The Importance of Being Formal

Martin Henz

February 19, 2014
1 Syntax of Predicate Logic
   - Need for Richer Language
   - Predicates
   - Variables
   - Functions

2 Predicate Logic as a Formal Language

3 Semantics of Predicate Logic

4 Proof Theory

5 Equivalences
Propositional logic can easily handle simple declarative statements such as:

**Example**

Student Peter Lim enrolled in UIT2206.
More Declarative Sentences

- Propositional logic can easily handle simple declarative statements such as:

**Example**

Student Peter Lim enrolled in UIT2206.

- Propositional logic can also handle combinations of such statements such as:

**Example**

Student Peter Lim enrolled in Tutorial 1, *and* student Julie Bradshaw is enrolled in Tutorial 2.
Propositional logic can easily handle simple declarative statements such as:

Example
Student Peter Lim enrolled in UIT2206.

Propositional logic can also handle combinations of such statements such as:

Example
Student Peter Lim enrolled in Tutorial 1, and student Julie Bradshaw is enrolled in Tutorial 2.

But: How about statements with “there exists...” or “every...” or “among...”?
What is needed?

Example

*Every* student is younger than *some* instructor.
What is needed?

Example

*Every* student is younger than *some* instructor.

What is this statement about?
Example

*Every* student is younger than *some* instructor.

What is this statement about?

- Being a student
- Being an instructor
- Being younger than somebody else
What is needed?

**Example**

*Every* student is younger than *some* instructor.

What is this statement about?

- Being a student
- Being an instructor
- Being younger than somebody else

These are *properties* of elements of a *set* of objects.
What is needed?

Example

*Every* student is younger than *some* instructor.

What is this statement about?

- Being a student
- Being an instructor
- Being younger than somebody else

These are *properties* of elements of a *set* of objects.

We express them in predicate logic using *predicates*.
Predicates

Example

*Every* student is younger than *some* instructor.
Predicates

Example

*Every* student is younger than *some* instructor.

- $S(andy)$ could denote that Andy is a student.
- $l(paul)$ could denote that Paul is an instructor.
- $Y(andy, paul)$ could denote that Andy is younger than Paul.
The Need for Variables

Example

*Every* student is younger than *some* instructor.
The Need for Variables

Example

*Every* student is younger than *some* instructor.

We use the predicate $S$ to denote student-hood.
The Need for Variables

Example

*Every* student is younger than *some* instructor.

We use the predicate $S$ to denote student-hood. How do we express “*every student*”? 
Example

Every student is younger than some instructor.

We use the predicate $S$ to denote student-hood. How do we express “every student”?

We need variables that can stand for constant values, and a quantifier symbol that denotes “every”.
The Need for Variables

Example

*Every* student is younger than *some* instructor.

Using variables and quantifiers, we can write:

\[ \forall x (S(x) \rightarrow \exists y (I(y) \land Y(x, y))) \]

Literally: For every \( x \), if \( x \) is a student, then there is some \( y \) such that \( y \) is an instructor and \( x \) is younger than \( y \).
The Need for Variables

Example

*Every* student is younger than *some* instructor.

Using variables and quantifiers, we can write:

$$\forall x (S(x) \rightarrow (\exists y (I(y) \land Y(x, y))))$$

Literally: For every $x$, if $x$ is a student, then there is some $y$ such that $y$ is an instructor and $x$ is younger than $y$. 
Another Example

English

Not all birds can fly.

Predicates

B(x): x is a bird

F(x): x can fly

The sentence in predicate logic is:

\[ \neg (\forall x (B(x) \rightarrow F(x))) \]
Another Example

English

Not all birds can fly.

Predicates

\[ B(x): x \text{ is a bird} \]
\[ F(x): x \text{ can fly} \]
Another Example

English
Not all birds can fly.

Predicates
- \( B(x) \): \( x \) is a bird
- \( F(x) \): \( x \) can fly

The sentence in predicate logic
\[ \neg (\forall x (B(x) \rightarrow F(x))) \]
A Third Example

English
Every girl is younger than her mother.

Predicates
- \( G(x) \) : \( x \) is a girl
- \( M(y, x) \) : \( x \) is \( y \)'s mother
- \( Y(x, y) \) : \( x \) is younger than \( y \)

The sentence in predicate logic:
\[
\forall x \forall y (G(x) \land M(y, x) \rightarrow Y(x, y))
\]

The Importance of Being Formal
A Third Example

English
Every girl is younger than her mother.

Predicates

\[ G(x) : x \text{ is a girl} \]
\[ M(x, y) : x \text{ is } y\text{'s mother} \]
\[ Y(x, y) : x \text{ is younger than } y \]
### A Third Example

**English**

Every girl is younger than her mother.

**Predicates**

- \( G(x) \): \( x \) is a girl
- \( M(x, y) \): \( x \) is \( y \)'s mother
- \( Y(x, y) \): \( x \) is younger than \( y \)

**The sentence in predicate logic**

\[
\forall x \forall y (G(x) \land M(y, x) \rightarrow Y(x, y))
\]
A “Mother” Function

The sentence in predicate logic

$$\forall x \forall y (G(x) \land M(y, x) \rightarrow Y(x, y))$$

Note that $y$ is only introduced to denote the mother of $x$. 
A “Mother” Function

The sentence in predicate logic

$$\forall x \forall y (G(x) \land M(y, x) \rightarrow Y(x, y))$$

Note that $y$ is only introduced to denote the mother of $x$.

If everyone has exactly one mother, the predicate $M(y, x)$ is a function, when read from right to left.
A “Mother” Function

The sentence in predicate logic

\[ \forall x \forall y (G(x) \land M(y, x) \rightarrow Y(x, y)) \]

Note that \( y \) is only introduced to denote the mother of \( x \).

If everyone has exactly one mother, the predicate \( M(y, x) \) is a function, when read from right to left.

We introduce a function symbol \( m \) that can be applied to variables and constants as in

\[ \forall x (G(x) \rightarrow Y(x, m(x))) \]
A Drastic Example

English

Andy and Paul have the same maternal grandmother.
A Drastic Example

English

Andy and Paul have the same maternal grandmother.

The sentence in predicate logic without functions

\[ \forall x \forall y \forall u \forall v (M(x, y) \land M(y, \text{andy}) \land M(u, v) \land M(v, \text{paul}) \rightarrow x = u) \]
A Drastic Example

English

Andy and Paul have the same maternal grandmother.

The sentence in predicate logic without functions

$$\forall x \forall y \forall u \forall v (M(x, y) \land M(y, andy) \land M(u, v) \land M(v, paul) \rightarrow x = u)$$

The same sentence in predicate logic with functions

$$m(m(andy)) = m(m(paul))$$
Syntax: We formalize the language of predicate logic, including scoping and substitution.
**Syntax:** We formalize the language of predicate logic, including scoping and substitution.

**Semantics:** We describe models in which predicates, functions, and formulas have meaning.
Syntax: We formalize the language of predicate logic, including scoping and substitution.

Semantics: We describe models in which predicates, functions, and formulas have meaning.

Proof theory: We extend natural deduction from propositional to predicate logic.
Syntax: We formalize the language of predicate logic, including scoping and substitution.

Semantics: We describe models in which predicates, functions, and formulas have meaning.

Proof theory: We extend natural deduction from propositional to predicate logic

Further topics: Soundness/completeness, undecidability, incompleteness results, compactness results
At any point in time, we want to describe the features of a particular “world”, using predicates, functions, and constants. Thus, we introduce for this world:

- a set of predicate symbols $\mathcal{P}$
At any point in time, we want to describe the features of a particular “world”, using predicates, functions, and constants. Thus, we introduce for this world:

- a set of predicate symbols $\mathcal{P}$
- a set of function symbols $\mathcal{F}$
Every function symbol in $\mathcal{F}$ and predicate symbol in $\mathcal{P}$ comes with a fixed arity, denoting the number of arguments the symbol can take.
Every function symbol in $\mathcal{F}$ and predicate symbol in $\mathcal{P}$ comes with a fixed arity, denoting the number of arguments the symbol can take.

**Special case: Nullary Functions**

Function symbols with arity 0 are called *constants*. 
Arity of Functions and Predicates

Every function symbol in $\mathcal{F}$ and predicate symbol in $\mathcal{P}$ comes with a fixed arity, denoting the number of arguments the symbol can take.

**Special case: Nullary Functions**
Function symbols with arity 0 are called *constants*.

**Special case: Nullary Predicates**
Predicate symbols with arity 0 denotes predicates that do not depend on any arguments.
Every function symbol in $\mathcal{F}$ and predicate symbol in $\mathcal{P}$ comes with a fixed arity, denoting the number of arguments the symbol can take.

**Special case: Nullary Functions**

Function symbols with arity 0 are called *constants*.

**Special case: Nullary Predicates**

Predicate symbols with arity 0 denotes predicates that do not depend on any arguments. They correspond to propositional atoms.
Terms

\[ t ::= x \mid c \mid f(t, \ldots, t) \]
Terms

\[ t ::= x \mid c \mid f(t, \ldots, t) \]

where

- \( x \) ranges over a given set of variables \( \mathcal{V} \),
Terms

\[ t ::= x \mid c \mid f(t, \ldots, t) \]

where

- \( x \) ranges over a given set of variables \( \mathcal{V} \),
- \( c \) ranges over nullary function symbols in \( \mathcal{F} \), and
Terms

t ::= x | c | f(t, ..., t)

where
- x ranges over a given set of variables $\mathcal{V}$,
- c ranges over nullary function symbols in $\mathcal{F}$, and
- f ranges over function symbols in $\mathcal{F}$ with arity $n > 0$. 
Examples of Terms

If $n$ is nullary, $f$ is unary, and $g$ is binary, then examples of terms are:

- $g(f(n), n)$
- $f(g(n, f(n)))$
More Examples of Terms

If 0, 1, 2 are nullary (constants), s is unary, and +, − and * are binary, then

\[ *(-(2, +(s(x), y)), x) \]

is a term.
More Examples of Terms

If 0, 1, 2 are nullary (constants), s is unary, and +, − and * are binary, then

\[ *(−(2, +(s(x), y)), x) \]

is a term.

Occasionally, we allow ourselves to use infix notation for function symbols as in

\[ (2 − (s(x) + y)) * x \]
Formulas

\[ \phi \ ::= \quad P(t, \ldots, t) \mid \lnot \phi \mid (\phi \land \phi) \mid (\phi \lor \phi) \mid (\phi \rightarrow \phi) \mid (\forall x \phi) \mid (\exists x \phi) \]
Formulas

\[
\phi ::= \ P(t, \ldots, t) \mid (\neg \phi) \mid (\phi \land \phi) \mid (\phi \lor \phi) \mid \\
(\phi \rightarrow \phi) \mid (\forall x \phi) \mid (\exists x \phi)
\]

where

- \( P \in \mathcal{P} \) is a predicate symbol of arity \( n \geq 0 \),
Formulas

\[ \phi ::= P(t, \ldots, t) \mid (\neg \phi) \mid (\phi \land \phi) \mid (\phi \lor \phi) \mid (\phi \rightarrow \phi) \mid (\forall x \phi) \mid (\exists x \phi) \]

where

- \( P \in \mathcal{P} \) is a predicate symbol of arity \( n \geq 0 \),
- \( t \) are terms over \( F \) and \( V \), and
Formulas

\[ \phi \ ::= \ P(t, \ldots, t) \mid (\neg \phi) \mid (\phi \land \phi) \mid (\phi \lor \phi) \mid (\phi \rightarrow \phi) \mid (\forall x \phi) \mid (\exists x \phi) \]

where
- \( P \in \mathcal{P} \) is a predicate symbol of arity \( n \geq 0 \),
- \( t \) are terms over \( \mathcal{F} \) and \( \mathcal{V} \), and
- \( x \) are variables in \( \mathcal{V} \).
Conventions

Just like for propositional logic, we introduce convenient conventions to reduce the number of parentheses:

- \( \neg, \forall x \) and \( \exists x \) bind most tightly;
- then \( \land \) and \( \lor \);
- then \( \rightarrow \), which is right-associative.
Parse Trees

\[ \forall x ((P(x) \rightarrow Q(x)) \land S(x, y)) \]

has parse tree

```
     ∨
    / \  
   /   \ 
  /     \
∀x     S
  /      /
 /       /
P  Q    x  y
  /   \
 X  X
```
Another Example

Every son of my father is my brother.

Predicates

\[ S(x, y): \text{x is a son of y} \]
\[ B(x, y): \text{x is a brother of y} \]
Another Example

Every son of my father is my brother.

Predicates

$S(x, y)$: $x$ is a son of $y$

$B(x, y)$: $x$ is a brother of $y$

Functions

$m$: constant for “me”

$f(x)$: father of $x$
Another Example

Every son of my father is my brother.

**Predicates**

- $S(x, y)$: $x$ is a son of $y$
- $B(x, y)$: $x$ is a brother of $y$

**Functions**

- $m$: constant for “me”
- $f(x)$: father of $x$

The sentence in predicate logic

$$\forall x (S(x, f(m)) \rightarrow B(x, m))$$
Another Example

Every son of my father is my brother.

Predicates

\[ S(x, y) : x \text{ is a son of } y \]
\[ B(x, y) : x \text{ is a brother of } y \]

Functions

\[ m : \text{ constant for "me"} \]
\[ f(x) : \text{ father of } x \]

The sentence in predicate logic

\[ \forall x (S(x, f(m)) \rightarrow B(x, m)) \]
Equality as a common predicate, usually used in infix notation.

\[ = \in \mathcal{P} \]
Equality as Predicate

Equality is a common predicate, usually used in infix notation.

\[ = \in P \]

Example

Instead of the formula

\[ = (f(x), g(x)) \]

we usually write the formula

\[ f(x) = g(x) \]
Consider the formula

$$\forall x((P(x) \rightarrow Q(x)) \land S(x, y))$$
Free and Bound Variables

Consider the formula

$$\forall x((P(x) \rightarrow Q(x)) \land S(x, y))$$

What is the relationship between variable “binder” $x$ and occurrences of $x$?
Consider the formula

$$\forall x((P(x) \rightarrow Q(x)) \land S(x, y))$$

What is the relationship between variable “binder” $x$ and occurrences of $x$?

The diagram illustrates the structure of the formula:

- $\forall x$
- $\land$
- $\rightarrow$
- $S$
- $P$
- $Q$
- $x$
- $y$
- $X$
- $X$
Consider the formula

$$(\forall x (P(x) \land Q(x))) \rightarrow (\neg P(x) \lor Q(y))$$

Which variable occurrences are free; which are bound?
Consider the formula

$$ (\forall x (P(x) \land Q(x))) \rightarrow (\neg P(x) \lor Q(y)) $$

Which variable occurrences are free; which are bound?
Consider the formula

$$(\forall x(P(x) \land Q(x))) \rightarrow (\neg P(x) \lor Q(y))$$

Which variable occurrences are free; which are bound?
Variables are placeholders. Replacing them by terms is called substitution.
Substitution

Variables are *placeholders*. *Replacing* them by terms is called *substitution*.

**Definition**

Given a variable $x$, a term $t$ and a formula $\phi$, we define $[x \Rightarrow t] \phi$ to be the formula obtained by replacing each free occurrence of variable $x$ in $\phi$ with $t$. 
Variables are *placeholders*. Replacing them by terms is called *substitution*.

**Definition**

Given a variable $x$, a term $t$ and a formula $\phi$, we define $[x \Rightarrow t] \phi$ to be the formula obtained by replacing each free occurrence of variable $x$ in $\phi$ with $t$.

**Example**

\[
[x \Rightarrow f(x, y)]((\forall x (P(x) \land Q(x))) \rightarrow (\neg P(x) \lor Q(y)))
\]

\[
= \forall x (P(x) \land Q(x))) \rightarrow (\neg P(f(x, y)) \lor Q(y))
\]
Example as Parse Tree

\[ x \Rightarrow f(x, y) \](\((\forall x(P(x) \land Q(x))) \rightarrow (\neg P(x) \lor Q(y))) \]

\[ = (\forall x(P(x) \land Q(x))) \rightarrow (\neg P(f(x, y)) \lor Q(y)) \]
Example as Parse Tree

\[ [x \Rightarrow f(x, y)]((\forall x (P(x) \land Q(x)))) \rightarrow (\neg P(x) \lor Q(y)) \]

\[ = (\forall x (P(x) \land Q(x)))) \rightarrow (\neg P(f(x, y)) \lor Q(y)) \]
Example as Parse Tree

\[ \rightarrow \]
\[ \forall x \wedge \left( P \wedge \neg P \right) \wedge Q \]
\[ P \quad Q \quad P \quad y \]
\[ x \quad x \quad f \]
\[ x \quad y \]
Syntax of Predicate Logic

1. Syntax of Predicate Logic
2. Predicate Logic as a Formal Language
3. Semantics of Predicate Logic
   - Models
   - Equality
   - Free Variables
   - Satisfaction and Entailment
4. Proof Theory
5. Equivalences

The Importance of Being Formal
Let $\mathcal{F}$ contain function symbols and $\mathcal{P}$ contain predicate symbols. A model $\mathcal{M}$ for $(\mathcal{F}, \mathcal{P})$ consists of:

1. A non-empty set $A$, the *universe*;
2. for each nullary function symbol $f \in \mathcal{F}$ a concrete element $f^\mathcal{M} \in A$;
3. for each $f \in \mathcal{F}$ with arity $n > 0$, a concrete function $f^\mathcal{M} : A^n \rightarrow A$;
4. for each $P \in \mathcal{P}$ with arity $n > 0$, a function $P^\mathcal{M} : U^n \rightarrow \{F, T\}$.
5. for each $P \in \mathcal{P}$ with arity $n = 0$, a value from $\{F, T\}$. 

The Importance of Being Formal
Example

Let $\mathcal{F} = \{ e, \cdot \}$ and $\mathcal{P} = \{ \leq \}$.
Let model $\mathcal{M}$ for $(\mathcal{F}, \mathcal{P})$ be defined as follows:

1. Let $A$ be the set of binary strings over the alphabet $\{0, 1\}$;
2. let $e^\mathcal{M} = \epsilon$, the empty string;
3. let $\cdot^\mathcal{M}$ be defined such that $s_1 \cdot^\mathcal{M} s_2$ is the concatenation of the strings $s_1$ and $s_2$; and
4. let $\leq^\mathcal{M}$ be defined such that $s_1 \leq^\mathcal{M} s_2$ iff $s_1$ is a prefix of $s_2$. 

Example (continued)

1. Let $A$ be the set of binary strings over the alphabet $\{0, 1\}$;
2. let $e^M = \epsilon$, the empty string;
3. let $\cdot^M$ be defined such that $s_1 \cdot^M s_2$ is the concatenation of the strings $s_1$ and $s_2$; and
4. let $\leq^M$ be defined such that $s_1 \leq^M s_2$ iff $s_1$ is a prefix of $s_2$. 

Some Elements of $A$

10001
1010
10101100
000
Example (continued)

1. Let $A$ be the set of binary strings over the alphabet $\{0, 1\}$;
2. Let $e^M = \epsilon$, the empty string;
3. Let $\cdot^M$ be defined such that $s_1 \cdot^M s_2$ is the concatenation of the strings $s_1$ and $s_2$; and
4. Let $\leq^M$ be defined such that $s_1 \leq^M s_2$ iff $s_1$ is a prefix of $s_2$.

Some Elements of $A$

- $10001$
- $\epsilon$
- $1010 \cdot^M 1100$
Example (continued)

1. Let $A$ be the set of binary strings over the alphabet $\{0, 1\}$;
2. let $e^M = \epsilon$, the empty string;
3. let $\cdot^M$ be defined such that $s_1 \cdot^M s_2$ is the concatenation of the strings $s_1$ and $s_2$; and
4. let $\leq^M$ be defined such that $s_1 \leq^M s_2$ iff $s_1$ is a prefix of $s_2$.

Some Elements of $A$

- $10001$
- $\epsilon$
- $1010 \cdot^M 1100 = 10101100$
Example (continued)

1. Let $A$ be the set of binary strings over the alphabet $\{0, 1\}$;
2. let $e^M = \epsilon$, the empty string;
3. let $\cdot^M$ be defined such that $s_1 \cdot^M s_2$ is the concatenation of the strings $s_1$ and $s_2$; and
4. let $\leq^M$ be defined such that $s_1 \leq^M s_2$ iff $s_1$ is a prefix of $s_2$.

Some Elements of $A$

- 10001
- $\epsilon$
- $1010 \cdot^M 1100 = 10101100$
- $000 \cdot^M \epsilon$
Example (continued)

1. Let $A$ be the set of binary strings over the alphabet $\{0, 1\}$;
2. let $e^M = \epsilon$, the empty string;
3. let $\cdot^M$ be defined such that $s_1 \cdot^M s_2$ is the concatenation of the strings $s_1$ and $s_2$; and
4. let $\leq^M$ be defined such that $s_1 \leq^M s_2$ iff $s_1$ is a prefix of $s_2$.

Some Elements of $A$

- 10001
- $\epsilon$
- $1010 \cdot^M 1100 = 10101100$
- $000 \cdot^M \epsilon = 000$
Equality Revisited

Interpretation of equality

Usually, we require that the equality predicate $=$ is interpreted as same-ness.
Equality Revisited

Interpretation of equality

Usually, we require that the equality predicate $\equiv$ is interpreted as same-ness.

Extensionality restriction

This means that allowable models are restricted to those in which $a \equiv^M b$ holds if and only if $a$ and $b$ are the same elements of the model’s universe.
Example (continued)

1. Let $A$ be the set of binary strings over the alphabet $\{0, 1\}$;
2. let $e^M = \epsilon$, the empty string;
3. let $\cdot^M$ be defined such that $s_1 \cdot^M s_2$ is the concatenation of
   the strings $s_1$ and $s_2$; and
4. let $\leq^M$ be defined such that $s_1 \leq^M s_2$ iff $s_1$ is a prefix of $s_2$.  

Equality

The Importance of Being Formal
Example (continued)

1. Let $A$ be the set of binary strings over the alphabet $\{0, 1\}$;
2. let $e^M = \epsilon$, the empty string;
3. let $\cdot^M$ be defined such that $s_1 \cdot^M s_2$ is the concatenation of the strings $s_1$ and $s_2$; and
4. let $\leq^M$ be defined such that $s_1 \leq^M s_2$ iff $s_1$ is a prefix of $s_2$.

Equality in $M$

- $000 =^M 000$
- $001 \neq^M 100$
Another Example

Let \( F = \{ z, s \} \) and \( P = \{ \leq \} \).
Another Example

Let $\mathcal{F} = \{z, s\}$ and $\mathcal{P} = \{\leq\}$. Let model $\mathcal{M}$ for $(\mathcal{F}, \mathcal{P})$ be defined as follows:

1. Let $A$ be the set of natural numbers;
Another Example

Let \( \mathcal{F} = \{ z, s \} \) and \( \mathcal{P} = \{ \leq \} \).

Let model \( \mathcal{M} \) for \((\mathcal{F}, \mathcal{P})\) be defined as follows:

1. Let \( A \) be the set of natural numbers;
2. let \( z^\mathcal{M} = 0 \);
Let $\mathcal{F} = \{z, s\}$ and $\mathcal{P} = \{\leq\}$.

Let model $\mathcal{M}$ for $(\mathcal{F}, \mathcal{P})$ be defined as follows:

1. Let $A$ be the set of natural numbers;
2. let $z^M = 0$;
3. let $s^M$ be defined such that $s(n) = n + 1$; and
Another Example

Let $\mathcal{F} = \{z, s\}$ and $\mathcal{P} = \{\leq\}$.
Let model $\mathcal{M}$ for $(\mathcal{F}, \mathcal{P})$ be defined as follows:

1. Let $A$ be the set of natural numbers;
2. let $z^\mathcal{M} = 0$;
3. let $s^\mathcal{M}$ be defined such that $s(n) = n + 1$; and
4. let $\leq^\mathcal{M}$ be defined such that $n_1 \leq^\mathcal{M} n_2$ iff the natural number $n_1$ is less than or equal to $n_2$. 
How To Handle Free Variables?

Idea

We can give meaning to formulas with free variables by providing an environment (lookup table) that assigns variables to elements of our universe:

\[ I : \forall \rightarrow A. \]
How To Handle Free Variables?

Idea

We can give meaning to formulas with free variables by providing an environment (lookup table) that assigns variables to elements of our universe:

\[ l : \mathcal{V} \rightarrow A. \]

Environment extension

We define environment extension such that \( l[x \mapsto a] \) is the environment that maps \( x \) to \( a \) and any other variable \( y \) to \( l(y) \).
The model $\mathcal{M}$ satisfies $\phi$ with respect to environment $l$, written $\mathcal{M} \models_l \phi$.
The model $\mathcal{M}$ satisfies $\phi$ with respect to environment $l$, written $\mathcal{M} \models_l \phi$:

- in case $\phi$ is of the form $P(t_1, t_2, \ldots, t_n)$, if $a_1, a_2, \ldots, a_n$ are the results of evaluating $t_1, t_2, \ldots, t_n$ with respect to $l$, and if $P^\mathcal{M}(a_1, a_2, \ldots, a_n) = T$;
Satisfaction Relation

The model $\mathcal{M}$ satisfies $\phi$ with respect to environment $I$, written $\mathcal{M} \models_I \phi$:

- in case $\phi$ is of the form $P(t_1, t_2, \ldots, t_n)$, if $a_1, a_2, \ldots, a_n$ are the results of evaluating $t_1, t_2, \ldots, t_n$ with respect to $I$, and if $P^\mathcal{M}(a_1, a_2, \ldots, a_n) = T$;

- in case $\phi$ is of the form $P$, if $P^\mathcal{M} = T$;
Satisfaction Relation

The model $\mathcal{M}$ satisfies $\phi$ with respect to environment $I$, written $\mathcal{M} \models_I \phi$:

- in case $\phi$ is of the form $P(t_1, t_2, \ldots, t_n)$, if $a_1, a_2, \ldots, a_n$ are the results of evaluating $t_1, t_2, \ldots, t_n$ with respect to $I$, and if $P^{\mathcal{M}}(a_1, a_2, \ldots, a_n) = T$;
- in case $\phi$ is of the form $P$, if $P^{\mathcal{M}} = T$;
- in case $\phi$ has the form $\forall x \psi$, if the $\mathcal{M} \models_I [x \mapsto a] \psi$ holds for all $a \in A$;
The model $\mathcal{M}$ satisfies $\phi$ with respect to environment $l$, written $\mathcal{M} \models_{l} \phi$:

- in case $\phi$ is of the form $P(t_1, t_2, \ldots, t_n)$, if $a_1, a_2, \ldots, a_n$ are the results of evaluating $t_1, t_2, \ldots, t_n$ with respect to $l$, and if $P^\mathcal{M}(a_1, a_2, \ldots, a_n) = T$;
- in case $\phi$ is of the form $P$, if $P^\mathcal{M} = T$;
- in case $\phi$ has the form $\forall x \psi$, if the $\mathcal{M} \models_{l[x \rightarrow a]} \psi$ holds for all $a \in A$;
- in case $\phi$ has the form $\exists x \psi$, if the $\mathcal{M} \models_{l[x \rightarrow a]} \psi$ holds for some $a \in A$;
Satisfaction Relation (continued)

- in case \( \phi \) has the form \( \neg \psi \), if \( M \models \psi \) does not hold;
in case $\phi$ has the form $\neg\psi$, if $M \models I \psi$ does not hold;

in case $\phi$ has the form $\psi_1 \lor \psi_2$, if $M \models I \psi_1$ holds or $M \models I \psi_2$ holds;
in case $\phi$ has the form $\neg \psi$, if $\mathcal{M} \models \psi$ does not hold;

in case $\phi$ has the form $\psi_1 \lor \psi_2$, if $\mathcal{M} \models \psi_1$ holds or $\mathcal{M} \models \psi_2$ holds;

in case $\phi$ has the form $\psi_1 \land \psi_2$, if $\mathcal{M} \models \psi_1$ holds and $\mathcal{M} \models \psi_2$ holds; and
in case $\phi$ has the form $\neg \psi$, if $M \models_{I} \psi$ does not hold;

in case $\phi$ has the form $\psi_1 \lor \psi_2$, if $M \models_{I} \psi_1$ holds or $M \models_{I} \psi_2$ holds;

in case $\phi$ has the form $\psi_1 \land \psi_2$, if $M \models_{I} \psi_1$ holds and $M \models_{I} \psi_2$ holds; and

in case $\phi$ has the form $\psi_1 \rightarrow \psi_2$, if $M \models_{I} \psi_2$ holds whenever $M \models_{I} \psi_1$ holds.
If a formula $\phi$ has no free variables, we call $\phi$ a *sentence*. 
If a formula $\phi$ has no free variables, we call $\phi$ a *sentence*. $\mathcal{M} \models_{I} \phi$ holds or does not hold regardless of the choice of $I$. Thus we write $\mathcal{M} \models \phi$ or $\mathcal{M} \not\models \phi$. 
Semantic Entailment and Satisfiability

Let $\Gamma$ be a possibly infinite set of formulas in predicate logic and $\psi$ a formula.
Semantic Entailment and Satisfiability

Let $\Gamma$ be a possibly infinite set of formulas in predicate logic and $\psi$ a formula.

**Entailment**

$\Gamma \models \psi$ iff for all models $M$ and environments $I$, whenever $M \models_I \phi$ holds for all $\phi \in \Gamma$, then $M \models_I \psi$. 

Satisfiability of Formulas

$\psi$ is satisfiable iff there is some model $M$ and some environment $I$ such that $M \models_I \psi$ holds.

Satisfiability of Formula Sets

$\Gamma$ is satisfiable iff there is some model $M$ and some environment $I$ such that $M \models_I \phi$, for all $\phi \in \Gamma$. 

The Importance of Being Formal
Let $\Gamma$ be a possibly infinite set of formulas in predicate logic and $\psi$ a formula.

**Entailment**

$\Gamma \models \psi$ iff for all models $M$ and environments $I$, whenever $M \models_I \phi$ holds for all $\phi \in \Gamma$, then $M \models_I \psi$.

**Satisfiability of Formulas**

$\psi$ is satisfiable iff there is some model $M$ and some environment $I$ such that $M \models_I \psi$ holds.
Semantic Entailment and Satisfiability

Let $\Gamma$ be a possibly infinite set of formulas in predicate logic and $\psi$ a formula.

**Entailment**

$\Gamma \models \psi$ iff for all models $\mathcal{M}$ and environments $I$, whenever $\mathcal{M} \models_I \phi$ holds for all $\phi \in \Gamma$, then $\mathcal{M} \models_I \psi$.

**Satisfiability of Formulas**

$\psi$ is satisfiable iff there is some model $\mathcal{M}$ and some environment $I$ such that $\mathcal{M} \models_I \psi$ holds.

**Satisfiability of Formula Sets**

$\Gamma$ is satisfiable iff there is some model $\mathcal{M}$ and some environment $I$ such that $\mathcal{M} \models_I \phi$, for all $\phi \in \Gamma$. 
Semantic Entailment and Satisfiability

Let $\Gamma$ be a possibly infinite set of formulas in predicate logic and $\psi$ a formula.

Validity

$\psi$ is valid iff for all models $\mathcal{M}$ and environments $I$, we have $\mathcal{M} \models_I \psi$. 
Entailment ranges over models

Semantic entailment between sentences: $\phi_1, \phi_2, \ldots, \phi_n \models \psi$

requires that in all models that satisfy $\phi_1, \phi_2, \ldots, \phi_n$, the sentence $\psi$ is satisfied.
The Problem with Predicate Logic

Entailment ranges over models

Semantic entailment between sentences: \( \phi_1, \phi_2, \ldots, \phi_n \models \psi \)
requires that in all models that satisfy \( \phi_1, \phi_2, \ldots, \phi_n \), the sentence \( \psi \) is satisfied.

How to effectively argue about all possible models?

Usually the number of models is infinite; it is very hard to argue on the semantic level in predicate logic.
The Problem with Predicate Logic

Entailment ranges over models

Semantic entailment between sentences: $\phi_1, \phi_2, \ldots, \phi_n \models \psi$
requires that in all models that satisfy $\phi_1, \phi_2, \ldots, \phi_n$, the sentence $\psi$ is satisfied.

How to effectively argue about all possible models?

Usually the number of models is infinite; it is very hard to argue on the semantic level in predicate logic.

Idea from propositional logic

Can we use natural deduction for showing entailment?
If we consider propositions as nullary predicates, propositional logic is a sub-language of predicate logic.
Natural Deduction for Predicate Logic

Relationship between propositional and predicate logic
If we consider propositions as nullary predicates, propositional logic is a sub-language of predicate logic.

Inheriting natural deduction
We can translate the rules for natural deduction in propositional logic directly to predicate logic.
Natural Deduction for Predicate Logic

If we consider propositions as nullary predicates, propositional logic is a sub-language of predicate logic.

We can translate the rules for natural deduction in propositional logic directly to predicate logic.

Example

\[
\phi, \psi \quad \vdash [\land i] \\
\phi \land \psi
\]
Built-in Rules for Equality

\[
\begin{align*}
&\text{[= i]} \\
&\frac{t = t}{t_1 = t_2} \\
&\frac{[x \Rightarrow t_1] \phi}{[x \Rightarrow t_2] \phi}
\end{align*}
\]

\[\text{[= e]}\]
Properties of Equality

We show:

\[ f(x) = g(x) \vdash h(g(x)) = h(f(x)) \]

using

\[ t_1 = t_2 \quad [x \Rightarrow t_1] \phi \]

\[ t = t \quad [x \Rightarrow t_2] \phi \]
Properties of Equality

We show:

\[ f(x) = g(x) \vdash h(g(x)) = h(f(x)) \]

using

\[
\begin{align*}
& t_1 = t_2 \quad [x \Rightarrow t_1] \phi \\
& \top = \top \\
\end{align*}
\]

\[
\begin{align*}
& [x \Rightarrow t_2] \phi \\
& \phi_{1,2} = e
\end{align*}
\]

1. \( f(x) = g(x) \)  \quad premise
2. \( h(f(x)) = h(f(x)) \)  \quad \( = i \)
3. \( h(g(x)) = h(f(x)) \)  \quad \( = e_{1,2} \)
Once you have proven $\forall x \phi$, you can replace $x$ by any term $t$ in $\phi$, provided that $t$ is free for $x$ in $\phi$.
Elimination of Universal Quantification

\[ \forall x \phi \]

\[ \frac{}{[\forall x \ e]} \]

\[ [x \Rightarrow t] \phi \]

Once you have proven \( \forall x \phi \), you can replace \( x \) by any term \( t \) in \( \phi \)
Once you have proven $\forall x \phi$, you can replace $x$ by any term $t$ in $\phi$, provided that $t$ is free for $x$ in $\phi$. 
Example

\[
\begin{align*}
\forall x \phi \\
\quad \quad \quad \quad \quad \quad [\forall x \ e] \\
\quad [x \Rightarrow t] \phi
\end{align*}
\]

We prove: \( S(g(john)) \), \( \forall x (S(x) \rightarrow \neg L(x)) \models \neg L(g(john)) \)
Example

\[ \forall x \phi \]

\[ [\forall x \ e] \]

\[ [x \Rightarrow t] \phi \]

We prove: \( S(g(\text{john})) \), \( \forall x (S(x) \rightarrow \neg L(x)) \vdash \neg L(g(\text{john})) \)

1. \( S(g(\text{john})) \) \hspace{2cm} \text{premise}
2. \( \forall x (S(x) \rightarrow \neg L(x)) \) \hspace{2cm} \text{premise}
3. \( S(g(\text{john})) \rightarrow \neg L(g(\text{john})) \) \hspace{2cm} \( \forall x \ e \ 2 \)
4. \( \neg L(g(\text{john})) \) \hspace{2cm} \( \rightarrow \ e \ 3,1 \)
If we manage to establish a formula $\phi$ about a fresh variable $x_0$, we can assume $\forall x \phi$.

The variable $x_0$ must be fresh; we cannot introduce the same variable twice in nested boxes.
If we manage to establish a formula \( \phi \) about a fresh variable \( x_0 \), we can assume \( \forall x \phi \).
If we manage to establish a formula $\phi$ about a fresh variable $x_0$, we can assume $\forall x \phi$.

The variable $x_0$ must be fresh; we cannot introduce the same variable twice in nested boxes.
Example

\[ \forall x (P(x) \rightarrow Q(x)), \forall x P(x) \vdash \forall x Q(x) \text{ via } \forall x \phi \]

\[ [x \Rightarrow x_0] \phi \]

\[ x_0 \]

The Importance of Being Formal
Example

\[ \forall x (P(x) \rightarrow Q(x)), \forall x P(x) \vdash \forall x Q(x) \quad \text{via} \quad \forall x \phi \]

<table>
<thead>
<tr>
<th>Step</th>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \forall x (P(x) \rightarrow Q(x)) )</td>
<td>premise</td>
</tr>
<tr>
<td>2</td>
<td>( \forall x P(x) )</td>
<td>premise</td>
</tr>
<tr>
<td>3</td>
<td>( P(x_0) \rightarrow Q(x_0) )</td>
<td>( \forall x \ e 1 )</td>
</tr>
<tr>
<td>4</td>
<td>( P(x_0) )</td>
<td>( \forall x \ e 2 )</td>
</tr>
<tr>
<td>5</td>
<td>( Q(x_0) )</td>
<td>( \rightarrow e 3,4 )</td>
</tr>
<tr>
<td>6</td>
<td>( \forall x Q(x) )</td>
<td>( \forall x \ i 3–5 )</td>
</tr>
</tbody>
</table>
In order to prove $\exists x \phi$, it suffices to find a term $t$ as "witness", provided that $t$ is free for $x$ in $\phi$.

The Importance of Being Formal
Introduction of Existential Quantification

In order to prove $\exists x \phi$, it suffices to find a term $t$ as “witness”

$$[x \Rightarrow t] \phi$$

$$\frac{\exists x \ i}{\exists x \phi}$$
In order to prove $\exists x \phi$, it suffices to find a term $t$ as “witness”, provided that $t$ is free for $x$ in $\phi$. 

$$[x \Rightarrow t] \phi \quad \frac{[\exists x \ i]}{\exists x \phi}$$
Example

\[ \forall x \phi \vdash \exists x \phi \]

Remark: Compare this with Traditional Logic. Because \( U \) must not be empty, we should be able to prove the sequent above.
Example

\[ \forall x \phi \vdash \exists x \phi \]

Recall: Definition of Models

A model \( \mathcal{M} \) for \((\mathcal{F}, \mathcal{P})\) consists of:

1. A non-empty set \( U \), the universe;
2. ...
Example

\[ \forall x \phi \vdash \exists x \phi \]

Recall: Definition of Models

A model \( \mathcal{M} \) for \((\mathcal{F}, \mathcal{P})\) consists of:

1. A *non-empty* set \( U \), the *universe*;
2. ...

Remark

Compare this with Traditional Logic.
\[ \forall x \phi \vdash \exists x \phi \]

**Recall: Definition of Models**

A model $\mathcal{M}$ for $(\mathcal{F}, \mathcal{P})$ consists of:

1. A *non-empty* set $U$, the *universe*;
2. ...  

**Remark**

Compare this with Traditional Logic.

Because $U$ must not be empty, we should be able to prove the sequent above.
Example (continued)

\(\forall x \phi \vdash \exists x \phi\)
Example (continued)

∀xφ ⊢ ∃xφ

1  ∀xφ  premise
2  [x ⇒ x]φ  ∀x  e 1
3  ∃xφ  ∃x  i 2
Making use of ∃

If we know ∃xφ, we know that there exist at least one object x for which φ holds. We call that element x₀, and assume [x ⇒ x₀]φ.

Without assumptions on x₀, we prove χ (x₀ not in χ).
Elimination of Existential Quantification

If we know $\exists x \phi$, we know that there exist at least one object $x$ for which $\phi$ holds.
Making use of $\exists$

If we know $\exists x \phi$, we know that there exist at least one object $x$ for which $\phi$ holds. We call that element $x_0$, and assume $[x \Rightarrow x_0] \phi$. Without assumptions on $x_0$, we prove $\chi$ ($x_0$ not in $\chi$).
Elimination of Existential Quantification

If we know $\exists x \phi$, we know that there exist at least one object $x$ for which $\phi$ holds. We call that element $x_0$, and assume $[x \Rightarrow x_0] \phi$. Without assumptions on $x_0$, we prove $\chi$.

The Importance of Being Formal
Example

\[ \forall x (P(x) \rightarrow Q(x)), \exists x P(x) \vdash \exists x Q(x) \]

<table>
<thead>
<tr>
<th>Step</th>
<th>Formula</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \forall x (P(x) \rightarrow Q(x)) )</td>
<td>premise</td>
</tr>
<tr>
<td>2</td>
<td>( \exists x P(x) )</td>
<td>premise</td>
</tr>
<tr>
<td>3</td>
<td>( P(x_0) )</td>
<td>assumption, ( x_0 )</td>
</tr>
<tr>
<td>4</td>
<td>( P(x_0) \rightarrow Q(x_0) )</td>
<td>( \forall x e 1 )</td>
</tr>
<tr>
<td>5</td>
<td>( Q(x_0) )</td>
<td>( \rightarrow e 4,3 )</td>
</tr>
<tr>
<td>6</td>
<td>( \exists x Q(x) )</td>
<td>( \exists x i 5 )</td>
</tr>
<tr>
<td>7</td>
<td>( \exists x Q(x) )</td>
<td>( \exists x e 2,3,6 )</td>
</tr>
</tbody>
</table>
Example

\forall x (P(x) \rightarrow Q(x)), \exists x P(x) \vdash \exists x Q(x)

1. \forall x (P(x) \rightarrow Q(x)) \quad \text{premise}
2. \exists x P(x) \quad \text{premise}
3. P(x_0) \quad \text{assumption} \quad x_0
4. P(x_0) \rightarrow Q(x_0) \quad \forall x \ e \ 1
5. Q(x_0) \quad \rightarrow \ e \ 4,3
6. \exists x Q(x) \quad \exists x \ i \ 5
7. \exists x Q(x) \quad \exists x \ e \ 2,3–6

Note that \exists x Q(x) within the box does not contain \(x_0\), and therefore can be “exported” from the box.
Another Example

1. \( \forall x (Q(x) \rightarrow R(x)) \)  
   premise
2. \( \exists x (P(x) \land Q(x)) \)  
   premise
3. \( P(x_0) \land Q(x_0) \)  
   assumption \( x_0 \)
4. \( Q(x_0) \rightarrow R(x_0) \)  
   \( \forall x \ e \ 1 \)
5. \( Q(x_0) \)  
   \( \land e_2 \ 3 \)
6. \( R(x_0) \)  
   \( \rightarrow e \ 4,5 \)
7. \( P(x_0) \)  
   \( \land e_1 \ 3 \)
8. \( P(x_0) \land R(x_0) \)  
   \( \land i \ 7, 6 \)
9. \( \exists x (P(x) \land R(x)) \)  
   \( \exists x \ i \ 8 \)
10. \( \exists x (P(x) \land R(x)) \)  
    \( \exists x \ e \ 2,3\–9 \)
### Syntax of Predicate Logic

- **Predicate Logic as a Formal Language**
- **Semantics of Predicate Logic**
- **Proof Theory**
- **Equivalences**
- **Soundness and Completeness**

### The Importance of Being Formal

#### Variables must be fresh! This is not a proof!

1. **Premise**
   - $\exists x P(x)$
2. **Premise**
   - $\forall x (P(x) \rightarrow Q(x))$
3. **Assumption**
   - $x_0$
4. **Assumption**
   - $P(x_0)$
5. **Premise**
   - $\forall x (P(x) \rightarrow Q(x))$
   - $\forall x e 2$
6. **Conclusion**
   - $Q(x_0)$
   - $\rightarrow e 5,4$
7. **Conclusion**
   - $\exists x e 1, 4–6$
8. **Conclusion**
   - $\forall y Q(y)$
   - $\forall y i 3–7$

---

**06—From Propositional to Predicate Logic**
Two-way-provable

We write $\phi \dashv \vdash \psi$ iff $\phi \vdash \psi$ and also $\psi \vdash \phi$. 
Equivalences

Two-way-provable

We write $\phi \vdash \psi$ iff $\phi \vdash \psi$ and also $\psi \vdash \phi$.

Some simple equivalences

$$\neg \forall x \phi \iff \exists x \neg \phi$$
Equivalences

Two-way-provable

We write \( \phi \vdash \psi \) iff \( \phi \vdash \psi \) and also \( \psi \vdash \phi \).

Some simple equivalences

\[
\neg \forall x \phi \vdash \exists x \neg \phi \\
\neg \exists x \phi \vdash \forall x \neg \phi
\]
Two-way-provable

We write \( \phi \models \psi \) iff \( \phi \vdash \psi \) and also \( \psi \vdash \phi \).

Some simple equivalences

- \( \neg \forall x \phi \models \exists x \neg \phi \)
- \( \neg \exists x \phi \models \forall x \neg \phi \)
- \( \forall x \forall y \phi \models \forall y \forall x \phi \)
### Equivalences

#### Two-way-provable

We write \( \phi \vdash \psi \) iff \( \phi \vdash \psi \) and also \( \psi \vdash \phi \).

#### Some simple equivalences

- \( \neg \forall x \phi \vdash \exists x \neg \phi \)
- \( \neg \exists x \phi \vdash \forall x \neg \phi \)
- \( \forall x \forall y \phi \vdash \forall y \forall x \phi \)
- \( \exists x \exists y \phi \vdash \exists y \exists x \phi \)
Two-way-provable

We write $\phi \vdash \psi$ iff $\phi \models \psi$ and also $\psi \models \phi$.

Some simple equivalences

\[
\begin{align*}
\neg \forall x \phi & \quad \vdash \quad \exists x \neg \phi \\
\neg \exists x \phi & \quad \vdash \quad \forall x \neg \phi \\
\forall x \forall y \phi & \quad \vdash \quad \forall y \forall x \phi \\
\exists x \exists y \phi & \quad \vdash \quad \exists y \exists x \phi \\
\forall x \phi \land \forall x \psi & \quad \vdash \quad \forall x (\phi \land \psi)
\end{align*}
\]
Two-way-provable

We write $\phi \dashv \vdash \psi$ iff $\phi \vdash \psi$ and also $\psi \vdash \phi$.

Some simple equivalences

- $\neg \forall x \phi \dashv \vdash \exists x \neg \phi$
- $\neg \exists x \phi \dashv \vdash \forall x \neg \phi$
- $\forall x \forall y \phi \dashv \vdash \forall y \forall x \phi$
- $\exists x \exists y \phi \dashv \vdash \exists y \exists x \phi$
- $\forall x \phi \wedge \forall x \psi \dashv \vdash \forall x (\phi \wedge \psi)$
- $\exists x \phi \vee \exists x \psi \dashv \vdash \exists x (\phi \vee \psi)$
\[
\neg \forall x \phi \vdash \exists x \neg \phi
\]

1. \(\neg \forall x \phi\) premise
2. \(\neg \exists x \neg \phi\) assumption
3. \(x_0\)
4. \(\neg [x \Rightarrow x_0] \phi\) assumption
5. \(\exists x \neg \phi\) \(\exists x\ i\ 4\)
6. \(\bot\) \(-\ e\ 5,\ 2\)
7. \([x \Rightarrow x_0] \phi\) PBC 4–6
8. \(\forall x \phi\) \(\forall x\ i\ 3–7\)
9. \(\bot\) \(-\ e\ 8,\ 1\)
10. \(\exists x \neg \phi\) PBC 2–9
Assume that $x$ and $y$ are different variables.
### Assume that $x$ and $y$ are different variables.

<table>
<thead>
<tr>
<th>Step</th>
<th>Formula</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\exists x \exists y \phi$</td>
<td>premise</td>
</tr>
<tr>
<td>2</td>
<td>$[x \Rightarrow x_0](\exists y \phi)$</td>
<td>assumption $x_0$</td>
</tr>
<tr>
<td>3</td>
<td>$\exists y([x \Rightarrow x_0] \phi)$</td>
<td>def of subst ($x$, $y$ different)</td>
</tr>
<tr>
<td>4</td>
<td>$[y \Rightarrow y_0][x \Rightarrow x_0] \phi$</td>
<td>assumption $y_0$</td>
</tr>
<tr>
<td>5</td>
<td>$[x \Rightarrow x_0][y \Rightarrow y_0] \phi$</td>
<td>def of subst ($x$, $y$, $x_0$, $y_0$ different)</td>
</tr>
<tr>
<td>6</td>
<td>$\exists x[y \Rightarrow y_0] \phi$</td>
<td>$\exists x \ i \ 5$</td>
</tr>
<tr>
<td>7</td>
<td>$\exists y \exists x \phi$</td>
<td>$\exists y \ i \ 6$</td>
</tr>
<tr>
<td>8</td>
<td>$\exists y \exists x \phi$</td>
<td>$\exists y \ e \ 3, 4–7$</td>
</tr>
<tr>
<td>9</td>
<td>$\exists y \exists x \phi$</td>
<td>$\exists x \ e \ 1, 2–8$</td>
</tr>
</tbody>
</table>
Assume that $x$ is not free in $\psi$

<table>
<thead>
<tr>
<th>Formula</th>
<th>Equivalence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall x \phi \land \psi$</td>
<td>$\vdash \forall x (\phi \land \psi)$</td>
</tr>
<tr>
<td>$\forall x \phi \lor \psi$</td>
<td>$\vdash \forall x (\phi \lor \psi)$</td>
</tr>
<tr>
<td>$\exists x \phi \land \psi$</td>
<td>$\vdash \exists x (\phi \land \psi)$</td>
</tr>
<tr>
<td>$\exists x \phi \lor \psi$</td>
<td>$\vdash \exists x (\phi \lor \psi)$</td>
</tr>
</tbody>
</table>
Central Result of Natural Deduction

\[ \phi_1, \ldots, \phi_n \models \psi \quad \text{iff} \quad \phi_1, \ldots, \phi_n \vdash \psi \]

proven by Kurt Gödel, in 1929 in his doctoral dissertation
Central Result of Natural Deduction

\[ \phi_1, \ldots, \phi_n \models \psi \]

iff

\[ \phi_1, \ldots, \phi_n \vdash \psi \]

proven by Kurt Gödel, in 1929 in his doctoral dissertation (just one year before his most famous result, the incompleteness results of mathematical logic)