CS5733 Program Synthesis

#7.FOL

Ashish Mishra, August 20, 2024

Partly based on slides by Roopsha Samata at Purdue

Roadmap

- Previously
  - PL
  - SAT Solving
- Today
  - Syntax and Semantics of first order logic (FOL)
  - Semantic argument method for FOL validity
  - Properties of FOL

Propositional Logic

$$P \wedge Q \rightarrow P \vee \neg Q$$

- Simple, not very expressive
- Decidable
  - Automated reasoning about satisfiability/validity

First-Order Logic
(predicate logic/predicate calculus/
relational logic)

$$\forall x.\, p(x, y) \rightarrow \exists y.\, \neg q(x, y)$$

- Very expressive
- Semi-decidable
  - Not fully automated

Syntax of FOL

constants: a, b, c
variables: x, y, z
n-ary functions: f, g, h
n-ary predicates: p, q, r

logical connectives: ¬, ∨, ∧, →, ↔
quantifiers: ∃, ∀

Term
constant, variable, or,
n-ary function applied to n terms

Atom
⊤, ⊥, or,
n-ary predicate applied to n terms

Literal
atom or its negation

FOL formula:
Literal, or, application of logical connectives to an FOL formula, or, application of a quantifier to an FOL formula

Quantifiers

existential quantifier: $\exists x.\, F(x)$ “there exists an x such that $F(x)$”

universal quantifier: $\forall x.\, \underbrace{F(x)}$ “for all x, $F(x)$”

Quantified variable

Scope of quantified variable

A variable is **bound** if there exists an occurrence in the scope of some quantifier

A variable is **free** if there exists an occurrence not bound by any quantifier

A variable may be both bound and free!

Closed/Ground formula: no free variables

Open formula: some free variables

Ground, quantifier-free formula: no variables

In a given formula

Example

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

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

English to 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.

no three positive integers x, y, and z satisfy the equation $x^n + y^n = z^n$ for any integer value of n greater than 2.

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

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

    $$f_I : \; D_I^n \rightarrow D_I$$

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

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

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

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 <u>true</u> 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. $I \triangleleft \{x \mapsto \mathsf{v}\} \models F$

I is an interpretation of ∀x. F iff all x-variants of I are interpretations of F . I is an interpretation of ∃x. F iff some x-variant of I is an interpretation of 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

$$\begin{array}{ll} J_1 : I \lhd \{x \mapsto \mathrm{v}\} & \quad J_2 : J_1 \lhd \{y \mapsto \frac{\mathrm{v}}{2}\} \\ x\text{-variant of } I & \quad y\text{-variant of } J_1 \end{array}$$

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

Same as PL

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

F is valid iff $\neg F$ is unsatisfiable

Semantic rules: given an interpretation I with domain D_I,

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

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

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

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

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:

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

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.

Examples

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

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

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.

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.

Example: Show

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

To prove F is invalid,

just find an I. I $\models \neg$F

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.

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 $F[x]\{x \mapsto t\}$.

Scope and 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):

$$\begin{array}{c} (\forall x.\ p(x) \ \rightarrow\ \exists x.\ q(x)) \ \wedge\ r(x) \\ \quad\ \uparrow \forall x \qquad\qquad \uparrow \exists x \qquad \uparrow \text{free} \end{array}$$

replace by the equivalent formula

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

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

$$\text{bound by } \forall x \nearrow \ \nwarrow \text{free} \qquad \text{free} \nearrow \qquad \nwarrow \text{free}$$

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

Safe Substitution I

Care has to be taken in the presence of quantifiers:

$$\mathsf{F}\,[x] : \exists y.\ y = Succ(x)$$

$\uparrow$ free

What is $F[y]$? Variable Capture

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

<u>Example</u>: Consider the following formula and substitution:

$$F : (\forall x.\ p(x, y)) \ \rightarrow \ q(f(y), \quad x) \qquad \sigma : \ \{x \mapsto g(x),\ y \mapsto f(x),\ q(f(y), x) \mapsto \exists x.\ h(x, y)\}$$

$$\uparrow \text{free} \uparrow$$

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 $F \Leftrightarrow F\sigma$.

Formula Schema

<u>Formula</u>

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

We proved the validity of this earlier

<u>Formula Schema</u>

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

↑ place holder

<u>Formula Schema (with side condition)</u>

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

<u>Valid Formula Schema</u>

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

<u>Example</u>: 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 $x \notin \text{free}(F)$
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 $x \notin \mathrm{free}(F)$
5.	I	$\models$	$\perp$	2, 4

Hence, H is a valid formula schema.

Normal forms are for FOL as well

1. Negation Normal Forms (NNF)

Augment the equivalence with (left-to-right)

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

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

Schema equivalences

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. $F' \ \Leftrightarrow\ F$.

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

<u>Example</u>: 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

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

$^{\uparrow}$Both are in the scope of $\forall x^{\uparrow}$

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

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

Alternately,

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

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

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

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

Some meta properties of FOL

Soundness and Completeness of Proof Rules

Semantic Argument Proof

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

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

(Un)Decidability of FOL

A problem is decidable if there exists a procedure that, for any input:
1. halts and says “yes” if answer is positive, and
2. halts and says “no” if answer is negative
(Such a procedure is called an algorithm or a decision procedure)

Undecidability of FOL [Church and Turing]:
Deciding the validity of an FOL formula is undecidable

Deciding the validity of a PL formula is decidable
The truth table method is a decision procedure

Church

Turing

Semi-decidability of FOL

A problem is semi-decidable iff there exists a procedure that, for any input:
1. halts and says “yes” if answer is positive, and
2. may not terminate if answer is negative.

Semi-decidability of FOL:
For every valid FOL formula, there exists a procedure (semantic argument method) that always terminates and says “yes”.
If an FOL formula is invalid, there exists no procedure that is guaranteed to terminate.

Summary and Logistics

- Thanks for the submissions and sorry for the confusion.
- No (compulsory) reading this week, will encourage reading CoC Text.
- Next Class, FO Theories and Satisfiability Modulo Theory (SMT) Solvers.
- Discuss the paper in the second half of the class.