COMP2111 Week 4
Term 1, 2019
Predicate Logic II

Summary of topics

- Re-introduction to Predicate Logic
- Syntax of Predicate Logic
- Semantics of Predicate Logic
- Natural Deduction for Predicate Logic

Vocabulary

A **vocabulary** indicates what predicates, functions and constants we can use to build up our formulas. Very similar to C header files, or Java interfaces or *database schemas*.

A vocabulary V is a set of:

- Predicate "symbols" P, Q, ..., each with an assoicated *arity* (number of arguments)
- Function "symbols" f, g, ..., each with an assoicated *arity* (number of arguments)
- Constant "symbols" c, d, ... (also known as 0-arity functions)

Example

$V = \{\leq, +, 1\}$ where $\leq$ is a binary predicate symbol, $+$ is a binary function symbol, and 1 is a constant symbol.

Vocabulary: example (databases)

Example

A database schema identifies the various tables, their attributes, and their attributes' types. For example:

Person	
Name:	String
Surname:	String
Address:	String

Employee	
ID:	int
Surname:	String

Tables *relate* a number of attributes

The above schema would be represented by the vocabulary:

$$DB = \{\text{Person}, \text{Employee}\}$$

where Person is a ternary predicate symbol and Employee is a binary predicate symbol

Vocabulary: example (databases)

Example

A database schema identifies the various tables, their attributes, and their attributes' types. For example:

Person	
Name:	String
Surname:	String
Address:	String

Employee	
ID:	int
Surname:	String

Tables *relate* a number of attributes (over several domains). The above schema would be represented by the vocabulary:

$$DB = \{\text{Person}, \text{Employee}\}$$

where Person is a ternary predicate symbol and Employee is a binary predicate symbol and isString and isInteger are unary predicate symbols.

Vocabulary: example (databases)

Example

A database schema identifies the various tables, their attributes, and their attributes' types. For example:

Person	
Name:	String
Surname:	String
Address:	String

Employee	
ID:	int
Surname:	String

Tables *relate* a number of attributes (over several domains). The above schema would be represented by the vocabulary:

$$DB = \{\mathsf{Person}, \mathsf{Employee}, \mathsf{isString}, \mathsf{isInteger}\}$$

where Person is a ternary predicate symbol and Employee is a binary predicate symbol and isString and isInteger are unary predicate symbols.

Terms

A **term** is defined recursively as follows:

- A variable is a term
- A constant symbol is a term
- If f is a function symbol with arity k, and t_1, ..., t_k are terms, then $f(t_1, t_2, \ldots, t_k)$ is a term.

NB

Terms will be interpreted as elements of the domain of discourse.

Terms: examples

Example

Over $V = \{\leq, +, 1\}$, the following are all terms:

- x
- 1
- $+(y, 1)$
- $+(y, +(x, 1))$

Formulas

A **formula of Predicate Logic** is defined recursively as follows:

- If P is a predicate symbol with arity k, and t_1, …, t_k are terms, then $P(t_1, t_2, \ldots, t_k)$ is a formula
- If t_1 and t_2 are terms then $(t_1 = t_2)$ is a formula
- If φ, ψ are a formulas then the following are formulas:
  - $\neg\varphi$
  - $(\varphi \wedge \psi)$
  - $(\varphi \vee \psi)$
  - $(\varphi \rightarrow \psi)$
  - $(\varphi \leftrightarrow \psi)$
  - $\forall x \varphi$
  - $\exists x \varphi$

NB

The base cases are known as **atomic** *formulas: they play a similar role in the parse tree as propositional variables.*

Formulas: examples

Example

Over $V = \{\leq, +, 1\}$, the following are all formulas:

- $\leq (x, y)$
- $\leq (1, 1)$
- $x = +(y, 1)$
- $\leq (x, y) \rightarrow (x = +(y, 1))$
- $\exists x(1 = +(1, 1))$
- $\forall x \forall y \ \leq (x, y) \rightarrow (x = +(y, 1))$

Formulas: example (databases)

In relational databases, formulas correspond to (select-)queries.

Example

For the vocabulary
$DB = \{\mathsf{Person}, \mathsf{Employee}, \mathsf{isString}, \mathsf{isInteger}\}$:

```
Select *
   from Person
```

$$\downarrow$$

$$\mathsf{Person}(x, y, z)$$

Formulas: example (databases)

In relational databases, formulas correspond to (select-)queries.

Example

For the vocabulary

$DB = \{\text{Person}, \text{Employee}, \text{isString}, \text{isInteger}, \text{Arya}\}$:

```
Select *
   from Person
   where Person.name = "Arya"
```

$\downarrow$

$\text{Person}(x, y, z) \wedge (x = \text{Arya})$

Formulas: example (databases)

In relational databases, formulas correspond to (select-)queries.

Example

For the vocabulary
$DB = \{\text{Person}, \text{Employee}, \text{isString}, \text{isInteger}, \text{Arya}\}$:

```
Select Person.surname,Person.address
   from Person
   where Person.name = "Arya"
```

↓

$\text{Person}(Arya, y, z)$

Formulas: example (databases)

In relational databases, formulas correspond to (select-)queries.

Example

For the vocabulary
$DB = \{\text{Person}, \text{Employee}, \text{isString}, \text{isInteger}, \text{Arya}\}$:

```
Select *
   from Person inner join Employee
   on Person.surname = Employee.surname
```

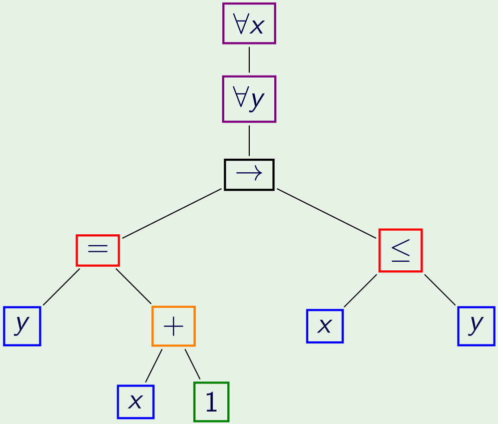
$\downarrow$

$\text{Person}(x, y, z) \vee \text{Employee}(w, y)$

Parse trees

Example

$$\forall x \forall y((y = x + 1) \rightarrow (x \leq y))$$



Free and Bound variables

A variable is **bound** to the closest matching quantifier that lies above it in the parse tree. A variable that is not bound is **free**.

Example

In $\varphi = \forall x \exists z \exists x P(x, y, z)) \wedge Q(x)$:

- z is bound to $\exists z$
- y is free
- First x is bound to $\exists x$
- Second x is free

A formula with no free variables is a **sentence**.

It can be useful to have "access" to the free variables of a formula. So if $x_1, \ldots, x_k$ are the free variables of φ, we may denote this as $\varphi(x_1, \ldots, x_k)$.

Free and Bound variables

A variable is **bound** to the closest matching quantifier that lies above it in the parse tree. A variable that is not bound is **free**.

Example

In $\varphi = \forall x \exists z \exists x P(x, y, z)) \wedge Q(x)$:

- z is bound to $\exists z$
- y is free
- First x is bound to $\exists x$
- Second x is free

A formula with no free variables is a **sentence**.

It can be useful to have "access" to the free variables of a formula. So if $x_1, \ldots, x_k$ are the free variables of φ, we may denote this as $\varphi(x_1, \ldots, x_k)$.

Free and Bound variables

A variable is **bound** to the closest matching quantifier that lies above it in the parse tree. A variable that is not bound is **free**.

Example

In $\varphi(x, y) = \forall x \exists z \exists x P(x, y, z)) \wedge Q(x)$:

- z is bound to $\exists z$
- y is free
- First x is bound to $\exists x$
- Second x is free

A formula with no free variables is a **sentence**.

It can be useful to have "access" to the free variables of a formula. So if $x_1, \ldots, x_k$ are the free variables of φ, we may denote this as $\varphi(x_1, \ldots, x_k)$.

Formulas as predicates

Formulas can be viewed as complex predicates: predicates that are built from other predicates, either by

- Combining them using the boolean operators
- "Simplifying" using quantification and term substitutions (**projection**)

The free variables represent the arity of the predicate, hence the notation $\varphi(x_1, \ldots, x_k)$.

Note

Variable names matter: $\varphi(x)$ and $\varphi(y)$ are different formulas! However, they will be *interpreted* as the same predicate.

Formulas as predicates

Example

From binary predicates P and Q; and constant c we can build complex predicates like:

- $\alpha(w, x, y, z) = (P(x, w) \vee Q(y, z)) \wedge (w = y) \wedge (z = c)$
- $\beta(x, y, z) = (P(x, y) \vee Q(y, z)) \wedge (z = c)$
- $\gamma(x, y) = P(x, y) \vee Q(y, c)$
- $\delta(x) = \exists y P(x, y) \vee \exists y Q(y, c)$

NB

α, β and γ are different predicates: α represents a 4-ary predicate, whereas β represents a 3-ary predicate and γ represents a binary predicate.

Formulas as predicates

Example

From binary predicates P and Q; and constant c we can build complex predicates like:

- $\alpha(w, x, y, z) = (P(x, w) \vee Q(y, z)) \wedge (w = y) \wedge (z = c)$
- $\beta(x, y, z) = (P(x, y) \vee Q(y, z)) \wedge (z = c)$
- $\gamma(x, y) = P(x, y) \vee Q(y, c)$
- $\delta(x) = \exists y P(x, y) \vee \exists y Q(y, c)$

NB

α, β and γ are different predicates: α represents a 4-ary predicate, whereas β represents a 3-ary predicate and γ represents a binary predicate.

Substitution

If t is a term, φ a formula, and $x \in FV(\varphi)$, then the **substitution of t for x in φ** (denoted $\varphi[t/x]$) is the formula obtained by replacing every free occurrence of x with t.

Alternatively (if the free variables are listed), substituting t for x in $\varphi(x)$ can be written as $\varphi(t)$.

Summary of topics

- Re-introduction to Predicate Logic
- Syntax of Predicate Logic
- Semantics of Predicate Logic
- Natural Deduction for Predicate Logic

Models

Predicate formulas are interpreted in **Models**.

Given a vocabulary V a model $\mathcal{M}$ defines:

- A (non-empty) domain $D = \mathrm{Dom}(\mathcal{M})$
- For every predicate symbol $P \in V$ with arity k: a k-ary relation $P^{\mathcal{M}}$ on D
- For every function symbol $f \in V$ with arity k: a function $f^{\mathcal{M}} : D^k \to D$
- For every constant symbol $c \in V$: an element, $c^{\mathcal{M}}$ of D

In this course (hopefully)

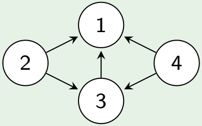
Formulas have **predicates**; Models have **relations**.

Models: examples

Example

For the vocabulary $V = \{\leq, +, 1\}$ the following are models:

- $\mathbb{N}$ with the standard definitions of $\leq$, $+$, and 1.
- $\{0, 1, 2, 3, 4\}$ with the standard definition of $\leq$ and 1, and $m + n$ defined as $m + n \pmod 5$.
- The directed graph $G = (V, E)$ shown below with $\leq = E$; and $v + w$ defined to be w.



Models: example (databases)

Example

For the vocabulary $DB = \{\text{Person}, \text{Employee}, \text{isString}, \text{isInteger}\}$, the following **database** is a model:

Person		
Name	**Surname**	**Address**
Arya	Stark	Winterfell
Jon	Snow	Winterfell
Cersei	Lannister	King's Landing

Employee	
ID	**Surname**
31415	Tyrell
27182	Lannister
16180	Targaryen

isString and isInteger are defined by what values are permitted in each of the columns (*sanitizing* the input).

Environments

Given a model $\mathcal{M}$, an **environment for** $\mathcal{M}$ (or **lookup table**) is a function from the set of variables to $\mathrm{Dom}(\mathcal{M})$.

Given an environment η, we denote by $\eta[x \mapsto c]$ the environment that agrees with η everywhere except possibly at x (where it has value c).

Environments

Given a model $\mathcal{M}$, an **environment for** $\mathcal{M}$ (or **lookup table**) is a function from the set of variables to $\mathrm{Dom}(\mathcal{M})$.

Given an environment η, we denote by $\eta[x \mapsto c]$ the environment that agrees with η everywhere except possibly at x (where it has value c).

Interpretations

An **interpretation** is a pair $(\mathcal{M}, \eta)$ where $\mathcal{M}$ is a model and η is an environment.

Interpretations

An **interpretation** is a pair $(\mathcal{M}, \eta)$ where $\mathcal{M}$ is a model and η is an environment.

An interpretation $(\mathcal{M}, \eta)$ maps terms to elements of $\mathrm{Dom}(\mathcal{M})$ recursively as follows:

- $[\![x]\!]^{\eta}_{\mathcal{M}} = \eta(x)$
- $[\![c]\!]^{\eta}_{\mathcal{M}} = c^{\mathcal{M}}$
- $[\![f(t_1, \ldots, t_k)]\!]^{\eta}_{\mathcal{M}} = f^{\mathcal{M}}([\![t_1]\!]^{\eta}_{\mathcal{M}}, \ldots, [\![t_k]\!]^{\eta}_{\mathcal{M}})$

Notation

We write $\mathcal{M}, \eta \models \varphi$ if $[\![\varphi]\!]^{\eta}_{\mathcal{M}} = \mathtt{true}$

Interpretations

An **interpretation** is a pair $(\mathcal{M}, \eta)$ where $\mathcal{M}$ is a model and η is an environment.

An interpretation $(\mathcal{M}, \eta)$ maps formulas to $\mathbb{B}$ recursively as follows:

- $[\![P(t_1, \ldots, t_k)]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $P^{\mathcal{M}}([\![t_1]\!]^{\eta}_{\mathcal{M}}, \ldots, [\![t_k]\!]^{\eta}_{\mathcal{M}})$ holds.
- $[\![t_1 = t_2]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $[\![t_1]\!]^{\eta}_{\mathcal{M}} = [\![t_2]\!]^{\eta}_{\mathcal{M}}$
- $[\![\forall x \varphi]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $[\![\varphi]\!]^{\eta[x \mapsto c]}_{\mathcal{M}} = \texttt{true}$ for all $c \in \mathrm{Dom}(\mathcal{M})$
- $[\![\exists x \varphi]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $[\![\varphi]\!]^{\eta[x \mapsto c]}_{\mathcal{M}} = \texttt{true}$ for some $c \in \mathrm{Dom}(\mathcal{M})$
- $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$ defined in the same way as Propositional Logic for all other formulas φ. For example $[\![\varphi \wedge \psi]\!]^{\eta}_{\mathcal{M}} = [\![\varphi]\!]^{\eta}_{\mathcal{M}} \&\& [\![\psi]\!]^{\eta}_{\mathcal{M}}$

Notation

We write $\mathcal{M}, \eta \models \varphi$ if $[\![\varphi]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$

Interpretations

An **interpretation** is a pair $(\mathcal{M}, \eta)$ where $\mathcal{M}$ is a model and η is an environment.

An interpretation $(\mathcal{M}, \eta)$ maps formulas to $\mathbb{B}$ recursively as follows:

- $[\![P(t_1, \ldots, t_k)]\!]^\eta_\mathcal{M} = \mathtt{true}$ if $P^\mathcal{M}([\![t_1]\!]^\eta_\mathcal{M}, \ldots, [\![t_k]\!]^\eta_\mathcal{M})$ holds.
- $[\![t_1 = t_2]\!]^\eta_\mathcal{M} = \mathtt{true}$ if $[\![t_1]\!]^\eta_\mathcal{M} = [\![t_2]\!]^\eta_\mathcal{M}$
- $[\![\forall x \varphi]\!]^\eta_\mathcal{M} = \mathtt{true}$ if $[\![\varphi]\!]^{\eta[x \mapsto c]}_\mathcal{M} = \mathtt{true}$ for all $c \in \mathrm{Dom}(\mathcal{M})$
- $[\![\exists x \varphi]\!]^\eta_\mathcal{M} = \mathtt{true}$ if $[\![\varphi]\!]^{\eta[x \mapsto c]}_\mathcal{M} = \mathtt{true}$ for some $c \in \mathrm{Dom}(\mathcal{M})$
- $[\![\varphi]\!]^\eta_\mathcal{M}$ defined in the same way as Propositional Logic for all other formulas φ. For example $[\![\varphi \wedge \psi]\!]^\eta_\mathcal{M} = [\![\varphi]\!]^\eta_\mathcal{M} \&\& [\![\psi]\!]^\eta_\mathcal{M}$

Notation

We write $\mathcal{M}, \eta \models \varphi$ if $[\![\varphi]\!]^\eta_\mathcal{M} = \mathtt{true}$

Interpretations

An **interpretation** is a pair $(\mathcal{M}, \eta)$ where $\mathcal{M}$ is a model and η is an environment.

An interpretation $(\mathcal{M}, \eta)$ maps formulas to $\mathbb{B}$ recursively as follows:

- $[\![P(t_1, \ldots, t_k)]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $P^{\mathcal{M}}([\![t_1]\!]^{\eta}_{\mathcal{M}}, \ldots, [\![t_k]\!]^{\eta}_{\mathcal{M}})$ holds.
- $[\![t_1 = t_2]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $[\![t_1]\!]^{\eta}_{\mathcal{M}} = [\![t_2]\!]^{\eta}_{\mathcal{M}}$
- $[\![\forall x \varphi]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $[\![\varphi]\!]^{\eta[x \mapsto c]}_{\mathcal{M}} = \texttt{true}$ for all $c \in \mathrm{Dom}(\mathcal{M})$
- $[\![\exists x \varphi]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $[\![\varphi]\!]^{\eta[x \mapsto c]}_{\mathcal{M}} = \texttt{true}$ for some $c \in \mathrm{Dom}(\mathcal{M})$
- $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$ defined in the same way as Propositional Logic for all other formulas φ. For example $[\![\varphi \wedge \psi]\!]^{\eta}_{\mathcal{M}} = [\![\varphi]\!]^{\eta}_{\mathcal{M}} \&\& [\![\psi]\!]^{\eta}_{\mathcal{M}}$

Notation

We write $\mathcal{M}, \eta \models \varphi$ if $[\![\varphi]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$

Interpretations

An **interpretation** is a pair $(\mathcal{M}, \eta)$ where $\mathcal{M}$ is a model and η is an environment.

An interpretation $(\mathcal{M}, \eta)$ maps formulas to $\mathbb{B}$ recursively as follows:

- $[\![P(t_1, \ldots, t_k)]\!]^\eta_\mathcal{M} = \texttt{true}$ if $P^\mathcal{M}([\![t_1]\!]^\eta_\mathcal{M}, \ldots, [\![t_k]\!]^\eta_\mathcal{M})$ holds.
- $[\![t_1 = t_2]\!]^\eta_\mathcal{M} = \texttt{true}$ if $[\![t_1]\!]^\eta_\mathcal{M} = [\![t_2]\!]^\eta_\mathcal{M}$
- $[\![\forall x \varphi]\!]^\eta_\mathcal{M} = \texttt{true}$ if $[\![\varphi]\!]^{\eta[x \mapsto c]}_\mathcal{M} = \texttt{true}$ for all $c \in \mathrm{Dom}(\mathcal{M})$
- $[\![\exists x \varphi]\!]^\eta_\mathcal{M} = \texttt{true}$ if $[\![\varphi]\!]^{\eta[x \mapsto c]}_\mathcal{M} = \texttt{true}$ for some $c \in \mathrm{Dom}(\mathcal{M})$
- $[\![\varphi]\!]^\eta_\mathcal{M}$ defined in the same way as Propositional Logic for all other formulas φ. For example $[\![\varphi \wedge \psi]\!]^\eta_\mathcal{M} = [\![\varphi]\!]^\eta_\mathcal{M} \&\& [\![\psi]\!]^\eta_\mathcal{M}$

Notation

We write $\mathcal{M}, \eta \models \varphi$ if $[\![\varphi]\!]^\eta_\mathcal{M} = \texttt{true}$

Interpretations

An **interpretation** is a pair $(\mathcal{M}, \eta)$ where $\mathcal{M}$ is a model and η is an environment.

An interpretation $(\mathcal{M}, \eta)$ maps formulas to $\mathbb{B}$ recursively as follows:

- $[\![P(t_1, \ldots, t_k)]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $P^{\mathcal{M}}([\![t_1]\!]^{\eta}_{\mathcal{M}}, \ldots, [\![t_k]\!]^{\eta}_{\mathcal{M}})$ holds.
- $[\![t_1 = t_2]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $[\![t_1]\!]^{\eta}_{\mathcal{M}} = [\![t_2]\!]^{\eta}_{\mathcal{M}}$
- $[\![\forall x \varphi]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $[\![\varphi]\!]^{\eta[x \mapsto c]}_{\mathcal{M}} = \texttt{true}$ for all $c \in \mathrm{Dom}(\mathcal{M})$
- $[\![\exists x \varphi]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $[\![\varphi]\!]^{\eta[x \mapsto c]}_{\mathcal{M}} = \texttt{true}$ for some $c \in \mathrm{Dom}(\mathcal{M})$
- $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$ defined in the same way as Propositional Logic for all other formulas φ. For example $[\![\varphi \wedge \psi]\!]^{\eta}_{\mathcal{M}} = [\![\varphi]\!]^{\eta}_{\mathcal{M}} \&\& [\![\psi]\!]^{\eta}_{\mathcal{M}}$

Notation

We write $\mathcal{M}, \eta \models \varphi$ if $[\![\varphi]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$

Interpretations

An **interpretation** is a pair $(\mathcal{M}, \eta)$ where $\mathcal{M}$ is a model and η is an environment.

An interpretation $(\mathcal{M}, \eta)$ maps formulas to $\mathbb{B}$ recursively as follows:

- $[\![P(t_1, \ldots, t_k)]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $P^{\mathcal{M}}([\![t_1]\!]^{\eta}_{\mathcal{M}}, \ldots, [\![t_k]\!]^{\eta}_{\mathcal{M}})$ holds.
- $[\![t_1 = t_2]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $[\![t_1]\!]^{\eta}_{\mathcal{M}} = [\![t_2]\!]^{\eta}_{\mathcal{M}}$
- $[\![\forall x \varphi]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $[\![\varphi]\!]^{\eta[x \mapsto c]}_{\mathcal{M}} = \texttt{true}$ for all $c \in \mathrm{Dom}(\mathcal{M})$
- $[\![\exists x \varphi]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$ if $[\![\varphi]\!]^{\eta[x \mapsto c]}_{\mathcal{M}} = \texttt{true}$ for some $c \in \mathrm{Dom}(\mathcal{M})$
- $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$ defined in the same way as Propositional Logic for all other formulas φ. For example $[\![\varphi \wedge \psi]\!]^{\eta}_{\mathcal{M}} = [\![\varphi]\!]^{\eta}_{\mathcal{M}} \&\& [\![\psi]\!]^{\eta}_{\mathcal{M}}$

Notation

We write $\mathcal{M}, \eta \models \varphi$ if $[\![\varphi]\!]^{\eta}_{\mathcal{M}} = \texttt{true}$

Interpretations: examples

Example

$$\forall x \forall y ((y = x + 1) \rightarrow (x \leq y))$$

- $\mathbb{N}$ with the standard definitions of $\leq$, $+$, and 1: true
- $\{0, 1, 2, 3, 4\}$ with the standard definition of $\leq$ and 1, and $m + n$ defined as $m + n \pmod 5$:
- The directed graph $G = (V, E)$ shown below with $\leq = E$; and $v + w$ defined to be w.

Interpretations: examples

Example

$$\forall x \forall y ((y = x + 1) \rightarrow (x \leq y))$$

- $\mathbb{N}$ with the standard definitions of $\leq$, $+$, and 1: `true`
- $\{0, 1, 2, 3, 4\}$ with the standard definition of $\leq$ and 1, and $m + n$ defined as $m + n \pmod 5$:
- The directed graph $G = (V, E)$ shown below with $\leq = E$; and $v + w$ defined to be w.

Interpretations: examples

Example

$$\forall x \forall y((y = x + 1) \rightarrow (x \leq y))$$

- $\mathbb{N}$ with the standard definitions of $\leq$, $+$, and 1: `true`
- $\{0, 1, 2, 3, 4\}$ with the standard definition of $\leq$ and 1, and $m + n$ defined as $m + n \pmod 5$: `false`
- The directed graph $G = (V, E)$ shown below with $\leq = E$; and $v + w$ defined to be w.

Interpretations: examples

Example

$$\forall x \forall y((y = x + 1) \to (x \leq y))$$

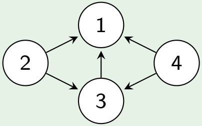
- $\mathbb{N}$ with the standard definitions of $\leq$, $+$, and 1: `true`
- $\{0, 1, 2, 3, 4\}$ with the standard definition of $\leq$ and 1, and $m + n$ defined as $m + n \pmod 5$: `false`
- The directed graph $G = (V, E)$ shown below with $\leq = E$; and $v + w$ defined to be w.

Interpretations: examples

Example

$$\forall x \forall y((y = x + 1) \rightarrow (x \leq y))$$

- $\mathbb{N}$ with the standard definitions of $\leq$, $+$, and 1: `true`
- $\{0, 1, 2, 3, 4\}$ with the standard definition of $\leq$ and 1, and $m + n$ defined as $m + n \pmod 5$: `false`
- The directed graph $G = (V, E)$ shown below with $\leq = E$; and $v + w$ defined to be w.



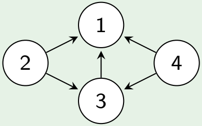
`true`

Interpretations: examples

Example

$$\forall x \forall y((y = x + 1) \to (x \leq y))$$

- $\mathbb{N}$ with the standard definitions of $\leq$, $+$, and 1: `true`
- $\{0, 1, 2, 3, 4\}$ with the standard definition of $\leq$ and 1, and $m + n$ defined as $m + n \pmod 5$: `false`
- The directed graph $G = (V, E)$ shown below with $\leq = E$; and $v + w$ defined to be w.



`true`

Why separate the environment from the model?

In the definition of $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$, η is only used to define values for the free variables. In particular, if φ is a sentence then $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$ is independent of η.

Define $[\![\cdot]\!]_{\mathcal{M}}$ by "delaying" the assigning of values to free variables, and propagating them out. That is, define:

$$[\![\varphi(x_1, x_2, \ldots, x_n)]\!]_{\mathcal{M}} = [\![\varphi]\!]_{\mathcal{M}}(x_1, x_2, \ldots, x_n)$$

where $[\![\varphi]\!]_{\mathcal{M}} : \mathrm{Dom}(\mathcal{M})^n \to \mathbb{B}$; that is, $[\![\varphi]\!]_{\mathcal{M}}$ is an n-ary relation on $\mathrm{Dom}(\mathcal{M})$.

This matches the perspective of formulas as complex predicates:

$$\varphi(x_1, x_2, \ldots, x_n) \text{ an } n\text{-ary predicate}$$
$$\Downarrow$$
$$[\![\varphi]\!]_{\mathcal{M}} \text{ an } n\text{-ary relation on } \mathrm{Dom}(\mathcal{M})$$

Why separate the environment from the model?

In the definition of $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$, η is only used to define values for the free variables. In particular, if φ is a sentence then $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$ is independent of η.

Define $[\![\cdot]\!]_{\mathcal{M}}$ by "delaying" the assigning of values to free variables, and propagating them out. That is, define:

$$[\![\varphi(x_1, x_2, \ldots, x_n)]\!]_{\mathcal{M}} = [\![\varphi]\!]_{\mathcal{M}}(x_1, x_2, \ldots, x_n)$$

where $[\![\varphi]\!]_{\mathcal{M}} : \mathrm{Dom}(\mathcal{M})^n \to \mathbb{B}$; that is, $[\![\varphi]\!]_{\mathcal{M}}$ is an n-ary relation on $\mathrm{Dom}(\mathcal{M})$.

This matches the perspective of formulas as complex predicates:

$$\varphi(x_1, x_2, \ldots, x_n) \text{ an } n\text{-ary predicate}$$
$$\Downarrow$$
$$[\![\varphi]\!]_{\mathcal{M}} \text{ an } n\text{-ary relation on } \mathrm{Dom}(\mathcal{M})$$

Why separate the environment from the model?

In the definition of $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$, η is only used to define values for the free variables. In particular, if φ is a sentence then $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$ is independent of η.

Define $[\![\cdot]\!]_{\mathcal{M}}$ by "delaying" the assigning of values to free variables, and propagating them out. That is, define:

$$[\![\varphi(x_1, x_2, \ldots, x_n)]\!]_{\mathcal{M}} = [\![\varphi]\!]_{\mathcal{M}}(x_1, x_2, \ldots, x_n)$$

where $[\![\varphi]\!]_{\mathcal{M}} : \mathrm{Dom}(\mathcal{M})^n \to \mathbb{B}$; that is, $[\![\varphi]\!]_{\mathcal{M}}$ is an n-ary relation on $\mathrm{Dom}(\mathcal{M})$.

This matches the perspective of formulas as complex predicates:

$$\varphi(x_1, x_2, \ldots, x_n) \text{ an } n\text{-ary predicate}$$
$$\Downarrow$$
$$[\![\varphi]\!]_{\mathcal{M}} \text{ an } n\text{-ary relation on } \mathrm{Dom}(\mathcal{M})$$

Why separate the environment from the model?

In the definition of $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$, η is only used to define values for the free variables. In particular, if φ is a sentence then $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$ is independent of η.

Define $[\![\cdot]\!]_{\mathcal{M}}$ by "delaying" the assigning of values to free variables, and propagating them out. That is, define:

$$[\![\varphi(x_1, x_2, \ldots, x_n)]\!]_{\mathcal{M}} = [\![\varphi]\!]_{\mathcal{M}}(x_1, x_2, \ldots, x_n)$$

where $[\![\varphi]\!]_{\mathcal{M}} : \mathsf{Dom}(\mathcal{M})^n \to \mathbb{B}$; that is, $[\![\varphi]\!]_{\mathcal{M}}$ is an n-ary relation on $\mathsf{Dom}(\mathcal{M})$.

This matches the perspective of formulas as complex predicates:

$$\varphi(x_1, x_2, \ldots, x_n) \text{ an } n\text{-ary predicate}$$
$$\Downarrow$$
$$[\![\varphi]\!]_{\mathcal{M}} \text{ an } n\text{-ary relation on } \mathsf{Dom}(\mathcal{M})$$

Why separate the environment from the model?

In the definition of $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$, η is only used to define values for the free variables. In particular, if φ is a sentence then $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$ is independent of η.

Define $[\![\cdot]\!]_{\mathcal{M}}$ by "delaying" the assigning of values to free variables, and propagating them out. That is, define:

$$[\![\varphi(x_1, x_2, \ldots, x_n)]\!]_{\mathcal{M}} = [\![\varphi]\!]_{\mathcal{M}}(x_1, x_2, \ldots, x_n)$$

where $[\![\varphi]\!]_{\mathcal{M}} : \mathrm{Dom}(\mathcal{M})^n \to \mathbb{B}$; that is, $[\![\varphi]\!]_{\mathcal{M}}$ is an n-ary relation on $\mathrm{Dom}(\mathcal{M})$.

This matches the perspective of formulas as complex predicates:

$$\varphi(x_1, x_2, \ldots, x_n) \text{ an } n\text{-ary predicate}$$

$$\Downarrow$$

$$[\![\varphi]\!]_{\mathcal{M}} \text{ an } n\text{-ary relation on } \mathrm{Dom}(\mathcal{M})$$

Why separate the environment from the model?

In the definition of $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$, η is only used to define values for the free variables. In particular, if φ is a sentence then $[\![\varphi]\!]^{\eta}_{\mathcal{M}}$ is independent of η.

Define $[\![\cdot]\!]_{\mathcal{M}}$ by "delaying" the assigning of values to free variables, and propagating them out. That is, define:

$$[\![\varphi(x_1, x_2, \ldots, x_n)]\!]_{\mathcal{M}} = [\![\varphi]\!]_{\mathcal{M}}(x_1, x_2, \ldots, x_n)$$

where $[\![\varphi]\!]_{\mathcal{M}} : \mathrm{Dom}(\mathcal{M})^n \to \mathbb{B}$; that is, $[\![\varphi]\!]_{\mathcal{M}}$ is an n-ary relation on $\mathrm{Dom}(\mathcal{M})$.

This matches the perspective of formulas as complex predicates:

$$\begin{gathered}\varphi(x_1, x_2, \ldots, x_n) \text{ an } n\text{-ary predicate} \\ \Downarrow \\ [\![\varphi]\!]_{\mathcal{M}} \text{ an } n\text{-ary relation on } \mathrm{Dom}(\mathcal{M})\end{gathered}$$

Interpretations: example (databases)

Example

- Vocabulary: database schema
- Formulas: queries (φ)
- Models: databases ($\mathcal{D}$)
- Interpretation: $[\![\varphi]\!]_{\mathcal{D}}$ is a relation on $\mathrm{Dom}(\mathcal{D})$, i.e. a (derived) table in $\mathcal{D}$
- Environment:
- $[\![\varphi]\!]^{\eta}_{\mathcal{D}}$: Success/fail outcome of looking up a specific entry in a query result on $\mathcal{D}$.

Interpretations: example (databases)

Example

- Vocabulary: database schema
- Formulas: queries (φ)
- Models: databases ($\mathcal{D}$)
- Interpretation: $[\![\varphi]\!]_{\mathcal{D}}$ is a relation on $\mathrm{Dom}(\mathcal{D})$, i.e. a (derived) table in $\mathcal{D}$
- Environment:
- $[\![\varphi]\!]_{\mathcal{D}}^{\eta}$: Success/fail outcome of looking up a specific entry in a query result on $\mathcal{D}$.

Interpretations: example (databases)

Example

- Vocabulary: database schema
- Formulas: queries (φ)
- Models: databases ($\mathcal{D}$)
- Interpretation: $[\![\varphi]\!]_{\mathcal{D}}$ is a relation on $\mathrm{Dom}(\mathcal{D})$, i.e. a (derived) table in $\mathcal{D}$
- Environment: "looks up" an entry in a (derived) table and returns whether the lookup was successful
- $[\![\varphi]\!]^{\eta}_{\mathcal{D}}$: Success/fail outcome of looking up a specific entry in a query result on $\mathcal{D}$.

Interpretations: example (databases)

Example

- Vocabulary: database schema
- Formulas: queries (φ)
- Models: databases ($\mathcal{D}$)
- Interpretation: $[\![\varphi]\!]_{\mathcal{D}}$ is a relation on $\mathrm{Dom}(\mathcal{D})$, i.e. a (derived) table in $\mathcal{D}$
- Environment: “looks up” an entry in a (derived) table and returns whether the lookup was successful
- $[\![\varphi]\!]^{\eta}_{\mathcal{D}}$: Success/fail outcome of looking up a specific entry in a query result on $\mathcal{D}$.

Interpretations: example (databases)

Example

- Vocabulary: database schema
- Formulas: queries (φ)
- Models: databases ($\mathcal{D}$)
- Interpretation: $[\![\varphi]\!]_{\mathcal{D}}$ is a relation on $\mathrm{Dom}(\mathcal{D})$, i.e. a (derived) table in $\mathcal{D}$
- Environment: "looks up" an entry in a (derived) table and returns whether the lookup was successful
- $[\![\varphi]\!]_{\mathcal{D}}^{\eta}$: Success/fail outcome of looking up a specific entry in a query result on $\mathcal{D}$.

Satisfiability, truth, validity

A formula φ of predicate logic is:

- **satisfiable** if there is some model $\mathcal{M}$ and some environment η such that $\mathcal{M}, \eta \models \varphi$. That is, there is some interpretation $(\mathcal{M}, \eta)$ that **satisfies** φ.
- **true in a model** $\mathcal{M}$ if for all environments η we have $\mathcal{M}, \eta \models \varphi$.
- a **logical validity** if it is true in all models.

NB

For sentences the first two definitions coincide.

Example

The sentence $\forall x \forall y((y = x + 1) \to (x \leq y))$ is satisfiable but not a logical validity.

Satisfiability, truth, validity

A formula φ of predicate logic is:

- **satisfiable** if there is some model $\mathcal{M}$ and some environment η such that $\mathcal{M}, \eta \models \varphi$. That is, there is some interpretation $(\mathcal{M}, \eta)$ that **satisfies** φ.
- **true in a model** $\mathcal{M}$ if for all environments η we have $\mathcal{M}, \eta \models \varphi$.
- a **logical validity** if it is true in all models.

NB

For sentences the first two definitions coincide.

Example

The sentence $\forall x \forall y((y = x + 1) \to (x \leq y))$ is satisfiable but not a logical validity.

Entailment, Logical equivalence

- A theory T **entails** a formula φ, $T \models \varphi$, if φ is satisfied by any interpretation that satisfies all formulas in T.
- φ is **logically equivalent** to ψ, $\varphi \equiv \psi$, if $[\![\varphi]\!]^{\eta}_{\mathcal{M}} = [\![\psi]\!]^{\eta}_{\mathcal{M}}$ for all interpretations $(\mathcal{M}, \eta)$.

Theorem

- $\varphi_1, \ldots, \varphi_n \models \psi$ *if, and only if,* $(\varphi_1 \wedge \cdots \wedge \varphi_n) \rightarrow \psi$ *is a logical validity.*
- $\varphi \equiv \psi$ *if, and only if,* $\varphi \leftrightarrow \psi$ *is a logical validity.*

Entailment, Logical equivalence

- A theory T **entails** a formula φ, $T \models \varphi$, if φ is satisfied by any interpretation that satisfies all formulas in T.
- φ is **logically equivalent** to ψ, $\varphi \equiv \psi$, if $[\![\varphi]\!]^{\eta}_{\mathcal{M}} = [\![\psi]\!]^{\eta}_{\mathcal{M}}$ for all interpretations $(\mathcal{M}, \eta)$.

Theorem

- $\varphi_1, \ldots, \varphi_n \models \psi$ *if, and only if,* $(\varphi_1 \wedge \cdots \wedge \varphi_n) \to \psi$ *is a logical validity.*
- $\varphi \equiv \psi$ *if, and only if,* $\varphi \leftrightarrow \psi$ *is a logical validity.*

Summary of topics

- Re-introduction to Predicate Logic
- Syntax of Predicate Logic
- Semantics of Predicate Logic
- Natural Deduction for Predicate Logic

Motivation

Demonstrating satisfiability (and invalidity) is easy: just provide an interpretation which does (not) satisfy the formula. Note: *finding* such an interpretation is a different question.

How can you show a formula/entailment is valid?

Answer: Find a proof in a proof system that is sound.

Motivation

Demonstrating satisfiability (and invalidity) is easy: just provide an interpretation which does (not) satisfy the formula. Note: *finding* such an interpretation is a different question.

How can you show a formula/entailment is valid?

Answer: Find a proof in a proof system that is sound.

Natural deduction for Predicate Logic

Inference rules for Propositional Logic $+$ seven rules for quantifiers and equality

Operator	**Introduction**	**Elimination**
$\forall$	$\forall$-I	$\forall$-E
$\exists$	$\exists$-I	$\exists$-E
$=$	$=$-I	$=$-E1 $=$-E2

Arbitrary variables

Formulas of Predicate Logic involve variables. Unsurprisingly, the new inference rules involve manipulating variables.

A variable is **arbitrary** if it does not occur (as a free variable) in any undischarged assumption.

Intuitively: an arbitrary variable can be assigned any element of the domain and the formula will still hold.

$\forall$ Introduction and Elimination

$\forall$-elimination:

$$\frac{\forall x A(x)}{A(c)}\ (\forall\text{-E})$$

$\forall$-introduction:

$$\frac{A(c) \qquad \begin{array}{c}(c \text{ is arbitrary})\\ (x \text{ not free in } A(c))\\ (c \text{ not free in } A(x))\end{array}}{\forall x A(x)}\ (\forall\text{-I})$$

$\forall$ Introduction and Elimination

$\forall$-elimination:

$$\frac{\forall x A(x)}{A(c)} \ (\forall\text{-E})$$

$\forall$-introduction:

$$\frac{A(c) \qquad \begin{array}{c} (c \text{ is arbitrary}) \\ (x \text{ not free in } A(c)) \\ (c \text{ not free in } A(x)) \end{array}}{\forall x A(x)} \ (\forall\text{-I})$$

Proof example

Prove: $\forall x \forall y\, P(x,y) \vdash \forall y \forall x\, P(x,y)$

Line	Premises	Formula	Rule	References
1		$\forall x \forall y\, P(x,y)$	Premise	
2	1	$\forall y\, P(a,y)$	$\forall$-E	1
3	1	$P(a,b)$	$\forall$-E	2
4	1	$\forall x\, P(x,b)$	$\forall$-I	3
5	1	$\forall y \forall x\, P(x,y)$	$\forall$-I	4

Proof example

Prove: $\forall x \forall y\, P(x,y) \vdash \forall y \forall x\, P(x,y)$

Line	Premises	Formula	Rule	References
1		$\forall x \forall y\, P(x,y)$	Premise	
2	1	$\forall y\, P(a,y)$	$\forall$-E	1
3	1	$P(a,b)$	$\forall$-E	2
4	1	$\forall x\, P(x,b)$	$\forall$-I	3
5	1	$\forall y \forall x\, P(x,y)$	$\forall$-I	4

Proof example

Prove: $\forall x \forall y\, P(x,y) \vdash \forall y \forall x\, P(x,y)$

Line	Premises	Formula	Rule	References
1		$\forall x \forall y\, P(x,y)$	Premise	
2	1	$\forall y\, P(a,y)$	$\forall$-E	1
3	1	$P(a,b)$	$\forall$-E	2
4	1	$\forall x\, P(x,b)$	$\forall$-I	3
5	1	$\forall y \forall x\, P(x,y)$	$\forall$-I	4

Proof example

Prove: $\forall x \forall y\, P(x,y) \vdash \forall y \forall x\, P(x,y)$

Line	Premises	Formula	Rule	References
1		$\forall x \forall y\, P(x,y)$	Premise	
2	1	$\forall y\, P(a,y)$	$\forall$-E	1
3	1	$P(a,b)$	$\forall$-E	2
4	1	$\forall x\, P(x,b)$	$\forall$-I	3
5	1	$\forall y \forall x\, P(x,y)$	$\forall$-I	4

Proof example

Prove: $\forall x \forall y\ P(x,y) \vdash \forall y \forall x\ P(x,y)$

Line	Premises	Formula	Rule	References
1		$\forall x \forall y\ P(x,y)$	Premise	
2	1	$\forall y\ P(a,y)$	$\forall$-E	1
3	1	$P(a,b)$	$\forall$-E	2
4	1	$\forall x\ P(x,b)$	$\forall$-I	3
5	1	$\forall y \forall x\ P(x,y)$	$\forall$-I	4

Proof example

Prove: $\forall x \forall y\, P(x, y) \vdash \forall y \forall x\, P(x, y)$

Line	**Premises**	**Formula**	**Rule**	**References**
1		$\forall x \forall y\, P(x, y)$	Premise	
2	1	$\forall y\, P(a, y)$	$\forall$-E	1
3	1	$P(a, b)$	$\forall$-E	2
4	1	$\forall x\, P(x, b)$	$\forall$-I	3
5	1	$\forall y \forall x\, P(x, y)$	$\forall$-I	4

$\exists$ Introduction and Elimination

$\exists$-introduction:

$$\frac{A(c) \qquad (x \text{ not free in } A)}{\exists x A(x)}\ (\exists\text{-I})$$

$\exists$-elimination:

$$\frac{\exists x A(x) \qquad \begin{matrix}[A(c)]\\ \vdots \\ B\end{matrix} \qquad \begin{matrix}(x \text{ not free in } A(c))\\ (c \text{ is arbitrary})\\ (c \text{ not free in } B)\end{matrix}}{B}\ (\exists\text{-E})$$

$\exists$ Introduction and Elimination

$\exists$-introduction:

$$\frac{A(c) \qquad (x \text{ not free in } A)}{\exists x A(x)}\ (\exists\text{-I})$$

$\exists$-elimination:

$$\frac{\exists x A(x) \qquad \begin{array}{c} [A(c)] \\ \vdots \\ B \end{array} \qquad \begin{array}{c} (x \text{ not free in } A(c)) \\ (c \text{ is arbitrary}) \\ (c \text{ not free in } B) \end{array}}{B}\ (\exists\text{-E})$$

Proof example (Fitch)

Prove: $\exists x \exists y\, P(x,y) \vdash \exists y \exists x\, P(x,y)$

1. $\exists x \exists y\, P(x, y)$	
2. $\exists y\, P(a, y)$	
3. $P(a, b)$	
4. $\exists x\, P(x, b)$	$\exists$-I: 3
5. $\exists y \exists x\, P(x, y)$	$\exists$-I: 4
6. $\exists y \exists x\, P(x, y)$	$\exists$-E: 2, 3–5
7. $\exists y \exists x\, P(x, y)$	$\exists$-E: 1, 2–6

Proof example (Fitch)

Prove: $\exists x \exists y\, P(x, y) \vdash \exists y \exists x\, P(x, y)$

1. ∃x∃y P(x, y)
   2. ∃y P(a, y)
      3. P(a, b)
      4. ∃x P(x, b) ∃-I: 3
      5. ∃y∃x P(x, y) ∃-I: 4
   6. ∃y∃x P(x, y) ∃-E: 2, 3–5
7. ∃y∃x P(x, y) ∃-E: 1, 2–6

Proof example (Fitch)

Prove: $\exists x \exists y\, P(x,y) \vdash \exists y \exists x\, P(x,y)$

1. $\exists x \exists y\, P(x,y)$
2. $\exists y\, P(a,y)$
3. $P(a,b)$
4. $\exists x\, P(x,b)$ — $\exists$-I: 3
5. $\exists y \exists x\, P(x,y)$ — $\exists$-I: 4
6. $\exists y \exists x\, P(x,y)$ — $\exists$-E: 2, 3–5
7. $\exists y \exists x\, P(x,y)$ — $\exists$-E: 1, 2–6

Proof example (Fitch)

Prove: $\exists x \exists y\, P(x,y) \vdash \exists y \exists x\, P(x,y)$

1. $\exists x \exists y\, P(x,y)$	
2. $\exists y\, P(a,y)$	
3. $P(a,b)$	
4. $\exists x\, P(x,b)$	$\exists$-I: 3
5. $\exists y \exists x\, P(x,y)$	$\exists$-I: 4
6. $\exists y \exists x\, P(x,y)$	$\exists$-E: 2, 3–5
7. $\exists y \exists x\, P(x,y)$	$\exists$-E: 1, 2–6

Proof example (Fitch)

Prove: $\exists x \exists y\, P(x,y) \vdash \exists y \exists x\, P(x,y)$

1. $\exists x \exists y\, P(x,y)$
2. $\exists y\, P(a,y)$
3. $P(a,b)$
4. $\exists x\, P(x,b)$ — $\exists$-I: 3
5. $\exists y \exists x\, P(x,y)$ — $\exists$-I: 4
6. $\exists y \exists x\, P(x,y)$ — $\exists$-E: 2, 3–5
7. $\exists y \exists x\, P(x,y)$ — $\exists$-E: 1, 2–6

Proof example (Fitch)

Prove: $\exists x \exists y\, P(x, y) \vdash \exists y \exists x\, P(x, y)$

1. $\exists x \exists y\, P(x, y)$	
2. $\exists y\, P(a, y)$	
3. $P(a, b)$	
4. $\exists x\, P(x, b)$	$\exists$-I: 3
5. $\exists y \exists x\, P(x, y)$	$\exists$-I: 4
6. $\exists y \exists x\, P(x, y)$	$\exists$-E: 2, 3–5
7. $\exists y \exists x\, P(x, y)$	$\exists$-E: 1, 2–6

Proof example (Fitch)

Prove: $\exists x \exists y\, P(x,y) \vdash \exists y \exists x\, P(x,y)$

1.	$\exists x \exists y\, P(x,y)$	
2.	$\quad \exists y\, P(a,y)$	
3.	$\quad\quad P(a,b)$	
4.	$\quad\quad \exists x\, P(x,b)$	$\exists$-I: 3
5.	$\quad\quad \exists y \exists x\, P(x,y)$	$\exists$-I: 4
6.	$\quad \exists y \exists x\, P(x,y)$	$\exists$-E: 2, 3–5
7.	$\exists y \exists x\, P(x,y)$	$\exists$-E: 1, 2–6

$=$ Introduction and Elimination

$=$-introduction: $$\frac{}{a = a}\ (=\text{-I})$$

$=$-elimination (1): $$\frac{a = b \qquad A(a)}{A(b)}\ (=\text{-E1})$$

$=$-elimination (2): $$\frac{a = b \qquad A(b)}{A(a)}\ (=\text{-E1})$$

Proof example (Fitch)

Prove: $\vdash \forall x \forall y\, (x = y) \rightarrow (y = x)$

1. a = b	
2. a = a	=-I
3. b = a	=-E1: 1,2
4. (a = b) → (b = a)	→-I: 1–3
5. ∀y (a = y) → (y = a)	∀-I: 4
6. ∀x∀y (x = y) → (y = x)	∀-I: 5

Proof example (Fitch)

Prove: $\vdash \forall x \forall y\, (x = y) \rightarrow (y = x)$

1. a = b	
2. a = a	=-I
3. b = a	=-E1: 1,2
4. (a = b) → (b = a)	→-I: 1–3
5. ∀y (a = y) → (y = a)	∀-I: 4
6. ∀x∀y (x = y) → (y = x)	∀-I: 5

Proof example (Fitch)

Prove: $\vdash \forall x \forall y\,(x = y) \rightarrow (y = x)$

1. $a = b$	
2. $a = a$	$=$-I
3. $b = a$	$=$-E1: 1,2
4. $(a = b) \rightarrow (b = a)$	$\rightarrow$-I: 1–3
5. $\forall y\,(a = y) \rightarrow (y = a)$	$\forall$-I: 4
6. $\forall x \forall y\,(x = y) \rightarrow (y = x)$	$\forall$-I: 5

Proof example (Fitch)

Prove: $\vdash \forall x \forall y\, (x = y) \rightarrow (y = x)$

1. $a = b$	
2. $a = a$	$=$-I
3. $b = a$	$=$-E1: 1,2
4. $(a = b) \rightarrow (b = a)$	$\rightarrow$-I: 1–3
5. $\forall y\, (a = y) \rightarrow (y = a)$	$\forall$-I: 4
6. $\forall x \forall y\, (x = y) \rightarrow (y = x)$	$\forall$-I: 5

Proof example (Fitch)

Prove: $\vdash \forall x \forall y\, (x = y) \rightarrow (y = x)$

1. a = b	
2. a = a	=-I
3. b = a	=-E1: 1,2
4. (a = b) → (b = a)	→-I: 1–3
5. ∀y (a = y) → (y = a)	∀-I: 4
6. ∀x∀y (x = y) → (y = x)	∀-I: 5

Proof example (Fitch)

Prove: $\vdash \forall x \forall y\,(x = y) \rightarrow (y = x)$

1. $a = b$	
2. $a = a$	$=$-I
3. $b = a$	$=$-E1: 1,2
4. $(a = b) \rightarrow (b = a)$	$\rightarrow$-I: 1–3
5. $\forall y\,(a = y) \rightarrow (y = a)$	$\forall$-I: 4
6. $\forall x \forall y\,(x = y) \rightarrow (y = x)$	$\forall$-I: 5

Proof example (Fitch)

Prove: $\vdash \forall x \forall y\, (x = y) \rightarrow (y = x)$

1. $a = b$	
2. $a = a$	$=$-I
3. $b = a$	$=$-E1: 1,2
4. $(a = b) \rightarrow (b = a)$	$\rightarrow$-I: 1–3
5. $\forall y\, (a = y) \rightarrow (y = a)$	$\forall$-I: 4
6. $\forall x \forall y\, (x = y) \rightarrow (y = x)$	$\forall$-I: 5

Soundness and completeness

Theorem

Natural deduction is sound and complete for Predicate Logic:

$$T \vdash \varphi \quad \textit{if, and only if,} \quad T \models \varphi$$

- Use proofs to show validity
- Use countermodels to show unprovability

Soundness and completeness

Theorem

Natural deduction is sound and complete for Predicate Logic:

$$T \vdash \varphi \quad \textit{if, and only if,} \quad T \models \varphi$$

- Use proofs to show validity
- Use countermodels to show unprovability

Soundness and completeness

Theorem

Natural deduction is sound and complete for Predicate Logic:

$$T \vdash \varphi \quad \textit{if, and only if,} \quad T \models \varphi$$

- Use proofs to show validity
- Use countermodels to show unprovability