AAA528: Computational Logic

Lecture 5 — First-Order Theories

Hakjoo Oh
2025 Spring

Review: First-Order Logic

- FOL is an extension of PL with quantifiers and nonlogical symbols. A first-order logic formula consists of
  - variables,
  - logical symbol (boolean connectives and quantifiers), and
  - nonlogical symbols (function, predicate, and constant symbols).
- The semantics is determined by an interpretation. An interpretation consists of a domain (D) and an assignment (I) for free variables and nonlogical symbols.
  - For example, $\exists x. x + 0 = 1$ is true under the conventional interpretation but it is false if we choose to interpret $+$ as multiplication.

First-Order Theories

- In practice, we are not interested in pure logical validity (i.e. valid in all interpretations) of FOL formulas but in validity in a specific class of interpretations.
  - E.g. $\exists x. x + 0 = 1$
- First-order logic is rather a general framework for building a specific, restricted logic, which provides a generic syntax and building blocks for defining the restrictions, called **theories**.
- The restrictions are made on nonlogical symbols and interpretations. For instance, in the *theory of integers*, only $+$ and $-$ are allowed for function symbols with their conventional interpretations.
- One natural way for restricting interpretations is to provide a set of axioms; we only consider interpretations that satisfy the axioms.

Example: The Theory of Equality

A theory with a fixed interpretation for $=$. For example, the formula must be valid according to the conventional interpretation of $=$:

$$\forall x, y, z.\ (((x = y) \land \neg(y = z)) \implies \neg(x = z)).$$

To fix this interpretation, it is sufficient to enforce the following axioms:

1. Reflexivity: $\forall x.\ x = x$
2. Symmetry: $\forall x, y.\ x = y \implies y = x$
3. Transitivity: $\forall x, y, z.\ x = y \land y = z \implies x = z$

First-Order Theories

A first-order theory T is defined by the two components:

- **Signature**: A set of nonlogical symbols. Given a signature Σ, a Σ-formula is one whose nonlogical symbols are from Σ. Signature restricts the syntax.
- **Axioms**: A set of closed FOL formulas whose nonlogical symbols are from Σ. Axioms restrict the interpretations.

Terminologies

- Given a first-order theory T, a Σ-formula φ is T-**satisfiable** if there exists an interpretation that satisfies both the formula and the axioms.
- Similarly, φ is T-**valid** if all interpretations that satisfy the axioms also satisfy φ. We write

$$T \vDash F$$

  for T-validity of F.
- A theory T is **decidable** if there exists a decision procedure for checking T-validity: $T \vDash F$ is decidable for every Σ-formula F.
- A theory T is **consistent** if there is at least one T-interpretation, i.e., an interpretation that satisfies the axioms of T: for every $F \in A$,

$$I \vDash F$$

  In a consistent theory T, there does not exist a Σ-formula F such that $T \vDash F$ and $T \vDash \neg F$.
- A theory T is **complete** if for every closed Σ-formula F, $T \vDash F$ or $T \vDash \neg F$.

Fragments of Theories

A theory restricts only the nonlogical symbols. Restrictions on the logical symbols or the grammar are done by defining **fragments** of the logic. Two popular fragments:

- **Quantifier-free fragment**: the set of Σ-formulas without quantifiers.
- **Conjunctive fragment**: the set of formulas where the only boolean connective that is allowed is conjunction.

Many first-order theories are undecidable while their quantifier-free fragments are decidable. In practice, we are mostly interested in the satisfiability problem of the quantifier-free fragment of first-order theories.

First-Order Theories for Programs

First-order theories useful for reasoning about programs:

- Equality
- Integers, rationals, and reals
- Lists, arrays
- Pointers
- Bit-vectors
- ...

Theory of Equality (with Uninterpreted Functions)

The theory of equality T_E is the simplest and most widely-used first-order theory. Its signature

$$\Sigma_E : \{=, a, b, c, \ldots, f, g, h, \ldots, p, q, r, \ldots\}$$

consists of

- $=$ (equality), a binary predicate;
- and all constant, function, and predicate symbols.

Equality $=$ is an **interpreted** predicate symbol; its meaning will be defined via the axioms. The others are **uninterpreted** since functions, predicates, and constants are left unspecified.

Theory of Equality (with Uninterpreted Functions)

The axioms of T_E:

1. Reflexivity: $\forall x.\ x = x$
2. Symmetry: $\forall x, y.\ x = y \implies y = x$
3. Transitivity: $\forall x, y, z.\ x = y \land y = z \implies x = z$
4. Function congruence (consistency): for each positive integer n and n-ary function symbol f,

$$\forall \vec{x}, \vec{y}.\ (\bigwedge_{i=1}^{n} x_i = y_i) \rightarrow f(\vec{x}) = f(\vec{y}).$$

5. Predicate congruence (consistency): for each positive integer n and n-ary predicate symbol p,

$$\forall \vec{x}, \vec{y}.\ (\bigwedge_{i=1}^{n} x_i = y_i) \rightarrow (p(\vec{x}) \leftrightarrow p(\vec{y})).$$

Example

To prove that

$$F : a = b \land b = c \rightarrow g(f(a), b) = g(f(c), a)$$

is T_E-valid, assume otherwise to derive a contradiction:

1.	$I \not\models F$	assumption
2.	$I \models a = b \land b = c$	1, $\rightarrow$
3.	$I \not\models g(f(a), b) = g(f(c), a)$	1, $\rightarrow$
4.	$I \models a = b$	2, $\land$
5.	$I \models b = c$	2, $\land$
6.	$I \models a = c$	4, 5, transitivity
7.	$I \models f(a) = f(c)$	6, function congruence
8.	$I \models b = a$	4, symmetry
9.	$I \models g(f(a), b) = g(f(c), a)$	7, 8, function congruence
10.	$I \models \bot$	3, 9

Decidability

Like the full first-order logic, T_E-validity is undecidable. However, there exists an efficient decision procedure (see Chap. 9) for its quantifier-free fragment.

Uninterpreted Functions

- In T_E, function symbols are uninterpreted since the axioms do not assign meaning to them other than in the context of equality. The only thing we know about them is that they are functions (function congruence).
- Uninterpreted functions can also be used in other theories. For example, in the formula

$$F(x) = F(G(y)) \lor x + 1 = y,$$

  F and G are uninterpreted.

Use of Uninterpreted Functions

A main application of uninterpreted functions is to *abstract* complex formulas that are otherwise difficult to automatically reason about.

- In a formula F, treating a function symbol f as uninterpreted makes the formula weaker; we ignore the semantics of f except for congruence w.r.t. equality.
- Let φ^{UF} be the formula derived from φ by replacing some interpreted functions with uninterpreted ones. Then,

$$\vDash \varphi^{UF} \implies \vDash \varphi.$$

  while the converse is not true.
- φ_{UF} is an *approximation* of φ such that if φ^{UF} is valid so is φ. But φ^{UF} may fail to be valid though φ is.

Uninterpreted functions simplify proofs. Uninterpreted functions let us reason about systems while ignoring the semantics of irrelevant parts.

Example

Consider the task of proving that the two C functions behave the same:

```
int power3 (int in) {
  int i, out;
  out = in;
  for (i=0; i<2; i++)
    out = out * in;
  return out;
}
```

```
int power3_new (int in) {
  int out;
  out = (in * in) * in;
  return out;
}
```

We can prove the equivalence by transforming the programs into formulas

$$\varphi_a : out_0 = in \land out_1 = out_0 * in \land out_2 = out_1 * in$$
$$\varphi_b : out = (in * in) * in$$

and proving the validity of the following formula:

$$\varphi_a \land \varphi_b \to out_2 = out$$

Example

Deciding formula with multiplication is generally hard. Replacing the multiplication symbol with uninterpreted functions can aid the problem:

$$\begin{array}{l} \varphi_a^{UF} : out_0 = in \land out_1 = G(out_0, in) \land out_2 = G(out_1, in) \\ \varphi_b^{UF} : out = G(G(in, in), in) \end{array}$$

Check the validity of

$$\varphi_a^{UF} \land \varphi_b^{UF} \rightarrow out_2 = out$$

This abstract formula is valid and so is the original (concrete) formula.

Theory of Peano Arithmetic (First-Order Arithmetic)

A theory for natural numbers. The theory of Peano arithmetic T_{PA} has signature

$$\Sigma_{PA} : \{0, 1, +, \cdot, =\}$$

where

- $\mathbf{0}$ and $\mathbf{1}$ are constants;
- $+$ (addition) and $\cdot$ (multiplication) are binary functions;
- and $=$ (equality) is a binary predicate.

Theory of Peano Arithmetic

The axioms of T_{PA}:

1. Zero: $\forall x.\ \neg(x + 1 = 0)$
2. Successor: $\forall x, y.\ x + 1 = y + 1 \rightarrow x = y$
3. Plus zero: $\forall x.\ x + 0 = x$
4. Plus successor: $\forall x, y.\ x + (y + 1) = (x + y) + 1$
5. Times zero: $\forall x.\ x \cdot 0 = 0$
6. Times successor: $\forall x, y.\ (y + 1) = x \cdot y + x$
7. Induction: $F[0] \land (\forall x.\ F[x] \rightarrow F[x + 1]) \rightarrow \forall x.\ F[x]$ (for every Σ_{PA}-formula F with one free variable)

Example Formulas

- The formula $3x + 5 = 2y$ can be written as

$$(1 + 1 + 1) \cdot x + 1 + 1 + 1 + 1 + 1 = (1 + 1) \cdot y.$$

- The inequality $3x + 5 > 2y$ can be expressed by

$$\exists z.\ z \neq 0 \wedge 3x + 5 = 2y + z.$$

  The weak inequality $3x + 5 \geq 2y$?

- The Σ_{PA}-formula

$$\exists x, y, z.\ x \neq 0 \wedge y \neq 0 \wedge z \neq 0 \wedge xx + yy = zz$$

  is T_{PA}-valid.

- Every formula of the set

$$\{\forall x, y, z.\ x \neq 0 \wedge y \neq 0 \wedge z \neq 0 \rightarrow x^n + y^n \neq z^n \mid n > 2 \wedge n \in \mathbb{Z}\}$$

  is T_{PA}-valid.

Decidability and Completeness

T_{PA} is neither complete nor decidable. Even undecidable is its quantifier-free fragment. A fragment of T_{PA}, called Presburger arithmetic, is both complete and decidable.

Theory of Presburger Arithmetic

A restriction that does not allow multiplication. The theory has signature

$$\boldsymbol{\Sigma_{\mathbb{N}} : \{0, 1, +, =\}}$$

and axioms:

1. Zero: $\forall x.\ \neg(x + 1 = 0)$
2. Successor: $\forall x, y.\ x + 1 = y + 1 \rightarrow x = y$
3. Plus zero: $\forall x.\ x + 0 = x$
4. Plus successor: $\forall x, y.\ x + (y + 1) = (x + y) + 1$
5. Induction: $F[0] \wedge (\forall x.\ F[x] \rightarrow F[x + 1]) \rightarrow \forall x.\ F[x]$

Integers

- Integer reasoning can be performed with natural-number reasoning: formulas over all integers $\mathbb{Z} = \{\ldots, -1, 0, 1, \ldots\}$ can be encoded as $\Sigma_{\mathbb{N}}$-formulas.
- Idea: replace integer variables by the difference of variables of natural-numbers. For example, consider the formula

$$F_0 : \forall w, x. \exists y, z.\ x + 2y - z > -3w.$$

1. Introduce two variables, v_p and v_n, for each variable v of F_0:

$$\begin{aligned}&F_1 : \forall w_p, w_n, x_p, x_n. \exists y_p, y_n, z_p, z_n.\\ &(x_p - x_n) + 2(y_p - y_n) - (z_p - z_n) > -3(w_p - w_n).\end{aligned}$$

2. Move negated terms to the other side of the inequality:

$$\begin{aligned}&F_2 : \forall w_p, w_n, x_p, x_n. \exists y_p, y_n, z_p, z_n.\\ &x_p + 2y_p + z_n + 3w_p > x_n + 2y_n + z_p + 3w_n.\end{aligned}$$

F_2 is $T_{\mathbb{N}}$-valid precisely when F_0 is valid in the integer interpretation.

Theory of Integers

- Although integer reasoning can be done with natural numbers, it is convenient to have a theory of integers.
- The theory of integers $\boldsymbol{T_{\mathbb{Z}}}$ (with linear arithmetic) has signatures

  $$\boldsymbol{\Sigma_{\mathbb{Z}} : \{\ldots, -2, -1, 0, 1, 2, \ldots, -3\cdot, -2\cdot, 2\cdot, 3\cdot, \ldots, +, -, =, >\}}$$

  $\boldsymbol{T_{\mathbb{Z}}}$ is no more expressive but more convenient than Presburger arithmetic.
- $\boldsymbol{T_{\mathbb{Z}}}$ is both complete and decidable, and one of the most widely used theory.

Theories of Reals and Rationals

The theory of reals $T_{\mathbb{R}}$ has signature

$$\Sigma_{\mathbb{R}} : \{0, 1, +, -, \cdot, =, \geq\}$$

The theory of rationals $T_{\mathbb{Q}}$ has signature

$$\Sigma_{\mathbb{Q}} : \{0, 1, +, -, =, \geq\}$$

$T_{\mathbb{R}}$ and $T_{\mathbb{Q}}$ have complex axioms (see textbook).

Comparison with Peano Arithmetic

- T_{PA} is more complicated than $T_{\mathbb{R}}$: $T_{\mathbb{R}}$ is decidable while T_{PA} is not.
- Intuitively, T_{PA} is more difficult to decide because it is easier to find a solution in $T_{\mathbb{R}}$. Consider the formula

$$F : \exists x. 2x = 7.$$

  - In the theory of integers, F is invalid.
  - In the theories of reals, $x = 7/2$.

Theory of Lists

The theory of lists T_{cons}, has signature

$$\Sigma_{cons} : \{\mathbf{cons}, \mathbf{car}, \mathbf{cdr}, \mathbf{atom}, =\}$$

and axioms:

1. Reflexivity, symmetry, transitivity of T_E
2. Instantiations of function congruence for cons, car, and cdr:
    - $\forall x_1, x_2, y_1, y_2.\ x_1 = x_2 \wedge y_1 = y_2 \rightarrow \mathbf{cons}(x_1, y_1) = \mathbf{cons}(x_2, y_2)$
    - $\forall x, y.\ x = y \rightarrow \mathbf{car}(x) = \mathbf{car}(y)$
    - $\forall x, y.\ x = y \rightarrow \mathbf{cdr}(x) = \mathbf{cdr}(y)$
3. Instantiation of predicate congruence for atom:

$$\forall x, y.\ x = y \rightarrow (\mathbf{atom}(x) \leftrightarrow \mathbf{atom}(y))$$

4. $\forall x, y.\mathbf{car}(\mathbf{cons}(x, y)) = x$, $\forall x, y.\mathbf{cdr}(\mathbf{cons}(x, y)) = y$
5. $\forall x.\ \neg\mathbf{atom}(x) \rightarrow \mathbf{cons}(\mathbf{car}(x), \mathbf{cdr}(x)) = x$
6. $\forall x, y.\neg\mathbf{atom}(\mathbf{cons}(x, y))$

Example

The $(\Sigma_E \cup \Sigma_{cons})$-formula

$$F : \begin{array}{l} \mathsf{car}(a) = \mathsf{car}(b) \land \mathsf{cdr}(a) = \mathsf{cdr}(b) \land \neg \mathsf{atom}(a) \land \neg \mathsf{atom}(b) \\ \qquad \rightarrow f(a) = f(b) \end{array}$$

is $(\Sigma_E \cup \Sigma_{cons})$-valid:

1.	$I \not\models F$	assumption
2.	$I \models \mathsf{car}(a) = \mathsf{car}(b)$	$1, \rightarrow, \land$
3.	$I \models \mathsf{cdr}(b) = \mathsf{cdr}(b)$	$1, \rightarrow, \land$
4.	$I \models \neg \mathsf{atom}(a)$	$1, \rightarrow, \land$
5.	$I \models \neg \mathsf{atom}(b)$	$1, \rightarrow, \land$
6.	$I \not\models f(a) = f(b)$	$1, \rightarrow$
7.	$I \models \mathsf{cons}(\mathsf{car}(a), \mathsf{cdr}(a)) = \mathsf{cons}(\mathsf{car}(b), \mathsf{cdr}(b))$	$2, 3,$ congr.
8.	$I \models \mathsf{cons}(\mathsf{car}(a), \mathsf{cdr}(a)) = a$	$4,$ Axiom5
9.	$I \models \mathsf{cons}(\mathsf{car}(b), \mathsf{cdr}(b)) = b$	$5,$ Axiom5
10.	$I \models a = b$	$7, 8, 9,$ trans.
11.	$I \models f(a) = f(b)$	$10,$ congr.
12.	$I \models \bot$	$6, 11$

Theory of Arrays

The theory of arrays T_A has signature

$$\Sigma_A : \{\cdot[\cdot], \cdot\langle\cdot \lhd \cdot\rangle, =\}$$

where

- $a[i]$ (binary function) represents the value of array a at position i
- $a\langle i \lhd v\rangle$ (ternary function) represents the modified array a in which position i has value v
- $=$ is the equality predicate

The axioms of T_A:

1. the axioms of reflexivity, symmetry, and transitivity of T_E
2. (array congruence) $\forall a, i, j.\ i = j \rightarrow a[i] = a[j]$
3. (read-over-write 1) $\forall a, v, i, j.\ i = j \rightarrow a\langle i \lhd v\rangle[j] = v$
4. (read-over-write 2) $\forall a, v, i, j.\ i \neq j \rightarrow a\langle i \lhd v\rangle[j] = a[j]$

Example

The formula

$$F : a[i] = e \rightarrow \forall j.\ a\langle i \triangleleft e\rangle[j] = a[j]$$

is valid:

1.	$I \not\models F$	assumption
2.	$I \models a[i] = e$	$1, \rightarrow$
3.	$I \not\models \forall j. a\langle i \triangleleft e\rangle[j] = a[j]$	$1, \rightarrow$
4.	$I_1 : I \triangleleft \{j \mapsto v\} \not\models a\langle i \triangleleft e\rangle[j] = a[j]$	$3, \forall$, for some $v \in D$
5.	$I_1 \models a\langle i \triangleleft e\rangle[j] \neq a[j]$	$4, \neg$
6.	$I_1 \models i = j$	$5,$ read-over-wirte 2 (cntra
7.	$I_1 \models a[i] = a[j]$	$6,$ array congruence
8.	$I_1 \models a\langle i \triangleleft e\rangle[j] = e$	$6,$ read-over-write 1
9.	$I_1 \models a\langle i \triangleleft e\rangle[j] = a[j]$	$2, 7, 8,$ transitivity
10.	$I_1 \models \bot$	$4, 9$

Example

The formula

$$F : a[i] = e \rightarrow a\langle i \triangleleft e\rangle = a$$

is not T_A-valid, since $=$ is only defined for array elements. It becomes valid with the following axiom, called extensionality:

$$\forall a, b.\ (\forall i.\ a[i] = b[i]) \leftrightarrow a = b$$

1.	$I \not\models F$	assumption
2.	$I \models a[i] = e$	$1, \rightarrow$
3.	$I \not\models a\langle i \triangleleft e\rangle = a$	$1, \rightarrow$
4.	$I \models a\langle i \triangleleft e\rangle \neq a$	$3, \neg$
5.	$I \models \neg(\forall j.\ a\langle i \triangleleft e\rangle[j] = a[j])$	4, extensionality
6.	$I \not\models \forall j.\ a\langle i \triangleleft e\rangle[j] = a[j]$	$5, \neg$

The remaining proof proceeds as in the previous example.

Combining Theories

- In practice, the formulas we check for satisfiability or validity span multiple theories. For example, in program verification, we want to prove properties about a list of integers or an array of integers.
- Nelson and Oppen presented a general method for combining quantifier-free fragments of first-order theories. Given T_1 and T_2 such that $\Sigma_1 \cap \Sigma_2 = \{=\}$, the combines theory $T_1 \cup T_2$ has signature $\Sigma_1 \cup \Sigma_2$ and axioms $A_1 \cup A_2$. Nelson and Oppen showed that if
  - satisfiability in the quantifier-free fragments of T_1 is decidable
  - satisfiability in the quantifier-free fragments of T_2 is decidable
  - and certain conditions are met

  then satisfiability in the quantifier-free fragment of $T_1 \cup T_2$ is decidable. Furthermore, if the decision procedures for T_1 and T_2 are in P (in NP), then the combined decision procedure for $T_1 \cup T_2$ is in P (in NP).

Summary

Decidability of first-order theories:

Theory	Description	Full	QFF
$\boldsymbol{T_E}$	equality	no	yes
$\boldsymbol{T_{PA}}$	Peano arithmetic	no	no
$\boldsymbol{T_{\mathbb{N}}}$	Presburger arithmetic	yes	yes
$\boldsymbol{T_{\mathbb{Z}}}$	linear integers	yes	yes
$\boldsymbol{T_{\mathbb{R}}}$	reals (with $\cdot$)	yes	yes
$\boldsymbol{T_{\mathbb{Q}}}$	rationals (without $\cdot$)	yes	yes
$\boldsymbol{T_{RDS}}$	recursive data structures	no	yes
$\boldsymbol{T_A}$	arrays	no	yes
$\boldsymbol{T_A^=}$	arrays with extensionality	no	yes

Exercises

Use the semantic argument method to prove the validity of the following formulas, or identify a counterexample:

- In Theory of Equality:

$$f(f(f(a))) = f(f(a)) \land f(f(f(f(a)))) = a \rightarrow f(a) = a$$

- In Theory of Integers:

$$x \leq y \land z = x - 1 \rightarrow z \leq y$$

- In Theory of Lists:

$$\mathbf{car}(x) = y \land \mathbf{cdr}(x) = z \rightarrow x = \mathbf{cons}(y, z)$$

- In Theory of Arrays:

$$a\langle i \triangleleft e\rangle[j] = e \rightarrow i = j$$