The Calculus of Computation

Zohar Manna

Spring 2009

Chapter 2: First-Order Logic (FOL)

First-Order Logic (FOL)

Also called Predicate Logic or Predicate Calculus

FOL Syntax

variables	$x, y, z, \cdots$
constants	$a, b, c, \cdots$
functions	$f, g, h, \cdots$
terms	variables, constants or n-ary function applied to n terms as arguments $a,\ x,\ f(a),\ g(x, b),\ f(g(x, f(b)))$; ~~$f(g(x, f(b, y)))$~~ ??
predicates	$p, q, r, \cdots$
atom	$\top$, $\bot$, or an n-ary predicate applied to n terms
literal	atom or its negation $p(f(x), g(x, f(x))), \quad \neg p(f(x), g(x, f(x)))$

Note: 0-ary functions: constants
0-ary predicates (propositional variables): $P, Q, R, \ldots$

quantifiers

existential quantifier $\exists x.\ F[x]$
"there exists an x such that $F[x]$"
Note: the dot notation ($\exists x., \forall x.$) means the scope of the quantifier should extend as far as possible.

universal quantifier $\forall x.\ F[x]$
"for all x, $F[x]$"

FOL formula

literal,
application of logical connectives ($\neg, \vee, \wedge, \rightarrow, \leftrightarrow$) to formulae,
or application of a quantifier to a formula

Example: FOL formula

$$\forall x.\ \underbrace{p(f(x), x) \ \rightarrow\ (\exists y.\ \underbrace{p(f(g(x, y)), g(x, y))}_{G}) \wedge q(x, f(x))}_{F}$$

The scope of $\forall x$ is F.
The scope of $\exists y$ is G.
The formula reads:
"for all x,
if $p(f(x), x)$
then there exists a y such that $p(f(g(x, y)), g(x, y))$
and $q(x, f(x))$"

FOL Semantics

An interpretation $I : (D_I, \alpha_I)$ consists of:

- Domain D_I
  non-empty set of values or objects
  cardinality $|D_I|$ deck of cards (finite)
  integers (countably infinite)
  reals (uncountably infinite)
- Assignment α_I
  - each variable x assigned value $x_I \in D_I$
  - each n-ary function f assigned

$$f_I : \ D_I^n \to D_I$$

  In particular, each constant a (0-ary function) assigned value $a_I \in D_I$
  - each n-ary predicate p assigned

$$p_I : \ D_I^n \to \{\text{true}, \ \text{false}\}$$

  In particular, each propositional variable P (0-ary predicate) assigned truth value (true, false)

Example: $F : \ p(f(x, y), z) \ \to \ p(y, g(z, x))$

Interpretation $I : (D_I, \alpha_I)$ with

$$D_I = \mathbb{Z} = \{\cdots, -2, -1, 0, 1, 2, \cdots\}$$

$$\alpha_I : \left\{ \begin{array}{l} f \mapsto +, \ g \mapsto -, \ p \mapsto >, \\ x \mapsto 13, \ y \mapsto 42, \ z \mapsto 1 \end{array} \right\}$$

Therefore, we can write

$$F_I : 13 + 42 > 1 \ \to \ 42 > 1 - 13.$$

F is true under I.

Semantics: Quantifiers

An x-variant of interpretation $I : (D_I, \alpha_I)$ is an interpretation $J : (D_J, \alpha_J)$ such that

- $D_I = D_J$
- $\alpha_I[y] = \alpha_J[y]$ for all symbols y, except possibly x

That is, I and J agree on everything except possibly the value of x.

Denote by $J : \ I \triangleleft \{x \mapsto \mathrm{v}\}$ the x-variant of I in which $\alpha_J[x] = \mathrm{v}$ for some $\mathrm{v} \in D_I$. Then

- $I \ \models \ \forall x. \ F$ iff for all $\mathrm{v} \in D_I$, $I \triangleleft \{x \mapsto \mathrm{v}\} \ \models \ F$
- $I \ \models \ \exists x. \ F$ iff there exists $\mathrm{v} \in D_I$, s.t. $I \triangleleft \{x \mapsto \mathrm{v}\} \ \models \ F$

Example: Consider

$$F : \ \exists x. \ f(x) = g(x)$$

and the interpretation

$$I : (D : \{\circ, \bullet\}, \alpha_I)$$

in which

$$\alpha_I : \ \{f(\circ) \mapsto \circ, f(\bullet) \mapsto \bullet, g(\circ) \mapsto \bullet, g(\bullet) \mapsto \circ\}.$$

The truth value of F under I is false; i.e., $I[F] = \text{false}$.

Satisfiability and Validity I

F is satisfiable iff there exists I s.t. $I \models F$
F is valid iff for all I, $I \models F$

$$F \text{ is valid iff } \neg F \text{ is unsatisfiable}$$

Semantic rules: given an interpretation I with domain D_I,

$$\frac{I \models \forall x.\ F[x]}{I \triangleleft \{x \mapsto \mathsf{v}\} \models F[x]} \quad \text{for any } v \in D_I$$

$$\frac{I \not\models \forall x.\ F[x]}{I \triangleleft \{x \mapsto \mathsf{v}\} \not\models F[x]} \quad \text{for a } \underline{\text{fresh}}\ v \in D_I$$

$$\frac{I \models \exists x.\ F[x]}{I \triangleleft \{x \mapsto \mathsf{v}\} \models F[x]} \quad \text{for a } \underline{\text{fresh}}\ v \in D_I$$

$$\frac{I \not\models \exists x.\ F[x]}{I \triangleleft \{x \mapsto \mathsf{v}\} \not\models F[x]} \quad \text{for any } v \in D_I$$

Contradiction rule

A contradiction exists if two variants of the original interpretation I disagree on the truth value of an n-ary predicate p for a given tuple of domain values:

$$\frac{\begin{array}{l} J : I \triangleleft \cdots \models p(s_1, \ldots, s_n) \\ K : I \triangleleft \cdots \not\models p(t_1, \ldots, t_n) \end{array}}{I \models \bot} \quad \text{for } i \in \{1, \ldots, n\}, \alpha_J[s_i] = \alpha_K[t_i]$$

Intuition: The variants J and K are constructed only through the rules for quantification. Hence, the truth value of p on the given tuple of domain values is already established by I. Therefore, the disagreement between J and K on the truth value of p indicates a problem with I.

Example: Is

$$F : (\forall x.\ p(x)) \leftrightarrow (\neg\exists x.\ \neg p(x))$$

valid?

Suppose not. Then there is an I such that $I \not\models F$ (assumption).

First case:

1a.	I	$\not\models$	$(\forall x.\ p(x)) \rightarrow (\neg\exists x.\ \neg p(x))$	assumption and $\leftrightarrow$
2a.	I	$\models$	$\forall x.\ p(x)$	1a and $\rightarrow$
3a.	I	$\not\models$	$\neg\exists x.\ \neg p(x)$	1a and $\rightarrow$
4a.	I	$\models$	$\exists x.\ \neg p(x)$	3a and $\neg$
5a.	$I \triangleleft \{x \mapsto \mathsf{v}\}$	$\models$	$\neg p(x)$	4a and $\exists$, v$\in D_I$ fresh
6a.	$I \triangleleft \{x \mapsto \mathsf{v}\}$	$\not\models$	$p(x)$	5a and $\neg$
7a.	$I \triangleleft \{x \mapsto \mathsf{v}\}$	$\models$	$p(x)$	2a and $\forall$

6a and 7a are contradictory.

Example (continued):

Second case:

1b.	I	$\not\models$	$(\neg\exists x.\ \neg p(x)) \rightarrow (\forall x.\ p(x))$	assumption and $\leftrightarrow$
2b.	I	$\not\models$	$\forall x.\ p(x)$	1b and $\rightarrow$
3b.	I	$\models$	$\neg\exists x.\ \neg p(x)$	1b and $\rightarrow$
4b.	$I \triangleleft \{x \mapsto \mathsf{v}\}$	$\not\models$	$p(x)$	2b and $\forall$, v$\in D_I$ fresh
5b.	I	$\not\models$	$\exists x.\ \neg p(x)$	3b and $\neg$
6b.	$I \triangleleft \{x \mapsto \mathsf{v}\}$	$\not\models$	$\neg p(x)$	5b and $\exists$
7b.	$I \triangleleft \{x \mapsto \mathsf{v}\}$	$\models$	$p(x)$	6b and $\neg$

4b and 7b are contradictory.
Both cases end in contradictions for arbitrary I. Thus F is valid.

Example: Prove

$$F : \; p(a) \; \rightarrow \; \exists x.\; p(x)$$

is valid.

Assume otherwise; i.e., F is false under interpretation $I : (D_I, \alpha_I)$:

1.	I	$\not\models$	F	assumption
2.	I	$\models$	$p(a)$	1 and $\rightarrow$
3.	I	$\not\models$	$\exists x.\; p(x)$	1 and $\rightarrow$
4.	$I \triangleleft \{x \mapsto \alpha_I[a]\}$	$\not\models$	$p(x)$	3 and $\exists$

2 and 4 are contradictory. Thus, F is valid.

Translations of English Sentences (famous theorems) into FOL

- The length of one side of a triangle is less than the sum of the lengths of the other two sides

$$\forall x, y, z.\; triangle(x, y, z) \; \rightarrow \; length(x) < length(y) + length(z)$$

- Fermat's Last Theorem.

$$\begin{aligned} &\forall n.\; integer(n) \wedge n > 2 \\ &\quad \rightarrow \; \forall x, y, z. \\ &\qquad integer(x) \wedge integer(y) \wedge integer(z) \\ &\qquad \wedge x > 0 \wedge y > 0 \wedge z > 0 \\ &\qquad\quad \rightarrow \; \exp(x, n) + \exp(y, n) \neq \exp(z, n) \end{aligned}$$

Example: Show that

$$F : \; (\forall x.\; p(x, x)) \; \rightarrow \; (\exists x.\; \forall y.\; p(x, y))$$

is invalid.

Find interpretation I such that F is false under I.

Choose $D_I = \{0, 1\}$
$p_I = \{(0, 0),\; (1, 1)\}$ i.e., $p_I(0, 0)$ and $p_I(1, 1)$ are true
$p_I(0, 1)$ and $p_I(1, 0)$ are false

$I[\forall x.\; p(x, x)] = \text{true}$ and $I[\exists x.\; \forall y.\; p(x, y)] = \text{false}$.

If we can find a falsifying interpretation for F, then F is invalid.

Is $F : (\forall x.\; p(x, x)) \; \rightarrow \; (\forall x.\; \exists y.\; p(x, y))$ valid?

Substitution

Suppose we want to replace one term with another in a formula; e.g., we want to rewrite

$$F : \; \forall y.\; (p(x, y) \; \rightarrow \; p(y, x))$$

as follows:

$$G : \; \forall y.\; (p(a, y) \; \rightarrow \; p(y, a)).$$

We call the mapping from x to a a substitution denoted as

$$\sigma : \{x \mapsto a\}.$$

We write $F\sigma$ for the formula G.

Another convenient notation is $F[x]$ for a formula containing the variable x and $F[a]$ for $F\sigma$.

Substitution

Definition (Substitution)

A substitution is a mapping from terms to terms; e.g.,

$$\sigma : \{t_1 \mapsto s_1, \ldots, t_n \mapsto s_n\}.$$

By $F\sigma$ we denote the application of σ to formula F;
i.e., the formula F where all occurrences of $t_1, \ldots, t_n$ are replaced by $s_1, \ldots, s_n$.

For a formula named $F[x]$ we write $F[t]$ as shorthand for $F[x]\{x \mapsto t\}$.

Renaming

Replace x in $\forall x$ by x' and all free occurrences[1] of x in $G[x]$, the scope of $\forall x$, by x':

$$\forall x.\ G[x] \quad \Leftrightarrow \quad \forall x'.\ G[x'].$$

Same for $\exists x$:

$$\exists x.\ G[x] \quad \Leftrightarrow \quad \exists x'.\ G[x'],$$

where x' is a fresh variable.

Example (renaming):

$$(\forall x.\ p(x) \ \rightarrow\ \exists x.\ q(x)) \ \wedge\ r(x)$$
$$\uparrow \forall x \qquad\qquad \uparrow \exists x \qquad \uparrow \text{free}$$

replace by the equivalent formula

$$(\forall y.\ p(y) \ \rightarrow\ \exists z.\ q(z)) \ \wedge\ r(x)$$

[1] Note: these occurrences are free in $G[x]$, *not* in $\forall x.\ G[x]$.

Safe Substitution I

Care has to be taken in the presence of quantifiers:

$$[x] : \exists y.\ y = Succ(x)$$
$$\uparrow \text{free}$$

What is $F[y]$?

We need to rename bound variables occurring in the substitution:

$$F[x] : \exists y'.\ y' = Succ(x)$$

Bound variable renaming does not change the models of a formula:

$$(\exists y.\ y = Succ(x)) \ \Leftrightarrow\ (\exists y'.\ y' = Succ(x))$$

Then under safe substitution

$$F[y] : \exists y'.\ y' = Succ(y)$$

Safe Substitution II

Example: Consider the following formula and substitution:

$$F : (\forall x.\ p(x, y)) \ \rightarrow\ q(f(y), \quad x)$$
$$\uparrow \text{free} \uparrow$$

Note that the only bound variable in F is the x in $p(x, y)$. The variables x and y are free everywhere else.

What is $F\sigma$? Use safe substitution!

1. Rename the bound x with a fresh name x':

$$F' : (\forall x'.\ p(x', y)) \ \rightarrow\ q(f(y), x)$$

2. $F\sigma : (\forall x'.\ p(x', f(x))) \ \rightarrow\ q(h(x, y), g(x))$

Safe Substitution III

Proposition (Substitution of Equivalent Formulae)

$$\sigma : \{F_1 \mapsto G_1,\ \cdots,\ F_n \mapsto G_n\}$$

s.t. for each i, $F_i \ \Leftrightarrow\ G_i$

If $F\sigma$ is a safe substitution, then $F \ \Leftrightarrow\ F\sigma$.

Semantic Tableaux (with Substitution)

We assume that there are infinitely many constant symbols.
The following rules are used for quantifiers:

$$\frac{I \models \forall x.\ F[x]}{I \models F[t]} \quad \text{for any term } t$$

$$\frac{I \not\models \forall x.\ F[x]}{I \not\models F[a]} \quad \text{for a fresh constant } a$$

$$\frac{I \models \exists x.\ F[x]}{I \models F[a]} \quad \text{for a fresh constant } a$$

$$\frac{I \not\models \exists x.\ F[x]}{I \not\models F[t]} \quad \text{for any term } t$$

The contradiction rule is similar to that of propositional logic:

$$\frac{\begin{array}{c} I \models p(t_1, \ldots, t_n) \\ I \not\models p(t_1, \ldots, t_n) \end{array}}{I \models \bot}$$

Example: Show that

$F : (\exists x.\ \forall y.\ p(x,y)) \ \rightarrow\ (\forall x.\ \exists y.\ p(y,x))$ is valid.

Rename to $F' : (\exists x.\ \forall y.\ p(x,y)) \ \rightarrow\ (\forall x'.\ \exists y'.\ p(y',x'))$.

Assume otherwise.

1.	I	$\not\models$	F'	assumption
2.	I	$\models$	$\exists x.\ \forall y.\ p(x,y)$	1 and $\rightarrow$
3.	I	$\not\models$	$\forall x'.\ \exists y'.\ p(y',x')$	1 and $\rightarrow$
4.	I	$\models$	$\forall y.\ p(a,y)$	2, $\exists$ (a fresh)
5.	I	$\not\models$	$\exists y'.\ p(y',b)$	3, $\forall$ (b fresh)
6.	I	$\models$	$p(a,b)$	4, $\forall$ ($t := b$)
7.	I	$\not\models$	$p(a,b)$	5, $\exists$ ($t := a$)
8.	I	$\models$	$\bot$	6, 7 contradictory

Thus, the formula is valid.

Example: Is $F : (\forall x.\ p(x,x)) \ \rightarrow\ (\exists x.\ \forall y.\ p(x,y))$ valid?

Rename to $F' : (\forall z.\ p(z,z)) \ \rightarrow\ (\exists x.\ \forall y.\ p(x,y))$

Assume I falsifies F' and apply semantic argument:

1.	I	$\not\models$	F'	assumption
2.	I	$\models$	$\forall z.\ p(z,z)$	1 and $\rightarrow$
3.	I	$\not\models$	$\exists x.\ \forall y.\ p(x,y)$	1 and $\rightarrow$
4.	I	$\models$	$p(a_1,a_1)$	2, $\forall$, $a_1 \in D_I$ fresh
5.	I	$\not\models$	$\forall y.\ p(a_1,y)$	3, $\exists$
6.	I	$\not\models$	$p(a_1,a_2)$	5, $\forall$, $a_2 \in D_I$ fresh
7.	I	$\models$	$p(a_2,a_2)$	2, $\forall$
8.	I	$\not\models$	$\forall y.\ p(a_2,y)$	3, $\exists$
9.	I	$\not\models$	$p(a_2,a_3)$	8, $\forall$, $a_3 \in D_I$ fresh

⋮

No contradiction. Falsifying interpretation I:

$$D_I = \mathbb{N}, \quad p_I(x,y) = \begin{cases} \text{true} & y = x, \\ \text{false} & y = x+1, \\ \text{arbitrary} & \text{otherwise.} \end{cases}$$

Formula Schemata

Formula

$(\forall x.\ p(x)) \ \leftrightarrow\ (\neg\exists x.\ \neg p(x))$

Formula Schema

$H_1 : (\forall x.\ F) \ \leftrightarrow\ (\neg\exists x.\ \neg F)$

$\uparrow$ place holder

Formula Schema (with side condition)

$H_2 : (\forall x.\ F) \ \leftrightarrow\ F$ provided $x \notin \mathit{free}(F)$

Valid Formula Schema

H is valid iff it is valid for any FOL formula F_i obeying the side conditions.

Example: H_1 and H_2 are valid.

Substitution σ of H

$$\sigma : \{F_1 \mapsto G_1, \ldots, F_n \mapsto G_n\}$$

mapping place holders F_i of H to FOL formulae G_i,
obeying the side conditions of H

Proposition (Formula Schema)

If H is a valid formula schema, and
σ is a substitution obeying H's side conditions,
then $H\sigma$ is also valid.

Example:

$H : (\forall x.\ F) \ \leftrightarrow\ F$ provided $x \notin \mathit{free}(F)$ is valid.

$\sigma : \{F \mapsto p(y)\}$ obeys the side condition.

Therefore $H\sigma : \forall x.\ p(y) \ \leftrightarrow\ p(y)$ is valid.

Proving Validity of Formula Schemata I

Example: Prove validity of

$$H : (\forall x.\ F) \ \leftrightarrow\ F \quad \text{provided } x \notin \mathit{free}(F).$$

Proof by contradiction. Consider the two directions of $\leftrightarrow$.

- First case

1.	I	$\models$	$\forall x.\ F$	assumption
2.	I	$\not\models$	F	assumption
3.	I	$\models$	F	1, $\forall$, since $x \notin$ free(F)
4.	I	$\models$	$\bot$	2, 3

Proving Validity of Formula Schemata II

- Second Case

1.	I	$\not\models$	$\forall x.\ F$	assumption
2.	I	$\models$	F	assumption
3.	I	$\models$	$\exists x.\ \neg F$	1 and $\neg$
4.	I	$\models$	$\neg F$	3, $\exists$, since $x \notin \text{free}(F)$
5.	I	$\models$	$\bot$	2, 4

Hence, H is a valid formula schema.

Normal Forms

1. Negation Normal Forms (NNF)

Apply the additional equivalences (left-to-right)

$$\neg\forall x.\ F[x] \Leftrightarrow \exists x.\ \neg F[x]$$

$$\neg\exists x.\ F[x] \Leftrightarrow \forall x.\ \neg F[x]$$

when converting PL formulae into NNF.

Example: $G:\ \forall x.\ (\exists y.\ p(x,y) \wedge p(x,z)) \rightarrow \exists w. p(x,w)$.

1. $\forall x.\ (\exists y.\ p(x,y) \wedge p(x,z)) \rightarrow \exists w.\ p(x,w)$
2. $\forall x.\ \neg(\exists y.\ p(x,y) \wedge p(x,z)) \vee \exists w.\ p(x,w)$
   $F_1 \rightarrow F_2 \Leftrightarrow \neg F_1 \vee F_2$
3. $\forall x.\ (\forall y.\ \neg(p(x,y) \wedge p(x,z))) \vee \exists w.\ p(x,w)$
   $\neg\exists x.\ F[x] \Leftrightarrow \forall x.\ \neg F[x]$
4. $G' : \forall x.\ (\forall y.\ \neg p(x,y) \vee \neg p(x,z)) \vee \exists w.\ p(x,w)$

G' in NNF and $G' \Leftrightarrow G$.

2. Prenex Normal Form (PNF)

All quantifiers appear at the beginning of the formula

$$Q_1 x_1 \cdots Q_n x_n.\ F[x_1, \cdots, x_n]$$

where $Q_i \in \{\forall,\ \exists\}$ and F is quantifier-free.

Every FOL formula F can be transformed to formula F' in PNF s.t. $F' \Leftrightarrow F$:

- Write F in NNF,
- rename quantified variables to fresh names, and
- move all quantifiers to the front. Be careful!

Example: Find equivalent PNF of

$$F:\ \forall x.\ \neg(\exists y.\ p(x,y) \wedge p(x,z)) \vee \exists y.\ p(x,y)$$

↑ to the end of the formula

1. Write F in NNF

$$F_1:\ \forall x.\ (\forall y.\ \neg p(x,y) \vee \neg p(x,z)) \vee \exists y.\ p(x,y)$$

2. Rename quantified variables to fresh names

$$F_2:\ \forall x.\ (\forall y.\ \neg p(x,y) \vee \neg p(x,z)) \vee \exists w.\ p(x,w)$$

↑Both are in the scope of $\forall x$↑

3. Remove all quantifiers to produce quantifier-free formula

$$F_3:\ \neg p(x,y) \vee \neg p(x,z) \vee p(x,w)$$

4. Add the quantifiers before F_3

$$F_4 : \forall x.\ \forall y.\ \exists w.\ \neg p(x,y) \vee \neg p(x,z) \vee p(x,w)$$

Alternately,

$$F_4' : \forall x.\ \exists w.\ \forall y.\ \neg p(x,y) \vee \neg p(x,z) \vee p(x,w)$$

Note: In F_2, $\forall y$ is in the scope of $\forall x$, therefore the order of quantifiers must be $\cdots \forall x \cdots \forall y \cdots$.
Also, $\exists w$ is in the scope of $\forall x$, therefore the order of the quantifiers must be $\cdots \forall x \cdots \exists w \cdots$

$$\boxed{F_4 \Leftrightarrow F \text{ and } F_4' \Leftrightarrow F}$$

Note: However, possibly, $G \not\Leftrightarrow F$ *and* $G' \not\Leftrightarrow F$, for

$$G : \forall y.\ \exists w.\ \forall x.\ \neg p(x,y) \vee \neg p(x,z) \vee p(x,w)$$

$$G' : \exists w.\ \forall x.\ \forall y.\ \cdots .$$

Decidability of FOL

- FOL is undecidable (Turing & Church)
  There does not exist an algorithm for deciding if a FOL formula F is {valid, satisfiable}; i.e., that always halts and says "yes" if F is {valid, satisfiable} or "no" if F is {invalid, unsatisfiable}.
- FOL is semi-decidable
  There is a procedure that always halts and says "yes" if F is {valid, unsatisfiable}, but may not halt if F is {invalid, satisfiable}.

On the other hand,

- PL is decidable
  There does exist an algorithm for deciding if a PL formula F is {valid, satisfiable}; e.g., the truth-table procedure.

Semantic Argument Method

To show FOL formula F is valid, assume $I \not\models F$ and derive a contradiction $I \models \bot$ in all branches

- Method is sound
  If every branch of a semantic argument proof reaches $I \models \bot$, then F is valid
- Method is complete
  Each valid formula F has a semantic argument proof in which every branch reaches $I \models \bot$