Mathematical Logic

Lecture 9: The Semantics of Predicate Logic

Ruizhi Yang 杨睿之

School of Philosophy, Fudan University

Summer 2026

The Semantics of Predicate Logic

We already have the shell of a complete language, and we can even use it to write inferences.

Example

$$\frac{\forall x(Sx \to Mx) \qquad \forall x(Mx \to Px)}{Sc \to Pc}$$

The Semantics of Predicate Logic

We already have the shell of a complete language, and we can even use it to write inferences.

Example

$$\frac{\exists x \forall y R x y}{\forall y \exists x R x y}$$

We still need a set of standards to determine whether an inference in this much richer language is valid.

The Semantics of Predicate Logic

We already have the shell of a complete language, and we can even use it to write inferences.

Example

$$\frac{\exists x \forall y R x y}{\forall y \exists x R x y}$$

We still need a set of standards to determine whether an inference in this much richer language is valid.

The Semantics of Predicate Logic

- In general, how do we determine the validity of an inference?
- Similar to propositional logic, we need to define the situations that the language of predicate logic can talk about.
- We need to endow the language of predicate logic with semantics .
- A hidden task: we want to convince you that predicate logic is a "universal characteristic" (characteristica universalis).

The Semantics of Predicate Logic

- In general, how do we determine the validity of an inference?
- Similar to propositional logic, we need to define the situations that the language of predicate logic can talk about.
- We need to endow the language of predicate logic with semantics .
- A hidden task: we want to convince you that predicate logic is a "universal characteristic" (characteristica universalis).

The Semantics of Predicate Logic

- In general, how do we determine the validity of an inference?
- Similar to propositional logic, we need to define the situations that the language of predicate logic can talk about.
- We need to endow the language of predicate logic with semantics .
- A hidden task: we want to convince you that predicate logic is a "universal characteristic" (characteristica universalis).

The Semantics of Predicate Logic

- In general, how do we determine the validity of an inference?
- Similar to propositional logic, we need to define the situations that the language of predicate logic can talk about.
- We need to endow the language of predicate logic with semantics .
- A hidden task: we want to convince you that predicate logic is a "universal characteristic" (characteristica universalis).

The Semantics of Predicate Logic

- Based on the choice of constants and predicates, we can have different concrete languages of predicate logic or fragments of the language of predicate logic .

Example

- A syllogism can be written in a predicate logic with only 3 unary predicates.
- The language of graphs contains only 1 binary predicate.
- In practice, we always work in a specific language of predicate logic.

The Semantics of Predicate Logic

Given a specific language of predicate logic, its semantics is a set of interpretations of the symbols in it. We call this set of interpretations a structure or model .

Example

Consider a language containing a 1-ary predicate symbol P and a 2-ary predicate symbol R. A structure or model of it must provide the necessary information for us to know what a sentence like $\forall x(\exists y Ryx \rightarrow Px)$ is saying.

The Semantics of Predicate Logic

Given a specific language of predicate logic, its semantics is a set of interpretations of the symbols in it. We call this set of interpretations a structure or model .

Example

Consider a language containing a 1-ary predicate symbol P and a 2-ary predicate symbol R. A structure or model of it must provide the necessary information for us to know what a sentence like $\forall x(\exists y Ryx \rightarrow Px)$ is saying.

The Semantics of Predicate Logic

Example

$\forall x(\exists y Ryx \rightarrow Px)$ can mean "all loved ones are happy", or it can mean "all square numbers are non-negative".

To clarify its meaning, we first need to provide the interpretation for the quantifiers $\forall x$ and $\exists x$ — the domain . The domain is a set, denoted by D.Then, P should be interpreted as a subset of D, and R should be interpreted as a (binary) relation on D.

The Semantics of Predicate Logic

Example

$\forall x(\exists y Ryx \rightarrow Px)$ can mean "all loved ones are happy", or it can mean "all square numbers are non-negative".

To clarify its meaning, we first need to provide the interpretation for the quantifiers $\forall x$ and $\exists x$ — the domain .

The domain is a set, denoted by D. Then, P should be interpreted as a subset of D, and R should be interpreted as a (binary) relation on D.

The Semantics of Predicate Logic

Example

$\forall x(\exists y Ryx \rightarrow Px)$ can mean "all loved ones are happy", or it can mean "all square numbers are non-negative".

To clarify its meaning, we first need to provide the interpretation for the quantifiers $\forall x$ and $\exists x$ — the domain .

The domain is a set, denoted by D.Then, P should be interpreted as a subset of D, and R should be interpreted as a (binary) relation on D.

The Semantics of Predicate Logic

Definition (Model or Structure)

Given a specific predicate logic language $\mathcal{L}$, we define a model or structure of $\mathcal{L}$, denoted by $\mathcal{M} = (D, I)$, as an ordered pair. Here D is a non-empty set, which is the domain of $\mathcal{M}$. I is an interpretation function: for each n-ary predicate P in $\mathcal{L}$, $I(P)$ is an n-ary relation on D; for each constant c in $\mathcal{L}$, $I(c)$ is an element in D.

The Semantics of Predicate Logic

Convention

Given a (specific predicate logic) language $\mathcal{L}$, let $\mathcal{M} = (D, I)$ be a model of $\mathcal{L}$. For a predicate P in $\mathcal{L}$, we may use P^I or $P^{\mathcal{M}}$ to denote $I(P)$. Similarly, for a constant c, we use c^I or $c^{\mathcal{M}}$ to denote $I(c)$.

The Semantics of Predicate Logic

- Given a model $\mathcal{M} = (D, I)$ of a language $\mathcal{L}$ containing predicates P, R, we now know what the sentence $\forall x(\exists y Ryx \rightarrow Px)$ means.
- From our experience with propositional logic, the meaning of a formula can be derived from the meanings of its subformulas.But what does $\exists y Ryx \rightarrow Px$ mean?

The Semantics of Predicate Logic

- Given a model $\mathcal{M} = (D, I)$ of a language $\mathcal{L}$ containing predicates P, R, we now know what the sentence $\forall x(\exists y Ryx \rightarrow Px)$ means.
- From our experience with propositional logic, the meaning of a formula can be derived from the meanings of its subformulas. But what does $\exists y Ryx \rightarrow Px$ mean?

The Semantics of Predicate Logic

- Given a model $\mathcal{M} = (D, I)$ of a language $\mathcal{L}$ containing predicates P, R, we now know what the sentence $\forall x(\exists y Ryx \rightarrow Px)$ means.
- From our experience with propositional logic, the meaning of a formula can be derived from the meanings of its subformulas.But what does $\exists y Ryx \rightarrow Px$ mean?

The Semantics of Predicate Logic

Definition (Variable Assignment)

Suppose $\mathcal{V} = \{v_0, v_1, \dots\}$ is the set of variables of a predicate logic language, and $\mathcal{M} = (D, I)$ is a model of this language. A variable assignment on $\mathcal{M}$ is a function $s : \mathcal{V} \to D$ that maps variables in $\mathcal{V}$ to elements in D.

The Semantics of Predicate Logic

Example

Given a language containing predicates P, R. Consider the model $\mathcal{M} = (D, I)$, where D = {Alice, Bob, Carol}, P^I = {Alice, Bob}, R^I = {(Bob, Carol), (Carol, Bob)}. Let the variable assignment s satisfy $s(v_0)$ = Alice, $s(v_1)$ = Bob, $s(v_2)$ = Carol. Then,

- Does $\forall x(\exists y Ryx \rightarrow Px)$ hold?
- Let $x = v_0$, what about $\exists y Ryx \rightarrow Px$?

The Semantics of Predicate Logic

Example

Given a language containing predicates P, R. Consider the model $\mathcal{M} = (D, I)$, where D = {Alice, Bob, Carol}, P^I = {Alice, Bob}, R^I = {(Bob, Carol), (Carol, Bob)}. Let the variable assignment s satisfy $s(v_0)$ = Alice, $s(v_1)$ = Bob, $s(v_2)$ = Carol. Then,

- Does $\forall x(\exists y Ryx \rightarrow Px)$ hold?
- Let $x = v_0$, what about $\exists y Ryx \rightarrow Px$?

The Semantics of Predicate Logic

Example

Given a language containing predicates P, R. Consider the model $\mathcal{M} = (D, I)$, where D = {Alice, Bob, Carol}, P^I = {Alice, Bob}, R^I = {(Bob, Carol), (Carol, Bob)}. Let the variable assignment s satisfy $s(v_0)$ = Carol, $s(v_1)$ = Bob, $s(v_2)$ = Carol. Then,

- Does $\forall x(\exists y Ryx \rightarrow Px)$ hold?
- Let $x = v_0$, what about $\exists y Ryx \rightarrow Px$?

The Semantics of Predicate Logic

To better explain the recursive semantic relationship between formulas like $\forall x(\exists y Ryx \rightarrow Px)$ and $\exists y Ryx \rightarrow Px$, we need to define the local modification of a variable assignment.

The Semantics of Predicate Logic

Definition (Local Modification of Variable Assignment)

Given a language, a model $\mathcal{M} = (D, I)$ of the language, and a variable assignment s. Suppose $x \in \mathcal{V}$ is a variable and $d \in D$, we use $s[x := d]$ to denote a variable assignment function such that for any variable $y \in \mathcal{V}$:

$$s[x := d](y) = \begin{cases} d & \text{if } y = x, \\ s(y) & \text{if } y \neq x \end{cases}$$

The Semantics of Predicate Logic

Example

Returning to the previous example. Suppose $s(v_0) = \text{Alice}$, $s(v_1) = \text{Bob}$, $s(v_2) = \text{Carol}$. Then $s[v_0 := \text{Carol}](v_0) = \text{Carol}$, $s[v_0 := \text{Carol}](v_1) = \text{Bob}$, $s[v_0 := \text{Carol}](v_2) = \text{Carol}$.

The Semantics of Predicate Logic

Given a language $\mathcal{L}$, a model $\mathcal{M} = (D, I)$ of $\mathcal{L}$, and a variable assignment $s : \mathcal{V} \to D$, we want to define the satisfaction relation:

$$\mathcal{M} \vDash_s \varphi$$

read as "(formula) φ is satisfied in (model) $\mathcal{M}$ by (assignment) s", or simply "$\mathcal{M}, s$ satisfies φ".

The Semantics of Predicate Logic

First, for convenience of writing, we unify the notation for the semantic interpretation of terms (variables or constants).

Notation

Given a language $\mathcal{L}$, a model $\mathcal{M} = (D, I)$ of $\mathcal{L}$, and a variable assignment $s : \mathcal{V} \to D$. Let t be a term, define:

$$[t]_s^I = \begin{cases} c^I & \text{if } t \text{ is a constant } c, \\ s(x) & \text{if } t \text{ is a variable } x \end{cases}$$

The Semantics of Predicate Logic

Definition (Tarski's Definition of Truth)

Given a language $\mathcal{L}$, a model $\mathcal{M} = (D, I)$ of $\mathcal{L}$, and a variable assignment $s : \mathcal{V} \to D$, we recursively define $\mathcal{M} \vDash_s \varphi$ based on the construction of the formula:

- For an atomic formula $Pt_1 \dots t_n$, $\mathcal{M} \vDash_s Pt_1 \dots t_n$ if and only if $([t_1]^I_s, \dots, [t_n]^I_s) \in P^I$, i.e., the elements $[t_1]^I_s, \dots, [t_n]^I_s$ in D have the relation P^I.

  Specifically, for the equality symbol $=$ (a special 2-ary predicate), $\mathcal{M} \vDash_s t_1 = t_2$ if and only if $[t_1]^I_s = [t_2]^I_s$.

The Semantics of Predicate Logic

Definition (Tarski's Definition of Truth)

Given a language $\mathcal{L}$, a model $\mathcal{M} = (D, I)$ of $\mathcal{L}$, and a variable assignment $s : \mathcal{V} \to D$, we recursively define $\mathcal{M} \vDash_s \varphi$ based on the construction of the formula:

- For an atomic formula $Pt_1 \dots t_n$, $\mathcal{M} \vDash_s Pt_1 \dots t_n$ if and only if $([t_1]^I_s, \dots, [t_n]^I_s) \in P^I$, i.e., the elements $[t_1]^I_s, \dots, [t_n]^I_s$ in D have the relation P^I.

  Specifically, for the equality symbol $=$ (a special 2-ary predicate), $\mathcal{M} \vDash_s t_1 = t_2$ if and only if $[t_1]^I_s = [t_2]^I_s$.

The Semantics of Predicate Logic

Definition (Tarski's Definition of Truth)

- For Boolean combinations of formulas:
  - $\mathcal{M} \vDash_s \neg\varphi$ if and only if it is not the case that $\mathcal{M} \vDash_s \varphi$ (or $\mathcal{M} \nvDash_s \varphi$).
  - $\mathcal{M} \vDash_s \varphi \wedge \psi$ if and only if $\mathcal{M} \vDash_s \varphi$ and $\mathcal{M} \vDash_s \psi$.
  - $\mathcal{M} \vDash_s \varphi \vee \psi$ if and only if $\mathcal{M} \vDash_s \varphi$ or $\mathcal{M} \vDash_s \psi$.
  - $\mathcal{M} \vDash_s \varphi \rightarrow \psi$ if and only if $\mathcal{M} \vDash_s \varphi$ implies $\mathcal{M} \vDash_s \psi$.

The Semantics of Predicate Logic

Definition (Tarski's Definition of Truth)

- For quantifiers:
  - $\mathcal{M} \vDash_s \forall x \varphi$ if and only if for any $d \in D$, $\mathcal{M} \vDash_{s[x:=d]} \varphi$ holds.
  - $\mathcal{M} \vDash_s \exists x \varphi$ if and only if there exists $d \in D$ such that $\mathcal{M} \vDash_{s[x:=d]} \varphi$ holds.

The Semantics of Predicate Logic

This is a recursive definition based on the construction of the formula φ, which tells us step by step how to determine whether a formula holds.

Example

Returning to the previous example. D = {Alice, Bob, Carol}, P^I = {Alice, Bob}, R^I = {(Bob, Carol), (Carol, Bob)}. $s(v_0)$ = Alice, $s(v_1)$ = Bob, $s(v_2)$ = Carol. Let $x = v_0$, $y = v_1$.

- $\mathcal{M} \vDash_s Px$? $\mathcal{M} \vDash_{s[x:=\text{Bob}]} Px$? $\mathcal{M} \vDash_s Ryx$? $\mathcal{M} \vDash_s Ryx \to Px$?
- $\mathcal{M} \vDash_s \exists y Ryx \to Px$? $\mathcal{M} \vDash_s \forall x(\exists y Ryx \to Px)$?

The Semantics of Predicate Logic

This is a recursive definition based on the construction of the formula φ, which tells us step by step how to determine whether a formula holds.

Example

Returning to the previous example. D = {Alice, Bob, Carol}, P^I = {Alice, Bob}, R^I = {(Bob, Carol), (Carol, Bob)}. $s(v_0)$ = Alice, $s(v_1)$ = Bob, $s(v_2)$ = Carol. Let $x = v_0$, $y = v_1$.

- $\mathcal{M} \vDash_s Px ? \mathcal{M} \vDash_{s[x:=\text{Bob}]} Px ?$ $\mathcal{M} \vDash_s Ryx ? \mathcal{M} \vDash_s Ryx \to Px ?$
- $\mathcal{M} \vDash_s \exists y Ryx \to Px ? \mathcal{M} \vDash_s \forall x(\exists y Ryx \to Px) ?$

The Semantics of Predicate Logic

This is a recursive definition based on the construction of the formula φ, which tells us step by step how to determine whether a formula holds.

Example

Returning to the previous example. D = {Alice, Bob, Carol}, P^I = {Alice, Bob}, R^I = {(Bob, Carol), (Carol, Bob)}. $s(v_0)$ = Alice, $s(v_1)$ = Bob, $s(v_2)$ = Carol. Let $x = v_0$, $y = v_1$.

- $\mathcal{M} \vDash_s Px ? \mathcal{M} \vDash_{s[x:=\text{Bob}]} Px ?$ $\mathcal{M} \vDash_s Ryx ? \mathcal{M} \vDash_s Ryx \to Px ?$
- $\mathcal{M} \vDash_s \exists y Ryx \to Px ? \mathcal{M} \vDash_s \forall x(\exists y Ryx \to Px) ?$

The Semantics of Predicate Logic

This is a recursive definition based on the construction of the formula φ, which tells us step by step how to determine whether a formula holds.

Example

Returning to the previous example. D = {Alice, Bob, Carol}, P^I = {Alice, Bob}, R^I = {(Bob, Carol), (Carol, Bob)}. $s(v_0)$ = Alice, $s(v_1)$ = Bob, $s(v_2)$ = Carol. Let $x = v_0$, $y = v_1$.

- $\mathcal{M} \vDash_s Px ? \mathcal{M} \vDash_{s[x:=\text{Bob}]} Px ?$ $\mathcal{M} \vDash_s Ryx ? \mathcal{M} \vDash_s Ryx \to Px ?$
- $\mathcal{M} \vDash_s \exists y Ryx \to Px ? \mathcal{M} \vDash_s \forall x(\exists y Ryx \to Px) ?$

The Semantics of Predicate Logic

This is a recursive definition based on the construction of the formula φ, which tells us step by step how to determine whether a formula holds.

Example

Returning to the previous example. D = {Alice, Bob, Carol}, P^I = {Alice, Bob}, R^I = {(Bob, Carol), (Carol, Bob)}. $s(v_0)$ = Alice, $s(v_1)$ = Bob, $s(v_2)$ = Carol. Let $x = v_0$, $y = v_1$.

- $\mathcal{M} \vDash_s Px ? \mathcal{M} \vDash_{s[x:=\text{Bob}]} Px ?$ $\mathcal{M} \vDash_s Ryx ? \mathcal{M} \vDash_s Ryx \to Px ?$
- $\mathcal{M} \vDash_s \exists y Ryx \to Px ? \mathcal{M} \vDash_s \forall x(\exists y Ryx \to Px) ?$

The Semantics of Predicate Logic

This is a recursive definition based on the construction of the formula φ, which tells us step by step how to determine whether a formula holds.

Example

Returning to the previous example. D = {Alice, Bob, Carol}, P^I = {Alice, Bob}, R^I = {(Bob, Carol), (Carol, Bob)}. $s(v_0)$ = Alice, $s(v_1)$ = Bob, $s(v_2)$ = Carol. Let $x = v_0$, $y = v_1$.

- $\mathcal{M} \vDash_s Px ? \mathcal{M} \vDash_{s[x:=\text{Bob}]} Px ?$ $\mathcal{M} \vDash_s Ryx ? \mathcal{M} \vDash_s Ryx \to Px ?$
- $\mathcal{M} \vDash_s \exists y Ryx \to Px ? \mathcal{M} \vDash_s \forall x(\exists y Ryx \to Px) ?$

The Semantics of Predicate Logic

Example

Let $\mathcal{L}$ be a language with two constants c_0, c_1, the equality symbol $=$, and a binary predicate D. Consider a model $\mathcal{M} = (\mathbb{Z}, I)$ where the domain is the set of integers $\mathbb{Z}$, $c_0^I = 0$, $c_1^I = 1$, and $D^I = \{(a, b) \mid a \text{ divides } b\}$. Let s be an assignment where $s(x) = 2$.

- $\mathcal{M} \vDash_s Dc_1x$
- $\mathcal{M} \vDash_s \forall y(Dyc_0)$
- $\mathcal{M} \nvDash_s \forall y(Dyc_1 \rightarrow y = c_1)$

The Semantics of Predicate Logic

Note that in Tarski's definition, the interpretation of the equality symbol is strictly fixed to true identity across all models, rather than being open to arbitrary interpretation.

The Semantics of Predicate Logic

The definitions in Tarski's definition of truth regarding whether conjunctions, disjunctions, and implications hold seem to rely on our intuitive understanding of "and", "or", and "implies".We can understand this definition as telling us how to determine whether a compound formula is satisfied based on the following truth table:

$\mathcal{M} \vDash_s \varphi$	$\mathcal{M} \vDash_s \psi$	$\mathcal{M} \vDash_s \varphi \rightarrow \psi$
Yes	Yes	Yes
Yes	No	No
No	Yes	Yes
No	No	Yes

The Semantics of Predicate Logic

The definitions in Tarski's definition of truth regarding whether conjunctions, disjunctions, and implications hold seem to rely on our intuitive understanding of "and", "or", and "implies".We can understand this definition as telling us how to determine whether a compound formula is satisfied based on the following truth table:

$\mathcal{M} \vDash_s \varphi$	$\mathcal{M} \vDash_s \psi$	$\mathcal{M} \vDash_s \varphi \rightarrow \psi$
Yes	Yes	Yes
Yes	No	No
No	Yes	Yes
No	No	Yes

The Semantics of Predicate Logic

- Observation: According to Tarski's definition of truth, if variable assignments s_1, s_2 assign the same values to the variables occurring in formula φ, then $\mathcal{M} \vDash_{s_1} \varphi$ if and only if $\mathcal{M} \vDash_{s_2} \varphi$.

The Semantics of Predicate Logic

- Although a variable assignment s can involve assignments to infinitely many variables, when actually considering one or finitely many formulas, we only use the assignments s makes to the finitely many variables occurring in them.
- The condition of the above observation can be further relaxed to: "If s_1, s_2 assign the same values to the free variables occurring in formula φ".

The Semantics of Predicate Logic

- Although a variable assignment s can involve assignments to infinitely many variables, when actually considering one or finitely many formulas, we only use the assignments s makes to the finitely many variables occurring in them.
- The condition of the above observation can be further relaxed to: "If s_1, s_2 assign the same values to the free variables occurring in formula φ".

The Semantics of Predicate Logic

Therefore, if φ is a sentence, then either for all variable assignments s we have $\mathcal{M} \vDash_s \varphi$, or for all variable assignments we have $\mathcal{M} \nvDash_s \varphi$.

Definition

Suppose φ is a sentence. If for all variable assignments s we have $\mathcal{M} \vDash_s \varphi$, we say that φ is true in $\mathcal{M}$, denoted by $\mathcal{M} \vDash \varphi$.

The Semantics of Predicate Logic

Example

Let $\mathcal{M} = (\mathbb{Z}, I)$ where $D^I = \{(a, b) \mid a \text{ divides } b\}$.

- Consider $\psi = \exists y(Dxy)$. The only free variable is x. If $s_1(x) = s_2(x) = 2$, then even if $s_1(y) = 3$ and $s_2(y) = 5$, both assignments satisfy ψ because they agree on the free variable x.
- Consider the sentence $\chi = \forall x \exists y(Dxy)$. It has no free variables. Its truth value is completely independent of the assignment. Since it holds in $\mathbb{Z}$, we simply write $\mathcal{M} \vDash \chi$.

The Semantics of Predicate Logic

Example

Let $\mathcal{M} = (\mathbb{Z}, I)$ where $D^I = \{(a, b) \mid a \text{ divides } b\}$.

- Consider $\psi = \exists y(Dxy)$. The only free variable is x. If $s_1(x) = s_2(x) = 2$, then even if $s_1(y) = 3$ and $s_2(y) = 5$, both assignments satisfy ψ because they agree on the free variable x.
- Consider the sentence $\chi = \forall x \exists y(Dxy)$. It has no free variables. Its truth value is completely independent of the assignment. Since it holds in $\mathbb{Z}$, we simply write $\mathcal{M} \vDash \chi$.

The Semantics of Predicate Logic

Further Observation: How does whether $\forall x(\exists y Ryx \rightarrow Px)$ holds recursively depend on whether its subformula $\exists y Ryx \rightarrow Px$ holds?

- To test whether $\forall x(\exists y Ryx \rightarrow Px)$ holds, we need to test for each $d \in D = \{\text{Alice, Bob, Carol}\}$ whether $\mathcal{M} \vDash_{s[x:=d]} \exists y Ryx \rightarrow Px$ holds.
- When D is a finite set, this is feasible. When D is infinite, this might not be the case.

The Semantics of Predicate Logic

Further Observation: How does whether $\forall x(\exists y Ryx \rightarrow Px)$ holds recursively depend on whether its subformula $\exists y Ryx \rightarrow Px$ holds?

- To test whether $\forall x(\exists y Ryx \rightarrow Px)$ holds, we need to test for each $d \in D$ = {Alice, Bob, Carol} whether $\mathcal{M} \vDash_{s[x:=d]} \exists y Ryx \rightarrow Px$ holds.
- When D is a finite set, this is feasible. When D is infinite, this might not be the case.

The Semantics of Predicate Logic

Definition (Validity)

Given a predicate logic language $\mathcal{L}$:

- We say that an $\mathcal{L}$-formula φ is logically valid or valid (denoted by $\vDash \varphi$), if and only if for any model $\mathcal{M}$ of $\mathcal{L}$ and any variable assignment s on $\mathcal{M}$, we have $\mathcal{M} \vDash_s \varphi$.

The Semantics of Predicate Logic

Definition (Validity)

Given a predicate logic language $\mathcal{L}$:

- Suppose Σ is a set of $\mathcal{L}$-formulas and φ is an $\mathcal{L}$-formula. We say that Σ logically implies φ (denoted by $\Sigma \vDash \varphi$), if and only if for any model $\mathcal{M}$ of $\mathcal{L}$ and any variable assignment s on $\mathcal{M}$, if for every formula ψ in Σ we have $\mathcal{M} \vDash_s \psi$, then $\mathcal{M} \vDash_s \varphi$.

The Semantics of Predicate Logic

- $\Sigma \models \varphi$ means the inference from Σ to φ is valid.

  Conversely, if $\Sigma \not\models \varphi$, then there exists a model $\mathcal{M}$ and an assignment s that satisfies every formula in Σ but does not satisfy φ.

- Note: We use the symbol $\models$ in both $\mathcal{M} \models_s \varphi$ and $\Sigma \models \varphi$. This is an unfortunate choice of notation, please be careful to distinguish them.

- $\models \varphi$ means $\emptyset \models \varphi$.

The Semantics of Predicate Logic

- $\Sigma \models \varphi$ means the inference from Σ to φ is valid.

  Conversely, if $\Sigma \not\models \varphi$, then there exists a model $\mathcal{M}$ and an assignment s that satisfies every formula in Σ but does not satisfy φ.
- Note: We use the symbol $\models$ in both $\mathcal{M} \models_s \varphi$ and $\Sigma \models \varphi$. This is an unfortunate choice of notation, please be careful to distinguish them.
- $\models \varphi$ means $\emptyset \models \varphi$.

The Semantics of Predicate Logic

- $\Sigma \models \varphi$ means the inference from Σ to φ is valid.

  Conversely, if $\Sigma \not\models \varphi$, then there exists a model $\mathcal{M}$ and an assignment s that satisfies every formula in Σ but does not satisfy φ.
- Note: We use the symbol $\models$ in both $\mathcal{M} \models_s \varphi$ and $\Sigma \models \varphi$. This is an unfortunate choice of notation, please be careful to distinguish them.
- $\models \varphi$ means $\emptyset \models \varphi$.

The Semantics of Predicate Logic

How do we determine whether $\Sigma \vDash \varphi$ holds? Unfortunately, in predicate logic, unlike propositional logic and monadic predicate logic, there is no general mechanical method. This is not because logicians did not try hard enough. On the contrary, logicians have proven that no such method exists. However, in many specific cases, we can still determine whether it holds.

The Semantics of Predicate Logic

Example

- $\forall x Px \models \exists x Px$
- $\exists x Px \models \forall x Px$
- We can provide a specific model as a counterexample to prove that $\Sigma \not\models \varphi$.
- To argue that $\Sigma \models \varphi$ holds, we often need to provide a proof from Σ to φ.

The Semantics of Predicate Logic

Example

- $\forall x Px \models \exists x Px$
- $\exists x Px \models \forall x Px$
- We can provide a specific model as a counterexample to prove that $\Sigma \not\models \varphi$.
- To argue that $\Sigma \models \varphi$ holds, we often need to provide a proof from Σ to φ.

The Semantics of Predicate Logic

Example

- $\forall x Px \models \exists x Px$
- $\exists x Px \models \forall x Px$
- We can provide a specific model as a counterexample to prove that $\Sigma \not\models \varphi$.
- To argue that $\Sigma \models \varphi$ holds, we often need to provide a proof from Σ to φ.

Formal Proofs

Recall: When discussing propositional logic, we defined an Hilbert-style axiomatic system for propositional logic .

- Axioms:

  (P1) $\varphi \to \psi \to \varphi$

  (P2) $(\varphi \to \psi \to \chi) \to (\varphi \to \psi) \to (\varphi \to \chi)$

  (P3) $(\neg\varphi \to \psi) \to (\neg\varphi \to \neg\psi) \to \varphi$

- Deduction Rule: If φ and $\varphi \to \psi$ are theorems, then ψ is also a theorem. (Modus Ponens)

Formal Proofs

- To provide an axiomatic system for predicate logic, we need to specify its set of axioms and deduction rules .
- From this, we can define $\Sigma \vdash \varphi$, meaning there exists a finite sequence of formulas ending in φ, where each formula is either an axiom, a formula in Σ, or obtained from previous formulas by applying deduction rules. We call such a sequence a proof sequence from Σ to φ .

Formal Proofs

Set of Axioms:

1. All propositional logic tautologies are axioms.
2. $\forall x \varphi \rightarrow \varphi_t^x$ Condition: If t is a variable y, the occurrences of y replacing x in φ_y^x must not be bound by any existing $\forall y$ or $\exists y$ in φ.
3. $\forall x(\varphi \rightarrow \psi) \rightarrow (\forall x \varphi \rightarrow \forall x \psi)$
4. $\varphi \rightarrow \forall x \varphi$ Condition: x does not occur freely in φ.

Formal Proofs

Deduction Rules:

1. Modus Ponens (MP): If a proof sequence already contains formulas φ and $\varphi \to \psi$, we can append ψ to the sequence, and it remains a proof sequence.
2. Universal Generalization (UG): If there is a proof sequence from Σ to φ, and x does not occur freely in any formula of Σ, we can append $\forall x \varphi$ to the sequence, and it remains a proof sequence.

Formal Proofs

If the language contains the equality symbol , we can add two equality axioms :

- $x = x$
- $x = y \to \varphi \to \varphi'$ Condition: φ is a quantifier-free formula, and φ' is the formula obtained by replacing any number of occurrences of x in φ with y.

Formal Proofs

- Similar to propositional logic, what we provided are "axiom schemas", meaning "any formula of the form... is an axiom".
- The set of axioms for predicate logic is effectively decidable , meaning there exists a computer program that can determine whether an arbitrary string is an axiom.
- In particular, there is a program that can effectively reduce a predicate logic formula to a familiar propositional logic formula and determine if it is a tautology.

Formal Proofs

- The condition for the 2nd group of axioms $\forall x\varphi \to \varphi_t^x$ is necessary.
- The condition for the 4th group of axioms $\varphi \to \forall x\varphi$ is also necessary.

Example

- Consider: $\forall x \exists y Rxy \to \exists y Ryy$
- Consider: $Px \to \forall x Px$

Formal Proofs

- The symbols $\vee, \wedge, \leftrightarrow$ and $\exists$ do not appear in the axiomatic system; they can be viewed as being defined.
- By Universal Generalization, if φ is an axiom, then $\vdash \forall x\varphi$, $\vdash \forall y\forall x\varphi$, $\vdash \forall z\forall y\forall x\varphi$, etc.
- Soundness of the axiomatic system: $\Sigma \vdash \varphi$ implies $\Sigma \vDash \varphi$. In particular, all axioms are valid, and the deduction rules preserve logical implication.

Formal Proofs

Example

We prove: $\vdash \varphi_t^x \rightarrow \exists x \varphi$

- $\forall x \neg \varphi \rightarrow \neg \varphi_t^x$ (Axiom 2)
- $(\forall x \neg \varphi \rightarrow \neg \varphi_t^x) \rightarrow (\varphi_t^x \rightarrow \neg \forall x \neg \varphi)$ (Axiom 1)
- $\varphi_t^x \rightarrow \neg \forall x \neg \varphi$ (MP)

Formal Proofs

Example

We prove: $\vdash \varphi_t^x \to \exists x \varphi$

- $\forall x \neg \varphi \to \neg \varphi_t^x$ (Axiom 2)
- $(\forall x \neg \varphi \to \neg \varphi_t^x) \to (\varphi_t^x \to \neg \forall x \neg \varphi)$ (Axiom 1)
- $\varphi_t^x \to \neg \forall x \neg \varphi$ (MP)

Formal Proofs

Example

We prove: $\vdash \varphi_t^x \to \exists x \varphi$

- $\forall x \neg \varphi \to \neg \varphi_t^x$ (Axiom 2)
- $(\forall x \neg \varphi \to \neg \varphi_t^x) \to (\varphi_t^x \to \neg \forall x \neg \varphi)$ (Axiom 1)
- $\varphi_t^x \to \neg \forall x \neg \varphi$ (MP)

Formal Proofs

Example

We prove: $\vdash \varphi_t^x \to \exists x \varphi$

- $\forall x \neg \varphi \to \neg \varphi_t^x$ (Axiom 2)
- $(\forall x \neg \varphi \to \neg \varphi_t^x) \to (\varphi_t^x \to \neg \forall x \neg \varphi)$ (Axiom 1)
- $\varphi_t^x \to \exists x \varphi$ (Definition)

Formal Proofs

Example

Suppose x does not occur freely in φ, then

$\varphi \to \forall x\psi \vdash \forall x(\varphi \to \psi)$.

1. $\varphi \to \forall x\psi$ (Premise)
2. $\forall x\psi \to \psi$ (Axiom 2)
3. $(\varphi \to \forall x\psi) \to (\forall x\psi \to \psi) \to (\varphi \to \psi)$ (Axiom 1)
4. $\varphi \to \psi$ (MP twice)
5. $\forall x(\varphi \to \psi)$ (UG)

Lines 1-4 form a proof sequence from $\varphi \to \forall x\psi$ to $\varphi \to \psi$, and x

Formal Proofs

Example

Suppose x does not occur freely in φ, then

$\varphi \to \forall x\psi \vdash \forall x(\varphi \to \psi)$.

1. $\varphi \to \forall x\psi$ (Premise)
2. $\forall x\psi \to \psi$ (Axiom 2)
3. $(\varphi \to \forall x\psi) \to (\forall x\psi \to \psi) \to (\varphi \to \psi)$ (Axiom 1)
4. $\varphi \to \psi$ (MP twice)
5. $\forall x(\varphi \to \psi)$ (UG)

Lines 1-4 form a proof sequence from $\varphi \to \forall x\psi$ to $\varphi \to \psi$, and x

Natural Deduction System

Propositional Logic Rules:

- $\wedge$ Intro: $\Gamma \vdash \varphi$ and $\Gamma \vdash \psi \Rightarrow \Gamma \vdash \varphi \wedge \psi$
- $\wedge$ Elim: $\Gamma \vdash \varphi \wedge \psi \Rightarrow \Gamma \vdash \varphi \quad \Gamma \vdash \varphi \wedge \psi \Rightarrow \Gamma \vdash \psi$
- $\vee$ Intro: $\Gamma \vdash \varphi \Rightarrow \Gamma \vdash \varphi \vee \psi \quad \Gamma \vdash \psi \Rightarrow \Gamma \vdash \varphi \vee \psi$
- $\vee$ Elim: $\Gamma \vdash \varphi \vee \psi$, $\Gamma, \varphi \vdash \chi$ and $\Gamma, \psi \vdash \chi \Rightarrow \Gamma \vdash \chi$
- $\rightarrow$ Intro: $\Gamma, \varphi \vdash \psi \Rightarrow \Gamma \vdash \varphi \rightarrow \psi$
- $\rightarrow$ Elim: $\Gamma \vdash \varphi$ and $\Gamma \vdash \varphi \rightarrow \psi \Rightarrow \Gamma \vdash \psi$
- $\neg$ Intro: $\Gamma, \varphi \vdash \psi$ and $\Gamma, \varphi \vdash \neg\psi \Rightarrow \Gamma \vdash \neg\varphi$
- $\neg$ Elim: $\Gamma, \neg\varphi \vdash \psi$ and $\Gamma, \neg\varphi \vdash \neg\psi \Rightarrow \Gamma \vdash \varphi$

Natural Deduction System

Regarding $\forall$:

$$\frac{\begin{array}{c}\vdots\\ \varphi_t^x\end{array}}{\forall x\varphi}\ \forall \text{ Intro}$$

Condition: Term t does not occur in φ, nor does it occur in any unclosed assumptions in the proof tree of φ_t^x, i.e., it does not occur in Γ.

$\Gamma \vdash \varphi_t^x \Rightarrow \Gamma \vdash \forall x\varphi$

Natural Deduction System

Regarding $\forall$:

$$\frac{\begin{array}{c}\vdots\\ \varphi_t^x\end{array}}{\forall x\varphi}\ \forall \text{ Intro}$$

$\Gamma \vdash \varphi_t^x \Rightarrow \Gamma \vdash \forall x\varphi$

Condition: Term t does not occur in φ, nor does it occur in any unclosed assumptions in the proof tree of φ_t^x, i.e., it does not occur in Γ.

Natural Deduction System

Regarding $\forall$:

$$\frac{\begin{array}{c}\vdots\\ \forall x\varphi\end{array}}{\varphi_t^x}\ \forall \text{ Elim}$$

$$\Gamma \vdash \forall x\varphi \Rightarrow \Gamma \vdash \varphi_t^x$$

Natural Deduction System

Regarding $\exists$:

$$\frac{\begin{array}{c}\vdots\\ \varphi_t^x\end{array}}{\exists x\varphi}\ \exists \text{ Intro}$$

$$\Gamma \vdash \varphi_t^x \Rightarrow \Gamma \vdash \exists x\varphi$$

Natural Deduction System

Regarding $\exists$:

$$\dfrac{\exists x\varphi \qquad \begin{array}{c}[\varphi_t^x] \\ \vdots \\ \psi\end{array}}{\psi}\ \exists \text{ Elim}$$

Condition: Term t does not occur in $\exists x\varphi$, ψ, or in any unclosed assumptions in the proof tree of φ_t^x (i.e., it does not occur in Γ).

$\Gamma \vdash \exists x\varphi$ and $\Gamma, \varphi_t^x \vdash \psi \Rightarrow \Gamma \vdash \psi$

Natural Deduction System

Regarding =:

$$\frac{\begin{array}{c}[t = t] \\ \vdots \end{array}}{\Gamma, t = t \vdash \varphi} = \text{Intro}$$

$$\Gamma, t = t \vdash \varphi \Rightarrow \Gamma \vdash \varphi$$

Natural Deduction System

Regarding $=$: φ is a quantifier-free formula

$$\frac{\begin{array}{cc}\vdots & \vdots \\ \varphi_s^x & s = t\end{array}}{\varphi_t^x} = \text{Elim (L)} \qquad \frac{\begin{array}{cc}\vdots & \vdots \\ \varphi_t^x & s = t\end{array}}{\varphi_s^x} = \text{Elim (R)}$$

$\Gamma \vdash \varphi_s^x$ and $\Gamma \vdash s = t \Rightarrow \Gamma \vdash \varphi_t^x$

$\Gamma \vdash \varphi_t^x$ and $\Gamma \vdash s = t \Rightarrow \Gamma \vdash \varphi_s^x$

Natural Deduction System

Example

- $\vdash (p \to q) \to ((p \wedge r) \to (q \wedge r))$
- $\forall x \exists y (Pxy \wedge \neg Qxy) \vdash \forall x \exists y Pxy$
- $\exists x (Px \wedge Qx), \neg \exists x (Qx \wedge Rx) \vdash \exists x (Px \wedge \neg Rx)$
- $\vdash \exists x \exists y \; x = y$

Exercises & Discussion 9.1

Do the following hold?

1. $\vDash \exists x Px \vee \neg \exists x Px$
2. $\vDash \exists x Px \vee \forall x \neg Px$
3. $\vDash \exists x \exists y Rxy \rightarrow \exists x Rxx$
4. $\vDash \forall x \forall y Rxy \rightarrow \forall x Rxx$

Exercises & Discussion 9.2

Do the following hold?

5. $\vDash \exists x \exists y Rxy \rightarrow \exists x \exists y Ryx$
6. $\vDash \forall x Rxx \rightarrow \forall x \exists y Rxy$
7. $\vDash \forall x Rxy \rightarrow \forall x \exists y Rxy$
8. $\vDash \forall x \exists y Rxy \rightarrow \forall x Rxy$

Exercises & Discussion 9.3

Prove that for any formulas φ, ψ:

$$\varphi \vDash \psi \text{ if and only if } \vDash \varphi \to \psi$$

When only finitely many formulas are involved, we often use $\varphi_1, \ldots, \varphi_n \vDash \psi$ as an abbreviation for $\{\varphi_1, \ldots, \varphi_n\} \vDash \psi$.

Exercises & Discussion 9.4

Do the following hold?

1. $\forall x Px \vDash \exists x Px$
2. $\forall x \forall y (Rxy \rightarrow Ryx), Rab \vDash Rba$
3. $\forall x \forall y (Rxy \rightarrow Ryx), Rab \vDash Raa$
4. $\forall x \forall y \forall z (Rxy \rightarrow Ryz \rightarrow Rxz), Rab, \neg Rac \vDash \neg Rbc$

Exercises & Discussion 9.5

- If $\forall y \varphi_y^x$ is an alphabetical variant (change of bound variable) of $\forall x \varphi$, prove:
  - $\forall y \varphi_y^x \vdash \forall x \varphi$
  - $\vdash \forall y \varphi_y^x \rightarrow \forall x \varphi$
- Prove: $\vdash \exists x \forall y \varphi \rightarrow \forall y \exists x \varphi$