First-Order Logic
First-Order Theories

Roopsha Samanta

Partly based on slides by Aaron Bradley and Isil Dillig
Roadmap

- Review: propositional logic
- Syntax and semantics of first-order logic (FOL)
- Semantic argument method for FOL validity
- Properties of FOL
- Overview of first-order theories
Propositional logic (PL) syntax

Atom
- truth symbols $\top$ ("true") and $\bot$ ("false")
- propositional variables $p, q, r, p_1, q_1$

Literal
- atom $\alpha$ or its negation $\neg\alpha$

Formula
- literal or application of a logical connective to $F, F_1, F_2$
  - $\neg F$ "not" (negation)
  - $F_1 \lor F_2$ "or" (disjunction)
  - $F_1 \land F_2$ "and" (conjunction)
  - $F_1 \rightarrow F_2$ "implies" (implication)
  - $F_1 \leftrightarrow F_2$ "if and only if" (iff)
PL semantics

Interpretation $I$: mapping of each propositional variable to a truth value

$I: \{ p \mapsto \top, q \mapsto \bot, \ldots \}$

Satisfying interpretation: $F$ evaluates to $\top$ under $I$, written $I \models F$

Falsifying interpretation: $F$ evaluates to $\bot$ under $I$, written $I \not\models F$
PL semantics: inductive definition

<table>
<thead>
<tr>
<th>Base Cases:</th>
<th>Inductive Cases:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I \models T$</td>
<td>$I \models \neg F \quad \text{iff} \quad I \not\models F$</td>
</tr>
<tr>
<td>$I \not\models \bot$</td>
<td>$I \models F_1 \lor F_2 \quad \text{iff} \quad I \models F_1 \quad \text{or} \quad I \models F_2$</td>
</tr>
<tr>
<td>$I \models p \quad \text{iff} \quad I[p] = T$</td>
<td>$I \models F_1 \land F_2 \quad \text{iff} \quad I \models F_1 \quad \text{and} \quad I \models F_2$</td>
</tr>
<tr>
<td>$I \not\models p \quad \text{iff} \quad I[p] = \bot$</td>
<td>$I \models F_1 \rightarrow F_2 \quad \text{iff} \quad I \not\models F_1 \quad \text{or} \quad I \models F_2$</td>
</tr>
<tr>
<td>$I \models F_1 \leftrightarrow F_2 \quad \text{iff} \quad I \models F_1 \quad \text{and} \quad I \models F_2$, or,</td>
<td>$I \not\models F_1 \quad \text{and} \quad I \not\models F_2$</td>
</tr>
</tbody>
</table>
Satisfiability and Validity

\( F \) is **satisfiable** iff there exists \( I : I \models F \)

\( F \) is **valid** iff for all \( I : I \models F \)

**Duality:**
\( F \) is valid iff \( \neg F \) is unsatisfiable

Procedure for deciding satisfiability *or* validity suffices!
Deciding satisfiability

- SAT solvers!

- Basic techniques
  - Truth table method: search-based
  - Semantic argument method: deductive technique

- SAT solvers combine search and deduction
Propositional Logic

\[ P \land Q \rightarrow P \lor \neg Q \]

- 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: \( \neg, \lor, \land, \rightarrow, \leftrightarrow \)
quantifiers: \( \exists, \forall \)

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

Atom
\( T, \bot \), 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: $\forall x. F(x)$  “for all $x$, $F(x)$”

An occurrence of a variable is **bound** if it’s in the scope of some quantifier

An occurrence of a variable is **free** if it’s not in the scope of any quantifier

**Closed** formula: no free variables

**Open** formula: some free variables

**Ground** formula: no variables
Semantics of FOL: first-order structure \( \langle U, I \rangle \)

- **Universe** of discourse/domain, \( U \):
  - Non-empty set of values or objects of interest
  - May be finite (set of students at Purdue), countably infinite (integers) or uncountable infinite (positive reals)

- **Interpretation**, \( I \): Mapping of variables, functions and predicates to values in \( U \)
  - \( I \) maps each variable symbol \( x \) to some value \( I[x] \in U \)
  - \( I \) maps each \( n \)-ary function symbol \( f \) to some function \( f_I: U^n \to U \)
  - \( I \) maps each \( n \)-ary predicate symbol \( p \) to some predicate \( p_I: U^n \to \{true, false\} \)
Evaluation of formulas

If $F$ evaluates to $\top$ under $U, I$, we write $\langle U, I \rangle \models F$

If $F$ evaluates to $\bot$ under $U, I$, we write $\langle U, I \rangle \not\models F$

- Evaluation of terms: $I[f(t_1, ..., t_n)] = I[f](I[t_1], ..., I[t_n])$
- Evaluation of atoms: $I[p(t_1, ..., t_n)] = I[p](I[t_1], ..., I[t_n])$
Evaluation of formulas: inductive definition

Base Cases:
\[ \langle U, I \rangle \models \top \]
\[ \langle U, I \rangle \not\models \bot \]
\[ \langle U, I \rangle \models p(t_1, ..., t_n) \]
iff \[ I[p(t_1, ..., t_n)] = \text{true} \]

Inductive Cases:
\[ \langle U, I \rangle \models \neg F \] iff \[ \langle U, I \rangle \not\models F \]
\[ \langle U, I \rangle \models F_1 \lor F_2 \] iff \[ \langle U, I \rangle \models F_1 \] or \[ \langle U, I \rangle \models F_2 \]
\[ \text{...} \]
\[ \langle U, I \rangle \models \forall x. F \] iff for all \[ v \in U, I[x \mapsto v] \models F \]
\[ \langle U, I \rangle \models \exists x. F \] iff there exists \[ v \in U, I[x \mapsto v] \models F \]

\(x\)-variant of \(\langle U, I \rangle\) that agrees with \(U, I\) on everything except the variable \(x\), with \(I[x] = v\).
Satisfiability and Validity

$F$ is **satisfiable** iff there exists some structure $\langle U, I \rangle : \langle U, I \rangle \models F$

$F$ is **valid** iff for all structures $\langle U, I \rangle : \langle U, I \rangle \models F$

**Duality:**

$F$ is valid iff $\neg F$ is unsatisfiable
Proof by contradiction:
1. Assume $F$ is not valid
2. Apply proof rules
3. Contradiction (i.e, $\perp$) along every branch of proof tree $\Rightarrow F$ is valid
4. Otherwise, $F$ is not valid
Semantic argument method for validity

\[ \langle U, I \rangle \models \forall x. F \quad \frac{\langle U, I[x \mapsto c] \rangle \models F}{\text{(for any } c \in U)} \]

\[ \langle U, I \rangle \not\models \forall x. F \quad \frac{\langle U, I[x \mapsto c] \rangle \not\models F}{\text{(for some fresh } c \in U)} \]

\[ \langle U, I \rangle \models \exists x. F \quad \frac{\langle U, I[x \mapsto c] \rangle \models F}{\text{(for some fresh } c \in U)} \]

\[ \langle U, I \rangle \not\models \exists x. F \quad \frac{\langle U, I[x \mapsto c] \rangle \not\models F}{\text{(for any } c \in U)} \]

\[ \langle U, I \rangle \models \neg F \quad \frac{\langle U, I \rangle \not\models F}{\langle U, I \rangle \models \neg F} \]

\[ \langle U, I \rangle \models F \wedge G \quad \frac{\langle U, I \rangle \models F \quad \langle U, I \rangle \models G}{\langle U, I \rangle \models F \wedge G} \]

\[ \langle U, I \rangle \not\models F \quad \frac{\langle U, I \rangle \not\models G}{\langle U, I \rangle \not\models F \wedge G} \]

\[ \langle U, I \rangle \models F \rightarrow G \quad \frac{\langle U, I \rangle \not\models F \quad \langle U, I \rangle \models G}{\langle U, I \rangle \models F \rightarrow G} \]

\[ \langle U, I \rangle \not\models F \quad \frac{\langle U, I \rangle \not\models G}{\langle U, I \rangle \not\models F \rightarrow G} \]

\[ \langle U, I \rangle \models p(s_1, \ldots, s_n) \quad \langle U, I \rangle \not\models p(t_1, \ldots, t_n) \]

\[ \frac{I[s_i] = I[t_i] \text{ for all } i \in [1, n]}{\langle U, I \rangle \models \bot} \]
Soundness and Completeness of Proof Rules

**Soundness:**
If every branch of semantic argument proof derives $\bot$, then $F$ is valid.

**Completeness:**
If $F$ is valid, there exists a finite-length semantic argument proof in which every branch derives $\bot$. 
Undecidability 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
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.
- FOL is very expressive, powerful and undecidable in general

- Some application domains do not need the full power of FOL

- **First-order theories** are useful for reasoning about specific applications
  - e.g., programs with arithmetic operations over integers

- Specialized, efficient decision procedures!
First-Order Theories

Signature $\Sigma_T$: set of constant, function, and predicate symbols
Axioms $A_T$: set of closed formulas over $\Sigma_T$

Axioms provide the meaning of symbols in $\Sigma_T$

$\Sigma_T$-formula: constructed from symbols of $\Sigma_T$, and variables, logical connectives, and quantifiers

$T$-model: a first-order structure $M = \langle U, I \rangle$ such that $M \models A$ for all $A \in A_T$
Satisfiability and Validity Modulo $T$

$F$ is **satisfiable modulo** $T$ iff there exists some $T$-model $M : M \models F$

$F$ is **valid modulo** $T$ (written $T \models F$) iff for all $T$-models $M : M \models F$

The theory $T$ consists of all closed formulas that are valid modulo $T$

- How is validity modulo $T$ different from FOL-validity?
- If a formula is valid in FOL, is it also valid modulo $T$ for any $T$?
- If a formula is valid modulo $T$ for some $T$, is it valid in FOL?
Decidability of a theory

A theory $T$ is decidable iff for every formula $F$, there is an algorithm that:
1. terminates and answers “yes” if $F$ is valid modulo $T$, and
2. terminates and answers “no”, if $F$ is not valid modulo $T$

Next: decidable first-order theories, and theories with decidable fragments
Common first-order theories

- Theory of equality (with uninterpreted functions)
- Peano arithmetic (first-order arithmetic)
- Presburger arithmetic
- Theory of rationals
- Theory of arrays
Theory of equality $T_=$

**Signature**

- $=$ binary predicate, interpreted by axioms
- all constant, function, and predicate symbols

$$\Sigma_:= : \{=, a, b, c, ..., f, g, h, ..., p, q, r\}$$
Theory of equality $T_=$

Axioms

1. $\forall x. x = x$ (reflexivity)
2. $\forall x, y. (x = y) \rightarrow y = x$ (symmetry)
3. $\forall x, y, z. (x = y \land y = z) \rightarrow x = z$ (transitivity)
4. for $n$-ary function symbol $f$, (function congruence)
   $\forall x_1, \ldots, x_n, y_1, \ldots, y_n. (\land_i x_i = y_i) \rightarrow (f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n))$
5. for each $n$-ary predicate symbol $p$, (predicate congruence)
   $\forall x_1, \ldots, x_n, y_1, \ldots, y_n. (\land_i x_i = y_i) \rightarrow (((p(x_1, \ldots, x_n) \leftrightarrow p(y_1, \ldots, y_n)))$
Decidability results for $T_\equiv$

- $T_\equiv$ is undecidable
- Quantifier-free fragment of $T_\equiv$ is (efficiently) decidable
Theories with natural numbers and integers

Natural numbers $\mathbb{N} = \{0,1,2,\ldots\}$

Integers $\mathbb{Z} = \{\ldots,-2,-1,0,1,2,\ldots\}$

Peano arithmetic $T_{PA}$: natural numbers with addition and multiplication

Presburger arithmetic $T_{\mathbb{N}}$: natural numbers with addition

Theory of integers $T_{\mathbb{Z}}$: integers with $+, -, >$
Peano arithmetic $T_{PA}$

**Signature**

- 0, 1 constants
- $+$, . binary functions
- $=$ binary predicate

$$\Sigma_{PA} = \{0, 1, +, ., =\}$$
Peano arithmetic $T_{PA}$

**Axioms**
- Includes equality axioms: reflexivity, symmetry, transitivity
- In addition:
  1. $\forall x. \neg(x + 1 = 0)$ (zero)
  2. $\forall x. x + 0 = x$ (plus zero)
  3. $\forall x. x.0 = 0$ (times zero)
  4. $\forall x, y. (x + 1 = y + 1) \rightarrow x = y$ (successor)
  5. $\forall x, y. x + (y + 1) = (x + y) + 1$ (plus successor)
  6. $\forall x, y. x.(y + 1) = (x.y) + x$ (times successor)
  7. $(F[0] \land (\forall x. F[x] \rightarrow F[x + 1])) \rightarrow \forall x. F[x]$ (induction)

Can we express $<, \leq, >, \geq$ in $T_{PA}$?
Decidability and completeness results for $T_{PA}$

Validity in $T_{PA}$ is undecidable

Validity in quantifier-free fragment of $T_{PA}$ is also undecidable [Matiyasevitch, 1970]

$T_{PA}$ does not capture true arithmetic [Gödel]

There are valid propositions of number theory that are not valid in $T_{PA}$

Drop multiplication to get decidability and completeness!
Presburger arithmetic $T^\mathbb{N}$

**Signature**

- 0, 1 constants
- + binary function
- = binary predicate

$$\Sigma^\mathbb{N} = \{0, 1, +, =\}$$
Presburger arithmetic $T_N$

**Axioms**

- Includes equality axioms: reflexivity, symmetry, transitivity
- In addition:
  1. $\forall x. \neg(x + 1 = 0)$  
     (zero)
  2. $\forall x. x + 0 = x$  
     (plus zero)
  3. $\forall x, y. (x + 1 = y + 1) \rightarrow x = y$  
     (successor)
  4. $\forall x, y. x + (y + 1) = (x + y) + 1$  
     (plus successor)
  5. $(F[0] \land (\forall x. F[x] \rightarrow F[x + 1])) \rightarrow \forall x. F[x]$  
     (induction)
Decidability and completeness results for $T_\mathbb{N}$

Validity in quantifier-free fragment of $T_\mathbb{N}$ is decidable (coNP-complete)

Validity in $T_\mathbb{N}$ is also decidable [Presburger, 1929]

$T_\mathbb{N}$ is also complete: for every closed formula $F$, $T_\mathbb{N} \models F$ or $T_\mathbb{N} \models \neg F$

$T_\mathbb{N}$ admits quantifier elimination:
for every formula $F$, there exists an equivalent quantifier-free formula $F'$
Theory of integers $T_{\mathbb{Z}}$

**Signature**
- $\ldots, -2, -1, 0, 1, 2, \ldots$ constants
- $\ldots, -3 \cdot, -2 \cdot, 2 \cdot, 3 \cdot, \ldots$ unary functions
- $+, -, \ =, \ >$ binary functions
- $=, >$ binary predicates

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

- Also referred to as the theory of linear arithmetic over integers
- Equivalent in expressiveness to Presburger arithmetic
- More convenient notation
Theory of rationals $T_\mathbb{Q}$

Signature

- 0, 1 constants
- + binary function
- $=, \geq$ binary predicates

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

Can we express $>$ in $T_\mathbb{Q}$?
Theory of rationals $T\mathbb{Q}$

Too many axioms, won’t discuss.

Every formula valid in $T\mathbb{Z}$ is valid in $T\mathbb{Q}$, but not vice versa
Decidability results for $T_\mathbb{Q}$

Validity in $T_\mathbb{Q}$ is decidable

Validity in conjunctive quantifier-free fragment of $T_\mathbb{N}$ is (efficiently) decidable
Theory of arrays $T_A$

Signature

- $a[i]$ binary function “read($a, i$)"
- $a\langle i \triangleright v \rangle$ ternary function “write($a, i, v$)"

\[
\Sigma_A = \{ [\cdot], [\cdot\triangleright\cdot], = \}
\]
Theory of arrays $T_A$

Axioms

- Includes equality axioms: reflexivity, symmetry, transitivity
- In addition:
  1. $\forall a, i, j. \ (i = j) \rightarrow a[i] = a[j]$
      (array congruence)
  2. $\forall a, v, i, j. \ (i = j) \rightarrow a(i < v)[j] = v$
      (read-over-write 1)
  3. $\forall a, v, i, j. \ (i \neq j) \rightarrow a(i < v)[j] = a[j]$
      (read-over-write 2)
Decidability results for $T_A$

Validity in $T_A$ is not decidable

Quantifier-free fragment of $T_A$ is decidable
Combination of Theories

Given theories $T_1$ and $T_2$ that have the $=$ predicate, define combined theory $T_1 \cup T_2$:

**Signature** $\Sigma_1 \cup \Sigma_2$

**Axioms** $A_1 \cup A_2$
Decision procedures for combined theories

If
1. quantifier-free fragment of $T_1$ is decidable
2. quantifier-free fragment of $T_2$ is decidable
3. and $T_1$ and $T_2$ meet certain technical requirements
then quantifier-free fragment of of $T_1 \cup T_2$ is also decidable.
[Nelson and Oppen]
Summary

- Review: propositional logic
- Syntax and semantics of first-order logic (FOL)
- Semantic argument method for FOL validity
- Properties of FOL
- Overview of first-order theories