }
$V(R) = \{ \langle Alice, Alice \rangle, \langle Bob, Alice \rangle \}$

$Raa$ is true because $\langle V(a), V(a) \rangle$ is a member of $V(R)$.

$Rab$ is false because $\langle V(a), V(b) \rangle$ is not a member of $V(R)$.

# Semantics of predicate logic

$V(a)$ = Alice
$V(F)$ = { Alice, Bob }

Is $\forall x F x$ true?

It depends on whether the scenario involves other individuals than Alice and Bob.

So we also need to specify the set of all relevant individuals in the scenario.

# Semantics of predicate logic

A **(classical) first-order model** is a pair $\langle D, V \rangle$ consisting of

- a non-empty set $D$, and
- a function $V$ that assigns
  - to each name a member of $D$,
  - to each 1-place predicate a subset of $D$,
  - to each $n$-place predicate ($n > 1$) a set of $n$-tuples from $D$.

# Semantics of predicate logic

$D = \{Alice, Bob, Carol\}$
$V(a) = \text{Alice}$
$V(b) = \text{Bob}$
$V(F) = \{\text{ Alice, Bob }\}$

Which of these are true?

1. $Fa \to Fb$
2. $\forall x Fx$
3. $\exists x Fx$
4. $\forall x \exists y (Fx \lor \neg Fy)$

$D = \{Alice, Bob, Carol\}$
$V(a) = \text{Alice}$
$V(b) = \text{Bob}$
$V(F) = \{ \text{Alice, Bob} \}$

4. $\forall x \exists y (Fx \lor \neg Fy)$

$Fx$ and $Fy$ are neither true nor false.

$Fx$ is true if $x$ picks out Alice or Bob. $Fy$ is true if $y$ picks out Alice or Bob.

$Fx \lor \neg Fy$ is true if $x$ picks out Alice or Bob or $y$ picks out Carol.

$\exists y A$ is true iff there is some interpretation of $y$ that makes $A$ true.

$\exists y (Fx \lor \neg Fy)$ is true no matter what $x$ picks out.

# Semantics of predicate logic

$\mathfrak{L}_P$-sentences are true or false relative to an interpretation of the variables.

**Semantics of first-order predicate logic**

| | | |
|---|---|---|
| (a) | $M, g \models \phi t_1 \ldots t_n$ | iff $\langle [t_1]^{M,g}, \ldots, [t_n]^{M,g} \rangle \in V(\phi)$. |
| (b) | $M, g \models s = t$ | iff $[s]^{M,g} = [t]^{M,g}$. |
| (c) | $M, g \models \neg A$ | iff $M, g \not\models A$. |
| (d) | $M, g \models A \land B$ | iff $M, g \models A$ and $M, g \models B$. |
| (e) | $M, g \models A \lor B$ | iff $M, g \models A$ or $M, g \models B$. |
| (f) | $M, g \models A \rightarrow B$ | iff $M, g \models B$ or $M, g \not\models A$. |
| (g) | $M, g \models A \leftrightarrow B$ | iff $M, g \models (A \rightarrow B)$ and $M, g \models (B \rightarrow A)$. |
| (h) | $M, g \models \forall x A$ | iff $M, g' \models A$ for all $x$-variants $g'$ of $g$. |
| (i) | $M, g \models \exists x A$ | iff $M, g' \models A$ for some $x$-variant $g'$ of $g$. |

# Modal Predicate Logic: De dicto and de re

# Modal Predicate Logic: De dicto and de re

In modal predicate logic, we can "look inside" the sentence letters of $\mathfrak{L}_M$.

It is certain that all myriapods are oviparous.
It is possible that some arthropods are myriapods.

---

It is possible that some arthropods are oviparous.

$\mathfrak{L}_M$: $\Box p, \Diamond q \therefore \Diamond r$

$\mathfrak{L}_{MP}$: $\Box \forall x(Fx \rightarrow Gx), \Diamond \exists x(Hx \wedge Fx) \therefore \Diamond \exists x(Hx \wedge Gx)$

# Modal Predicate Logic: De dicto and de re

Let $F$ mean '– win the lottery'.

- $\forall x \Diamond Fx$
- $\Diamond \forall x Fx$

$\Diamond \forall x Fx$ is **de dicto**: it assert of a proposition ($\forall x Fx$) that it is possible.

$\forall x \Diamond Fx$ is **de re**: is attributes a modal property to certain things.

# Modal Predicate Logic: De dicto and de re

- Everyone in this room might have stolen the jewels.
- $\forall x(Ixr \rightarrow \Diamond Sxj)$
- $\Diamond \forall x(Ixr \rightarrow Sxj)$

- The Russian president might have been trustworthy.
- $\Diamond \exists x(Px \land \forall y(Py \rightarrow y{=}x) \land Tx)$
- $\exists x(Px \land \forall y(Py \rightarrow y{=}x) \land \Diamond Tx)$