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
	$a, x, f(a), g(x, b), f(g(x, g(b)))$
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: constant

0-ary predicates: $P, Q, R, \ldots$

quantifiers

existential quantifier $\exists x. F[x]$

"there exists an x such that $F[x]$"

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 $q(x,f(x))$"

Translations of English Sentences 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.

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

FOL Semantics

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

- Domain D_I
  non-empty set of values or objects
  cardinality $|D_I|$ finite (eg, 52 cards),
  countably infinite (eg, integers), or
  uncountably infinite (eg, reals)
- 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

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

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

Example:

$$F: \; p(f(x,y),z) \;\rightarrow\; p(y, g(z,x))$$

Interpretation $I : (D_I, \alpha_I)$

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

$\alpha_I : \{f \mapsto +, g \mapsto -, p \mapsto >\}$

Therefore, we can write

$$F_I : x + y > z \;\rightarrow\; y > z - x$$

(This is the way we'll write it in the future!)

Also

$\alpha_I : \{x \mapsto 13, y \mapsto 42, z \mapsto 1\}$

Thus

$$F_I : 13 + 42 > 1 \;\rightarrow\; 42 > 1 - 13$$

Compute the truth value of F under I

1.	I	$\models$	$x + y > z$	since $13 + 42 > 1$
2.	I	$\models$	$y > z - x$	since $42 > 1 - 13$
3.	I	$\models$	F	by 1, 2, and $\rightarrow$

F is true under I

Semantics: Quantifiers

x variable.
<u>x-variant</u> of interpretation 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 $J : \ I \triangleleft \{x \ \mapsto \ \mathsf{v}\}$ the x-variant of I in which $\alpha_J[x] = \mathsf{v}$ for some $\mathsf{v} \in D_I$. Then

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

Example

For $\mathbb{Q}$, the set of rational numbers, consider

$$F_I : \forall x.\ \exists y.\ 2 \times y = x$$

Compute the value of F_I (F under I):

Let

$$\begin{array}{lll} J_1 : I \triangleleft \{x \ \mapsto \ \mathrm{v}\} & \qquad & J_2 : J_1 \triangleleft \{y \ \mapsto \ \frac{\mathrm{v}}{2}\} \\ x\text{-variant of } I & & y\text{-variant of } J_1 \end{array}$$

for $\mathrm{v} \in \mathbb{Q}$.

Then

$$\begin{array}{llcll} 1. & J_2 & \models & 2 \times y = x & \text{since } 2 \times \frac{\mathrm{v}}{2} = v \\ 2. & J_1 & \models & \exists y.\ 2 \times y = x & \\ 3. & I & \models & \forall x.\ \exists y.\ 2 \times y = x & \text{since } \mathrm{v} \in \mathbb{Q} \text{ is arbitrary} \end{array}$$

Satisfiability and Validity

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

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

<u>Example</u>: $F : (\forall x.\ p(x)) \leftrightarrow (\neg \exists x.\ \neg p(x))$ valid?

Suppose not. Then there is I s.t.

0. $I \not\models (\forall x.\ p(x)) \leftrightarrow (\neg \exists x.\ \neg p(x))$

First case

1.	I	$\models$	$\forall x.\ p(x)$	assumption
2.	I	$\not\models$	$\neg \exists x.\ \neg p(x)$	assumption
3.	I	$\models$	$\exists x.\ \neg p(x)$	2 and $\neg$
4.	$I \triangleleft \{x \mapsto \mathsf{v}\}$	$\models$	$\neg p(x)$	3 and $\exists$, for some $\mathsf{v} \in D_I$
5.	$I \triangleleft \{x \mapsto \mathsf{v}\}$	$\models$	$p(x)$	1 and $\forall$

4 and 5 are contradictory.

Second case

$$
\begin{array}{lrcll}
1. & I & \not\models & \forall x.\ p(x) & \text{assumption} \\
2. & I & \models & \neg \exists x.\ \neg p(x) & \text{assumption} \\
3. & I \triangleleft \{x \mapsto \mathrm{v}\} & \not\models & p(x) & \text{1 and } \forall\text{, for some v} \in D_I \\
4. & I & \not\models & \exists x.\ \neg p(x) & \text{2 and } \neg \\
5. & I \triangleleft \{x \mapsto \mathrm{v}\} & \not\models & \neg p(x) & \text{4 and } \exists \\
6. & I \triangleleft \{x \mapsto \mathrm{v}\} & \models & p(x) & \text{5 and } \neg
\end{array}
$$

3 and 6 are contradictory.

Both cases end in contradictions for arbitrary $I \Rightarrow F$ is valid.

Example: Prove

$$F: \; p(a) \;\rightarrow\; \exists x.\; p(x) \quad \text{is valid.}$$

Assume otherwise.

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

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

Example: Show

$$F: \; (\forall x.\ p(x,x)) \;\rightarrow\; (\exists x.\ \forall y.\ p(x,y)) \quad \text{is invalid.}$$

Find interpretation I such that

$$I \;\models\; \neg[(\forall x.\ p(x,x)) \;\rightarrow\; (\exists x.\ \forall y.\ p(x,y))]$$

i.e.

$$I \;\models\; (\forall x.\ p(x,x)) \;\wedge\; \neg(\exists x.\ \forall y.\ p(x,y))$$

Choose $D_I = \{0,1\}$

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

$p_I(1,0)$ and $p_I(1,0)$ are false

I falsifying interpretation $\Rightarrow$ F is invalid.

Safe Substitution $F\sigma$

Example:

$$F : (\forall x.\ \overbrace{p(x, y)}^{\text{scope of } \forall x}\ )\ \rightarrow\ q(f(y),\ x)$$

bound by $\forall x$ ↗ ↖ free free ↗ ↖ free

$$free(F) = \{x,\ y\}$$

substitution

$$\sigma : \{x \mapsto g(x),\ y \mapsto f(x),\ q(f(y), x) \mapsto \exists x.\ h(x, y)\}$$

$F\sigma$?

1. Rename
$$F' : \forall x'.\ p(x', y)\ \rightarrow\ q(f(y), x)$$
$$\uparrow \qquad \uparrow$$
where x' is a fresh variable

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

Rename x by x':

replace x in $\forall x$ by x' and all free x in 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

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 $F\sigma$ a safe substitution, then $F \ \Leftrightarrow\ F\sigma$

Formula Schema

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)

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

Valid Formula Schema

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

Example: H_1 and H_2 are valid.

Substitution σ of H

$$\sigma : \{F_1 \mapsto \quad , \ldots , F_n \mapsto \quad \}$$

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

Proposition (Formula Schema)

If H is 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 Schema

Example: Prove validity of

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

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 $x \notin \text{free}(F)$
4.	I	$\models$	$\perp$	2, 3

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	$\models$	$\perp$	2, 4

Hence, H is a valid formula schema.

Normal Forms

1. Negation Normal Forms (NNF)

Augment the equivalence with (left-to-right)

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

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

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

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. $F' \Leftrightarrow F$.

Example: Find equivalent PNF of

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

$\uparrow$ 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

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

$\uparrow$ 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,

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

<u>Note</u>: In F_2, $\forall y$ is <u>in the scope</u> of $\forall x$, therefore the order of quantifiers must be $\cdots \forall x \cdots \forall y \cdots$

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

<u>Note</u>: However $G \ \not\Leftrightarrow \ F$

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

Decidability of FOL

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

On the other hand,

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

Similarly for satisfiability

Semantic Argument Proof

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

- Soundness
  If every branch of a semantic argument proof reach $I \models \perp$, then F is valid
- Completeness
  Each valid formula F has a semantic argument proof in which every branch reach $I \models \perp$