Logic & Discrete Math in Software Engineering (CAS 701)
Predicate Logic

Dr. Borzoo Bonakdarpour

Department of Computing and Software
McMaster University
Presentation outline

Predicate Logic

Syntax

Semantics

Proof systems
  Hilbert System
  Natural Deduction

Definability
Motivation

In propositional logic, only the logical forms of compound propositions are analyzed.

Propositional logic worked well with statements like *not, and, or, if ... then*.

We need some way to talk about *individuals* (also called *objects*) and refer to *some, all, among, and only* objects. Propositional logic fails to express such statements.
Consider this statement:

Every student is younger than some instructor.

This statement is about being a student, being an instructor, and being younger. These are all properties of some sort that we would like to be able to express along with logical connectives and dependencies.

Motivation
Motivation

More examples:

- For any natural number \( n \), there is a prime number greater than \( n \).
- \( 2^{100} \) is a natural number.
- There is a prime number greater than \( 2^{100} \).

First-order logic (also called predicate logic) gives us means to express and reason about objects.
Ingredients of FOL

FOL is a scientific theory with these ingredients:

- Domain of objects (individuals) (e.g., the set of natural numbers)
- Variables
- Designated individuals (e.g., ‘0’)
- Functions (e.g., ‘+’ and ‘.’)
- Relations (e.g., ‘=’)
- Quantifiers and Propositional connectives

Now, let’s explore the meaning of each of these.
We use **predicates** (i.e., relations) to express statements such as ‘being a student’.

For example, we could write $S(\text{andy})$ to denote that Andy is a student and $I(\text{paul})$ to denote that Paul is an instructor. Likewise, $Y(\text{andy}, \text{paul})$ could mean that Andy is younger than Paul.

In order to make predicates more expressive, we use **variables**. Think of variables as **place holders** that can be replaced by concrete objects.
FOL - Variables

For example:

\[ S(x) : x \text{ is a student} \]
\[ I(x) : x \text{ is an instructor} \]
\[ Y(x, y) : x \text{ is younger than } y \]

Notice that we can write the meaning of \( I \) by

\[ I(y) : y \text{ is an instructor} \]

or equivalently by writing

\[ I(z) : z \text{ is an instructor} \]
In general, we use variables that range over a domain of objects to make general statements:

\[ x^2 \geq 0 \]

and in expressing conditions which individuals may or may not satisfy:

\[ x + x = x \cdot x \]

This condition is satisfied only by 0 and 2.
We need to convey the meaning of ‘Every student $x$ is younger than some professor $y$’.

This is where use the terms “for all ” and “there exists” frequently (called quantifiers). For example:

- For all $\epsilon > 0$, there exists some $\delta > 0$ such that if $|x - a| < \delta$, then $|f(x) - b| < \epsilon$.

“For all” is called the universal quantifier ‘∀’ and “there exists” is the existential quantifier ‘∃’.
A quantifier is always attached to variables as in $\forall x$ (for all $x$) and $\exists z$ (there exists $z$).

We can now write our example entirely symbolically (although paraphrased!):

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

Or, the statement ‘Not all birds can fly’ can be written as:

$$\neg(\forall x. (B(x) \rightarrow F(x)))$$
In addition to predicates and quantifiers, FOL logic extends propositional logic by using **functions** as well. Consider the following statement:

Every child is younger than its mother.

One way to express this statement in FOL is the following:

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

But this means $x$ can have multiple mothers!
Functions in FOL gives us way to express statements more concisely. The previous example can be expressed as follows:

$$\forall x. (C(x) \rightarrow Y(x, m(x)))$$

where $m$ is a function: it takes one argument and returns the mother of that argument.
More examples:

- Andy and Paul have the same maternal grandmother:
  $$m(m(a)) = m(m(p))$$

- Ann likes Mary’s brother:
  $$\exists x. (B(x, m) \land L(a, x))$$
FOL - Evaluation

Consider:

For all \( x \), \( x \) is even.
There exists \( x \), such that \( x \) is even.

Since \( x \) ranges over \( \mathbb{N} \), they mean:

For all natural numbers \( x \), \( x \) is even.
There exists a natural number \( x \), such that \( x \) is even.

These have truth values!
Also,

‘4 is even’

is a proposition since 4 is an individual in \( \mathbb{N} \). If we replace 4 by a variable \( x \) ranging over \( \mathbb{N} \), then

‘\( x \) is even’

is not a proposition and has no truth value. It is a proposition function.

A **proposition function** on a domain \( D \) is an \( n \)-ary function mapping \( D^n \) into \( \{0, 1\} \).
Presentation outline

Predicate Logic

Syntax

Semantics

Proof systems
  Hilbert System
  Natural Deduction

Definability
1. Constant (individual) symbols (CS): \( c, d, c_1, c_2, \ldots, d_1, d_2 \ldots \)
2. Function Symbols (FS): \( f, g, h, f_1, f_2, \ldots, g_1, g_2 \)
3. Variables (VS): \( x, y, z, x_1, x_2, \ldots, y_1, y_2 \ldots \)
4. Predicate (Relational) Symbols (PS): \( P, Q, P_1, P_2, \ldots, Q_1, Q_2, \ldots \)
5. Logical Connectives: \( \neg, \land, \lor, \rightarrow \)
6. Quantifiers: \( \forall \) (for all) and \( \exists \) (there exists)
7. Punctuation: ‘(’, ‘)’, ‘.’, and ‘,’.
Example

- 0: constant ‘0’
- S: function (successor) $S(x)$ stands for: ‘$x + 1$’
- Eq: relation (equality) $Eq(x, y)$ stands for: ‘$x = y$’
- plus: function (addition) $\text{plus}(x, y)$ stands for: ‘$x + y$’

$$\forall x. Eq(\text{plus}(x, S(S(0))), S(S(x)))$$

means “Adding two to a number results in the second successor of that number”
Example

\[ \forall x. \forall y. \text{Eq}(\text{plus}(x, y), \text{plus}(y, x)) \]

means “Addition is commutative.”

\[ \neg \exists x. \text{Eq}(0, S(x)) \]

means “0 is not the successor of any number.”
The set $\text{Term}(\mathcal{L})$ of terms of $\mathcal{L}$ is defined using the following rules:

- All constants in $CS$ are terms
- All variables in $VS$ are terms
- If $t_1, \ldots, t_n \in \text{Term}(\mathcal{L})$ and $f$ is an $n$-ary function, then $f(t_1, \ldots, t_n) \in \text{Term}(\mathcal{L})$. 
Terms

**Example**

0, x, and y are terms and so are $S(0)$, $\text{plus}(x, y)$.

**Example**

Suppose $f$ is a unary and $g$ is a binary function, and, $a$ is a constant. Then, $g(f(a), a)$ and $f(g(a, f(a)))$ are terms, but $g(a)$ and $f(f(a), a)$ are not.
Atoms in FOL
Let $P$ be a predicate (i.e., an $n$-ary relation). An expression of $\mathcal{L}$ is an atom in $\text{Atoms}(\mathcal{L})$ iff it is of one of the forms $P(t_1, \ldots, t_n)$, where $t_1, \ldots, t_n$ are terms in $\text{Term}(\mathcal{L})$.

Formation Rules in FOL
We define the set $\text{Form}(\mathcal{L})$ of first-order logic formulas inductively as follows:

1. $\text{Atom}(\mathcal{L}) \subseteq \text{Form}(\mathcal{L})$
2. If $\varphi \in \text{Form}(\mathcal{L})$, then $(\neg \varphi) \in \text{Form}(\mathcal{L})$
3. If $\varphi, \psi \in \text{Form}(\mathcal{L})$, then $(\varphi * \psi) \in \text{Form}(\mathcal{L})$, where $* \in \{\land, \lor, \rightarrow\}$
4. If $\varphi \in \text{Form}(\mathcal{L})$ and $x \in \text{VS}$, then $(\forall x. \varphi) \in \text{Form}(\mathcal{L})$ and $(\exists x. \varphi) \in \text{Form}(\mathcal{L})$
Parse trees are similar to propositional formulas:

- Quantifiers $\forall x$ and $\exists y$ form nodes like negation (i.e., only one sub-tree).
- Predicates $P(t_1, t_2, \ldots, t_n)$ has $P$ as a node and terms $t_1, t_2, \ldots, t_n$ as children nodes.
\( (\forall x.((P(x) \rightarrow Q(x)) \land S(x, y))) \)
Example

How is the following formula generated?

$$\forall x. (F(b) \rightarrow \exists y. (\forall z. G(y, z) \lor H(u, x, y)))$$
To evaluate first-order formulas, we need to understand the nature of occurrence of variables. For example, in parse tree of Slide 25:

- three leaves labeled by $x$: if we walk up from these nodes, we reach a node labeled by $\forall x$
- one leaf labeled by $y$: if we walk up from this nodes, we will reach no quantifiers for $y$. 
Bound and Quantified Variables

We say that an occurrence of \( x \) is **free** in first-order formula \( \varphi \), if in the parse tree of \( \varphi \), there is no upwards path from \( x \) to a node labeled by \( \forall x \) or \( \exists x \).

An occurrence of \( x \) that is not free is called **bound** (or **quantified**).
Let $\varphi \in \text{Form}(\mathcal{L})$. We define the set $\text{FV}(A)$ of free variables of $A$ as follows:

1. $\{x \mid x \text{ appears in } t_i \text{ for some } 0 < i \leq \text{ar}(P)\}$, for $\varphi = P(t_1, \ldots, t_{\text{ar}(P)})$
2. $\text{FV}(\varphi)$ for $\psi = (\neg \varphi)$
3. $\text{FV}(\varphi) \cup \text{FV}(\psi)$ for $\gamma = (\varphi \ast \psi)$, where $\ast \in \{\land, \lor, \rightarrow\}$
4. $\text{FV}(\varphi) \setminus \{x\}$ for $\psi = (\forall x. \varphi)$ or $\psi = (\exists x. \varphi)$

Variables not in $\text{FV}(\varphi)$ are bound variables.
If $\forall x. A(x)$ or $\exists x. A(x)$ is a segment of $B$, $A(x)$ is called the **scope** in $B$ of the $\forall x$ or $\exists x$ on the left of $A(x)$.

In the following formula:

$$\exists x. \forall y. \exists z. F(x, y, z)$$

what is the scope of $\forall y$?
Is $x$ free or quantified?
### Closed Formula

A first-order formula \( A \in Form(\mathcal{L}) \) is **closed** (also called a **sentence**) if \( FV(A) = {} \).
Substitution

Given a variable \( x \), a term \( t \), and a formula, \( \varphi \), we define

\[ \varphi[x/t] \]

to be the formula obtained by replacing each free occurrence of variable \( x \) in \( \varphi \) with \( t \).

Example

Consider formula

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

We have

\[ \varphi[x/f(x, y)] = \varphi \]

because there is no free occurrence of \( x \).
Substitution

Example

Consider formula

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

We have

$$\varphi[x/f(x, y)] = (\forall x.((P(x) \land Q(x))) \rightarrow (\neg P(f(x, y)) \lor Q(y)))$$

We say that term $t$ is free for variable $x$ in formula $\varphi$, if no free occurrence of $x$ in $\varphi$ is in the scope of $\forall y$ or $\exists y$ for any variable $y$ occurring in $t$. 
Consider term $t = f(y, y)$ and formula

$$\varphi = S(x) \land (\forall y. (P(x) \rightarrow Q(y)))$$

The leftmost $x$ can be substituted by $t$ since it is not in the scope of any quantifier, but substituting the rightmost $x$ introduces a new variable $y$ in $t$, which becomes bound by $\forall y$. Hence, $f(y, y)$ is not free for $x$ in $\varphi$.

Such cases can be resolved by variable renaming, e.g., $t = f(z, z)$. 
Substitution - Formal Description

Substitution

1. For a term $t_1$, $(t_1)[x/t]$ is $t_1$ with each occurrence of the variable $x$ replaced by the term $t$.

2. For $\varphi = P(t_1, \ldots, t_{ar(P)})$, $(\varphi)[x/t] = P((t_1)[x/t], \ldots, (t_{ar(P)})[x/t])$.

3. For $\varphi = (\neg \psi)$, $(\varphi)[x/t] = (\neg (\psi)[x/t])$;

4. For $\varphi = (\psi \rightarrow \eta)$, $(\varphi)[x/t] = ((\psi)[x/t] \rightarrow (\eta)[x/t])$, and

5. for $\varphi = (\forall y. \psi)$, there are two cases:
   - if $x$ is $y$, then $(\varphi)[x/t] = \varphi = (\forall y. \psi)$, and
   - otherwise, then $(\varphi)[x/t] = (\forall z. (\psi[y/z])[x/t])$, where $z$ is any variable that is not free in $t$ or in $\varphi$.

In the last case above, the additional substitution $(\_)[y/z]$ (i.e., renaming the variable $y$ to $z$ in $\psi$) is needed in order to avoid an accidental capture of a variable by the quantifier (i.e., capture of any $y$ that is possibly free in $t$).
Presentation outline

Predicate Logic

Syntax

Semantics

Proof systems
  Hilbert System
  Natural Deduction

Definability
In propositional logic, semantics was described in terms of valuation of (the only ingredients) propositional variables.

FOL includes more ingredients (i.e., predicates and functions) and, hence, the interpretations for it are more complicated.

FO formulas are intended to express propositions (i.e, true/false valuation). This is accomplished by interpretations (also called models).
Interpretation

A first order interpretation \( I \) is a tuple \((D, (.)^I)\):

- \( D \) is a non-empty set called the domain (or universe); and
- \( (.)^I \) is an interpretation function that maps
  - constant symbols \( c \in CS \) to individuals \( c^I \in D \);
  - function symbols \( f \in FS \) to functions \( f^I : D^{ar(f)} \to D \); and
  - predicate symbols \( P \in PS \) to relations \( P^I \subseteq D^{ar(P)} \).

Example

Let functions \( f \) and \( g \) are respectively addition and squaring functions and \( P \) be the equality relation. Let

\[ P(f(g(a), g(b)), g(c)) \]

be a closed formula, where individuals \( a, b, \) and \( c \) be interpreted as 4, 5, and 6 in \( \mathbb{N} \). Then, the above predicate is interpreted as the false proposition (why?).
Example

Let

\[ f(g(a), f(b, c)) \]

be a term. Let individuals \( a, b, \) and \( c \) be interpreted as 4, 5, and 6 in \( \mathbb{N} \) and functions \( f \) and \( g \) are respectively as addition and squaring. Then, the above term is interpreted as

\[ 4^2 + (5 + 6) \]

which is the individual 27 in \( \mathbb{N} \).
Interpretation is extremely liberal and open ended.

Example
Consider the non-closed formula:

\[ P(f(g(u), g(b)), g(w)) \]

where only \( b \) is interpreted as 5. One can interpret this formula by:

\[ x^2 + 5^2 = y^2 \]

where \( x \) and \( y \) are free variables. This is not a proposition, but a binary proposition function in \( \mathbb{N} \).
Given an interpretation, in order to evaluate the truthfulness of a formula $\forall x. \varphi$ or $\exists x. \varphi$, we should check whether $\varphi$ holds for all or some value $a$ in the interpretation.

The mechanism to check this is by using substitution $\varphi[x/a]$ for values $a$ in an interpretation. This is called a valuation.

For example, in the previous frame, one can obtain a truth value by assigning individuals in $\mathbb{N}$ to $x$ and $y$. 
Valuation

A valuation $\theta$ (also called an assignment) is a mapping from $VS$, the set of variables, to domain $D$.

Example

For the non-closed formula

$$x^2 + 4^2 = y^2$$

$\theta(x) = 3$ and $\theta(y) = 5$ evaluates the formula to the true proposition.
Meaning of Terms

Term Interpretation/Valuation

Let $I$ be a first order interpretation and $\theta$ a valuation. For a term $t$ in $\text{Term}(\mathcal{L})$, we define interpretation and valuation of $t$, $t^I,\theta$, as follows:

1. $c^I,\theta = c^I$ for $t \in CS$ (i.e., $t$ is a constant);
2. $x^I,\theta = \theta(x)$ for $t \in VS$ (i.e., $t$ is a variable); and
3. $f(t_1, \ldots, t_{\text{ar}(f)})^I,\theta = f^I((t_1)^I,\theta, \ldots, (t_{\text{ar}(f)})^I,\theta)$, otherwise (i.e., for $t$ a functional term).
Meaning of Terms

Example

Suppose a language that

- has constant symbol 0,
- a unary function \( s \), and
- a binary function \( + \)

Let us write + in infix position, (i.e., \( x + y \) instead of \( +(x, y) \)).

Notice that \( s(s(0) + s(x)) \) and \( s(x + s(x + s(0))) \) are two terms.

The following are examples of interpretations and valuations:

- \( D = \{0, 1, 2, \ldots \} \), \( 0' = 0 \), \( s' \) is the successor function and \( +' \) is the addition operation. Then, if \( \theta(x) = 3 \), \( s(s(0) + s(x)) = 6 \) and \( s(x + s(x + s(0))) = 9 \).
Meaning of Terms

Example

- \( D \) is the collection of all words over the alphabet \( \{ a, b \} \), \( 0' = a \), \( s' \) is the operation that appends \( a \) to the end of the word, and \( +' \) is concatenation. Then, if \( \theta(x) = aba \), \( s(s(0) + s(x)) = aaabaaa \) and \( s(x + s(x + s(0))) = abaabaaaaa \).

- \( D = \{ \ldots, -2, -1, 0, 1, 2, \ldots \} \), \( 0' = 1 \), \( s' \) is the predecessor function and \( +' \) is the subtraction operation. Then, in general, \( s(s(0) + s(x)) = -\theta(x) \) and \( s(x + s(x + s(0))) = 0 \), given any valuation \( \theta \).
Satisfaction Relation in FOL

Satisfaction Relation

The satisfaction relation $\models$ between an interpretation $I$, a valuation $\theta$, and a first-order formula $\varphi$ is defined as:

- $I, \theta \models P(t_1, \ldots, t_{\text{ar}(P)})$ iff $\langle (t_1)^I, \theta, \ldots, (t_{\text{ar}(P)})^I, \theta \rangle \in P^I$ for $P \in PS$
- $I, \theta \models \neg \varphi$ if and only if $I, \theta \not\models \varphi$ is not true
- $I, \theta \models \varphi \land \psi$ if and only if $I, \theta \models \varphi$ and $I, \theta \models \psi$
- $I, \theta \models \varphi \lor \psi$ if and only if $I, \theta \models \varphi$ or $I, \theta \models \psi$
- $I, \theta \models (\forall x.\varphi)$ if and only if $I, \theta([x = v]) \models \varphi$ for all $v \in D$
- $I, \theta \models (\exists x.\varphi)$ if and only if $I, \theta([x = v]) \models \varphi$ for some $v \in D$

where the valuation $[x = v](y)$ is defined to be $v$ when $x = y$ and $\theta$ otherwise.
Example

Let loves be a binary predicate. Consider the following formula:

$$\forall x. \forall y. (\text{loves}(x, \text{alma}) \land \text{loves}(y, x) \rightarrow \neg\text{loves}(y, \text{alma}))$$

Let interpretation $I$ be the following: $D = \{a, b, c\}$, 
loves$^I = \{(a, a), (b, a), (c, a)\}$, $CS = \{\text{alma}\}$, alma$^I = a$. The above formulas intends to capture

*None of Alma's lovers' lovers love her.*

This is not the case! Why?
Suppose $R$ is a binary relation and $\oplus$ is a binary function:

- Consider the sentence $\exists y. R(x, y \oplus y)$. Suppose $D = \{1, 2, 3, \ldots\}$, $\oplus'$ is the addition operation, and $R'$ is the equality relation. Then, $I, \theta \models \exists y. R(x, y \oplus y)$ iff $\theta(x)$ is an even number.
The universal and existential quantifiers may be interpreted respectively as generalization of conjunction and disjunction. If the domain $D = \{\alpha_1, \ldots, \alpha_k\}$ is finite then:

For all $x$ st. $f(x)$ iff $R(\alpha_1)$ and ... and $R(\alpha_k)$

There exists $x$ st. $R(x)$ iff $R(\alpha_1)$ or ... or $R(\alpha_k)$

where $R$ is a property.
Relevance Lemma

Let $\varphi$ be a first-order formula, $I$ be an interpretation, and $\theta_1$ and $\theta_2$ be two valuations such that $\theta_1(x) = \theta_2(x)$ for all $x \in VS$. Then,

$$I, \theta_1 \models \varphi \text{ iff } I, \theta_2 \models \varphi$$

Proof by structural induction.
Satisfiability and Validity

**Satisfiability**

$\Sigma \subseteq Form(\mathcal{L})$ is **satisfiable** iff there is some interpretation $I$ and valuation $\theta$, such that $I, \theta \models \varphi$ for all $\varphi \in \Sigma$.

**Validity**

A formula $\varphi \in Form(\mathcal{L})$ is **valid** iff for all interpretations $I$ and valuations $\theta$, we have $I, \theta \models \varphi$.
Example

Let $\varphi = P(f(g(x), g(y)), g(z))$ be a formula. The formula is satisfiable:

- $f^I = \text{summation}$
- $g^I = \text{squaring}$
- $P^I = \text{equality}$
- $\theta(x) = 3, \theta(y) = 4, \theta(z) = 5$

$\varphi$ is not valid. (why?)
Suppose $\Sigma \subseteq \text{Form}(\mathcal{L})$ and $\varphi \in \text{Form}(\mathcal{L})$. We say that $\varphi$ is a logical consequence of $\Sigma$ (that is, of the formulas in $\Sigma$), written as $\Sigma \models \varphi$, iff for any interpretation $I$ and valuation $\theta$, we have $I, \theta \models \Sigma$ implies $I, \theta \models \varphi$.

$\models \varphi$ means that $\varphi$ is valid.
Example

Show that \( \models \forall x. (\varphi \rightarrow \psi) \rightarrow ((\forall x. \varphi) \rightarrow (\forall x. \psi)) \)

Proof by contradiction: there exists \( I \) and \( \theta \) st.

\( I, \theta \not\models \forall x. (\varphi \rightarrow \psi) \rightarrow ((\forall x. \varphi) \rightarrow (\forall x. \psi)) \)

\( I, \theta \models \forall x. (\varphi \rightarrow \psi) \)

\( I, \theta \models \forall x. \varphi \)

\( I, \theta \not\models \forall x. \psi \)

\( I, \theta ([x = v]) \models \varphi \)

\( I, \theta ([x = v]) \not\models \psi \)

\( I, \theta ([x = v]) \not\models \varphi \rightarrow \psi \)

\( I, \theta \not\models \forall x. (\varphi \rightarrow \psi) \) (contradiction)
Example

Show that \( \forall x. \neg A(x) \models \neg \exists x. A(x) \)

Proof by contradiction: there exists \( I \) and \( \theta \) st.

\[
\begin{align*}
I, \theta &\models \forall x. \neg A(x) \text{ and } I, \theta \not\models \neg \exists x. A(x) \\
I, \theta &\models \exists x. A(x)
\end{align*}
\]

\[
\begin{align*}
I, \theta([x = v]) &\models \neg A(x) \text{ for all } v \\
I, \theta([x = v]) &\models A(x) \text{ for some } v
\end{align*}
\]

Contradiction!
Show that \((\forall x. \varphi) \rightarrow (\forall x. \psi)) \neq \forall x. (\varphi \rightarrow \psi)\)
Replacability and Duality

**Theorem.** If $B \equiv C$ and $A'$ results from $A$ by replacing some (not necessarily all) occurrences of $B$ in $A$ by $C$, then $A \equiv A'$.

**Theorem.** Suppose $A$ is a formula composed of atoms and the connectives $\neg$, $\land$, and $\lor$ by the formation rules concerned, and $A'$ results by exchanging in $A$, $\land$ for $\lor$ and each atom for its negation. Then $A' \equiv \neg A$. ($A'$ is the dual of $A$)
Substitution Lemma

\[ \models \forall x. \varphi \rightarrow \varphi[x/t] \]

\[ I, \theta \models \varphi[x/t] \text{ iff } I, \theta[x = (t)^{I,\theta}] \models \varphi \]
Presentation outline

Predicate Logic

Syntax

Semantics

Proof systems
  Hilbert System
  Natural Deduction

Definability
Proof calculi for predicate logic are similar to those for propositional logic, except that we have new proof rules for dealing with the quantifiers.

Again, we will explore:

- Hilbert systems, and
- Natural deduction
Outline

Predicate Logic

Syntax

Semantics

Proof systems

Hilbert System

Natural Deduction

Definability
FOL Hilbert System

### Axioms

Ax1 \( \langle \forall^*(\varphi \rightarrow (\psi \rightarrow \varphi)) \rangle \);
Ax2 \( \langle \forall^*((\varphi \rightarrow (\psi \rightarrow \eta)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \eta))) \rangle \);
Ax3 \( \langle \forall^*((\neg \varphi) \rightarrow (\neg \psi)) \rightarrow (\psi \rightarrow \varphi) \rangle \);
Ax4 \( \langle \forall^*(\forall x. (\varphi \rightarrow \psi)) \rightarrow ((\forall x. \varphi) \rightarrow (\forall x. \psi)) \rangle \);
Ax5 \( \langle \forall^*(\forall x. \varphi) \rightarrow \varphi[x/t] \rangle \) for \( t \in T \) a term;
Ax6 \( \langle \forall^*(\varphi \rightarrow \forall x. \varphi) \rangle \) for \( x \notin \text{FV}(\varphi) \); and
MP \( \langle \varphi, (\varphi \rightarrow \psi), \psi \rangle \).

where \( \forall^* \) is a finite sequence of universal quantifiers (e.g., \( \forall x_1. \forall y. \forall x \)).
Generalization Theorem

Show that if $\Phi \vdash \varphi$ and $x \notin \text{FV}(\Phi)$, then $\Phi \vdash \forall x. \varphi$.

Proof by structural induction.

**Base case:** $\varphi$ is an axiom. Then, $\Phi \vdash \forall x. \varphi$.

**Induction step (1):** $\varphi \in \Phi$

$\Phi \vdash \varphi$

$\vdash \varphi \rightarrow \forall x. \varphi$

$\Phi \vdash \forall x. \varphi$ (MP and $x \notin \text{FV}(\varphi)$)
**Generalization of Axioms (why \( \forall^* \))**

---

**Induction step (2):** \( \psi \rightarrow \varphi \)

1. \( \Phi \vdash \psi \), \( \Phi \vdash \psi \rightarrow \varphi \)  
   (Induction hyp.)
2. \( \Phi \vdash (\forall x.\psi) \)  
   (Ax₆)
3. \( \Phi \vdash \forall x.(\psi \rightarrow \varphi) \)  
   (Ax₆)
4. \( \Phi \vdash (\forall x.\psi) \rightarrow \forall x.\varphi \)  
   (Ax₅)
5. \( \Phi \vdash (\forall x.\varphi) \)  
   (MP)
Example

Show that $\vdash \forall x. \forall y. \varphi \to \forall y. \forall x. \varphi$

1. $\forall x. \forall y. \varphi$  
   (Deduction theorem)
2. $\forall x. \forall y. \varphi \to (\forall y. \varphi)[x/t]$  
   ($Ax_5$)
3. $(\forall y. \varphi)[x/t]$  
   (MP)
4. $(\forall y. \varphi)[x/t] \to ((\varphi)[x/t])[y/t']$  
   ($Ax_5$)
5. $((\varphi)[x/t])[y/t']$  
   (MP)
6. $((\varphi)[x/t])[y/t'] \to \forall x. (\varphi)[x/t]$  
   ($Ax_6$)
7. $\forall x. (\varphi)[x/t]$  
   (MP)
8. $\forall x. (\varphi)[x/t] \to \forall y. \forall x. \varphi$  
   ($Ax_6$)
9. $\forall y. \forall x. \varphi$  
   (MP)
FOL Hilbert System

Example

Show that ⊢ ∀x.(A(x) → B(x)) → (∀x.A(x) → ∀x.B(x))

1. ∀x.(A(x) → B(x))  
   (Assumption)
2. ∀x.A(x)  
   (Assumption)
3. ∀x.A(x) → A(a)  
   (Ax_5)
4. A(a)  
   (MP 2, 3)
5. ∀x.(A(x) → B(x)) → (A(a) → B(a))  
   (Ax_5)
6. A(a) → B(a)  
   (MP 1, 5)
7. B(a)  
   (MP 4, 6)
8. B(a) → ∀x.B(x)  
   (Ax_6)
Outline

Predicate Logic

Syntax

Semantics

Proof systems
  Hilbert System
  Natural Deduction

Definability
Natural deduction in FOL is similar to propositional logic except we need to introduce rules for quantifier elimination and introduction.

Other proof techniques and tricks remain the same as natural deduction for propositional logic.
### Universal Quantifier

<table>
<thead>
<tr>
<th>Name</th>
<th>( \vdash ) notation</th>
<th>inference notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \forall )-elimination</td>
<td>If ( \Sigma \vdash \forall x. \varphi )</td>
<td>( \forall x. \varphi )</td>
</tr>
<tr>
<td>(( \forall - ))</td>
<td>then ( \Sigma \vdash \varphi[x/t] )</td>
<td>( \varphi[x/u] )</td>
</tr>
<tr>
<td>( \forall )-introduction</td>
<td>If ( \Sigma \vdash \varphi[x/u] )</td>
<td>( \varphi[x/t] )</td>
</tr>
<tr>
<td>(( \forall + ))</td>
<td>then ( \Sigma \vdash \forall x. \varphi )</td>
<td>( \forall x. \varphi )</td>
</tr>
</tbody>
</table>

In (\( \forall - \)), the formula \( \varphi[x/t] \) is obtained by substituting \( t \) for all occurrences of \( x \). In (\( \forall + \)), \( u \) should not occur in \( \Sigma \).
Universal Quantifier

The rule \((\forall +)\) is a bit tricky. Think of it this way:

\[
\text{if you want me to prove that } \forall x. \phi, \text{ then I show the truthfulness of } \phi \\
\text{for any ‘random’ } x \text{ you give me.}
\]

In other words, if we prove \(\varphi\) about any \(u\) that is not special in any way, then you can prove it for any \(x\) whatsoever. That is, the step from \(\varphi\) to \(\forall x. \varphi\) is legitimate if only we have arrived at \(\varphi\) in such a way that none of its assumptions contain \(x\) as a free variable.
Generalization

Rules of elimination and introduction in FOL natural deduction can be generalized to multiple quantifiers:

- If $\Sigma \vdash \forall x_1 \ldots x_n. \varphi$ then $\Sigma \vdash \varphi[x_1/t_1 \ldots x_n/t_n]$
- If $\Sigma \vdash \varphi[x_1/u_1 \ldots x_n/u_n]$ then $\Sigma \vdash \forall x_1 \ldots x_n. \varphi$, where $u_1 \ldots u_n$ do not occur in $\Sigma$. 
Example

Show that $\forall x.\forall y. A(x, y) \vdash \forall y.\forall x. A(x, y)$

1. $\forall xy. \varphi(x, y) \vdash \varphi(u, v)$
2. $\forall xy. \varphi(x, y) \vdash \forall yx. \varphi(x, y)$ (Generalized $\forall+$)

In Step 1, $u$ and $v$ should not occur in $\varphi(x, y)$
Example

Show that $P(t), \forall x.(P(x) \rightarrow \neg Q(x)) \vdash \neg Q(t)$

1. $\forall x.(P(x) \rightarrow \neg Q(x)) \vdash P(t) \rightarrow \neg Q(t)$
2. $P(t)$ \hspace{2cm} premise
3. $\neg Q(t)$ \hspace{2cm} $\rightarrow$
### Existential Quantifier

<table>
<thead>
<tr>
<th>Name</th>
<th>$\vdash$ notation</th>
<th>inference notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\exists$-elimination ($\exists -$)</td>
<td>If $\Sigma, \varphi(u) \vdash \psi$ then $\Sigma, \exists x . \varphi(x) \vdash \psi$</td>
<td>$\begin{array}{c} \psi \ \exists x . \varphi(x) \ \varphi(u) \end{array}$</td>
</tr>
</tbody>
</table>

In ($\exists -$), $u$ should not occur in $\Sigma$ or $\psi$. 
### Existential Quantifier

<table>
<thead>
<tr>
<th>Name</th>
<th>$\vdash$ notation</th>
<th>Inference notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\exists$-introduction</td>
<td>If $\Sigma \vdash \varphi[x/t]$</td>
<td>$\exists x.\varphi$</td>
</tr>
<tr>
<td>$(\exists +)$</td>
<td>then $\Sigma \vdash \exists x.\varphi$</td>
<td>$\varphi[x/t]$</td>
</tr>
</tbody>
</table>

In $(\exists +)$, $\varphi(x)$ is obtained by replacing some occurrences of $t$ in $\varphi$ by $x$. In the $(\exists +)$ rule notice that $\varphi[x/t]$ has more information that $\exists x.\varphi$. For example, let $t = y$ such that $\varphi[x/t]$ is $y = y$. Then, $\phi$ could be a number of things, such as $x = x$ or $x = y$. 
Show that $\exists x. \varphi(x) \vdash \exists y. \varphi(y)$

1. $\varphi(u) \vdash \varphi(u)$ (\(u\) not occurring in $A(y)$)
2. $\varphi(u) \vdash \exists y. \varphi(y)$ (\(\exists +\))
3. $\exists x. \varphi(x) \vdash \exists y. \varphi(y)$ (\(\exists -\))
Example

Show that $\neg \forall x. \varphi(x) \vdash \exists x. \neg \varphi(x)$

1. $\neg \varphi(u) \vdash \exists x. \neg \varphi(x)$ (u not occurring in $A(x)$)
2. $\neg \exists x. \neg \varphi(x) \vdash \varphi(u)$ (1)
3. $\neg \exists x. \neg \varphi(x) \vdash \forall x. \varphi(x)$
4. $\neg \forall x. \varphi(x) \vdash \exists x. \neg \varphi(x)$
Predicate Logic

Syntax

Semantics

Proof systems
   Hilbert System
   Natural Deduction

Definability
Definability

Let $I = (D, (.))^I$ be a first-order interpretation and $\varphi$ a first-order formula. A set $S$ of $k$-tuples over $D$, $S \subseteq D^k$, is defined by the formula $\varphi$ if

$$S = \{ (\theta(x_1), \ldots, \theta(x_k)) \mid I, \theta \models \varphi \}$$

A set $S$ is definable in first-order logic if it is defined by some first-order formula $\varphi$.

Let $\Sigma$ be a set of first-order sentences and $\mathcal{K}$ a set of interpretations. We say that $\Sigma$ defines $\mathcal{K}$ if

$$I \in \mathcal{K} \text{ if and only if } I \models \Sigma.$$ 

A set $\mathcal{K}$ is (strongly) definable if it is defined by a (finite) set of first-order formulas $\Sigma$. 
Compactness in FOL

Compactness Theorem

\[ \Sigma \subseteq \text{Form}(\mathcal{L}) \text{ is satisfiable iff every finite subset of } \Sigma \text{ is satisfiable.} \]
Graphs

Graph

An undirected graph is a tuple \((V, E)\), where \(V\) is a set of vertices and \(E\) is a set of edges. An edge is a pair \((v_1, v_2)\), where \(v_1, v_2 \in V\).

\[ V = \{v_1, v_2, v_3, v_4, v_5\} \]
\[ E = \{(v_1, v_2), (v_2, v_3), (v_2, v_4), (v_1, v_4), (v_1, v_5)\} \]

Adjacency

If \((v_1, v_2) \in E\), we say that \(v_1\) is adjacent to \(v_2\).

Adjacency in a graph can be expressed by a binary relation. Thus, relation \(E(v_1, v_2)\) is interpreted as “\(v_1\) is adjacent to \(v_2\)”. A graph is any model of the following 2 axioms:

1. \(\forall x. \forall y. E(x, y) \rightarrow E(y, x)\) (“if \(x\) is adjacent to \(y\), then \(y\) is adjacent to \(x\)”)  
2. \(\forall x. \neg E(x, x)\) (“no \(x\) is adjacent to itself”)
Graphs in FOL

We can express many properties of a graph in the language of first-order logic.

Example
For instance, the property “G contains a triangle” is the following formula:

$$\exists x. \exists y. \exists z. (E(x, y) \land E(y, z) \land E(z, x))$$

Example
Define first-order formulas for :

- A graph has girth of size 4
- A graph is 3-colorable
Graph Connectivity in FOL

We cannot express graph connectivity in FOL (i.e., graph connectivity is not definable in FOL).

Proof

Let predicate $C$ express “$G$ is a connected graph”. We add constants $s$ and $t$ vertices.

For any $k$, let $L_k$ be the proposition “there is no path of length $k$ between $s$ and $t$”. For example,

$$L_3 = \neg\exists x.\exists y. (E(s, x) \land E(x, y) \land E(y, t))$$

Now consider the set of formulas

$$\Sigma = \{\text{axiom}(1), \text{axiom}(2), C, L_1, L_2, \ldots \}$$

$\Sigma$ is finitely satisfiable: there do exist connected graphs with $s$ and $t$, that are connected by an arbitrarily long path. This is because any finite subset $F \subset \Sigma$ must have bounded $k$’s, such a graph satisfies $F$. 
Proof (cont’d)

- By the compactness theorem, $\Sigma$ is satisfiable; i.e., there exists some model $G$ of all formulas $\Sigma$, which is a graph that cannot be connected by a path of length $k$, for any $k$, for all $k$.
- This is clearly wrong. In a connected graph, any 2 nodes are connected by a path of finite length!