Overview

- So far: Automated reasoning in propositional logic.
- Propositional logic is simple and easy to automate, but not very expressive
- Today: First order logic, also known as relational logic, predicate logic, or first-order predicate calculus
- Much richer and more expressive, but does not admit completely automated reasoning (more on this later)

The Plan

- Syntax and informal semantics (review, today’s lecture)
- Formal semantics, model theory (today’s lecture)
- Semantic argument method for FOL and properties (next lecture)
- Unification, clausal form (third lecture)
- Resolution and first-order theorem proving (fourth lecture)

Constants in First-Order Logic

- In propositional logic, we had two constants $\top$ and $\bot$
- In first order logic, constants are more involved.
- Three kinds of constants:
  1. object constants
  2. function constants
  3. relation constants

Object Constants

- **Object constants** refer to objects in a universe of discourse.
- Objects can be anything we want to say something about: integers, people, real numbers, geometric shapes, . . .
- **Example**: If our universe of discourse is people, object constants can be *jack*, *jane*, *joe*, . . .
- As a convention, we will use letters starting with a – t or digits to denote object constants.
- Example: *a*, *art*, *beth*, 1 etc. refer to object constants.

Function Constants

- **Function constants** refer to functions
- Examples: *mother*, *age*, *plus*, *times*, . . .
- Each function constant has an associated **arity** indicating its number of arguments
- Example: *mother* has arity 1 (unary), *times* has arity 2 (binary) etc.
- An object constant is really a special case of a function constant with arity 0
Relation Constants

- Relation constants refer to relations between or properties of objects
- Example: loves, betterthan, ishappy, ...
- Each relation constant also has an associated arity
- Example: loves has arity 2, ishappy has arity 1 etc.

Formulas

- Formulas of \( L(C, F, R) \) are formed using the terms of this language, relation constants \( R \), logical connectives \( \neg, \land, \lor, \ldots \), and quantifiers \( \forall, \exists \).
- Atomic formula (predicate): Expression \( p(t_1, \ldots, t_n) \) where \( p \in R \) (of arity \( n \)), and \( t_1, \ldots, t_n \) are terms of \( L(C, F, R) \)
- If \( F_1 \) and \( F_2 \) are formulas, then so is \( F_1 \ast F_2 \) where \( \ast \) is any binary connective
- If \( F \) is a formula, then so is \( \neg F \)
- If \( F \) is a formula and \( x \) a variable, so are \( \forall x . F \) (asserts facts about all objects) and \( \exists x . F \) (asserts facts about some object)

Terms

- A set of object, function, and relation constants \( C, F, R \) specifies a first-order language, written \( L(C, F, R) \)
- \( C, F, R \) form the signature of the language.
- Terms \( t \) for a first order language are generated using \( C, F \)
- Basic terms: Any object constant in \( C \) and variables, denoted \( x, y, z, \ldots \)
- Composite terms: \( f(t_1, \ldots, t_k) \) where \( f \in F \) is function of arity \( k \), and \( t_1, \ldots, t_k \) are terms
- Examples: mary, \( x \), sister(mary), price(x, macys), age(mother(y)), ...

Important Reminder

- Predicates (e.g., \( p(x) \)) and function terms (e.g., \( f(x) \)) look similar, but they are very different!
- Function terms can be nested within each other and inside relation constants: \( f(f(x)), p(f(x)) \ldots \)
- Predicates such as \( p(x) \) cannot be nested within function terms or other predicates!
- \( f(p(x)), p(p(x)) \) etc. not valid in FOL!

Quantifiers and Scoping

- The subformula embedded inside a quantifier is called the scope of that quantifier.
- Example: \( \forall y . (\forall x . p(x)) \rightarrow q(x, y) \)
- An occurrence of a variable is bound if it is in the scope of some quantifier.
- An occurrence of a variable is free if it is not in the scope of any quantifiers.

Free vs. Bound Variable Example

- Consider the formula:
\[
\forall y . (\forall x . p(x)) \rightarrow q(x, y)
\]
- Is variable \( y \) bound or free? bound
- Is first occurrence of \( x \) bound or free? bound
- What about second occurrence of \( x \)? free
Closed, Open, and Ground Formulas

- A formula with no free variables is called a **closed** formula.
- A closed formula is also called a **sentence**.
- A formula containing free variables is said to be **open**.
- Example: Is the formula $\forall y.((\forall x.p(x)) \rightarrow q(x, y))$ closed or open? **open**
- Is the formula $\forall y.((\forall x.p(x)) \rightarrow (\exists x.q(x, y)))$ a sentence? **yes**
- A formula is said to be **ground** if it contains no variables.
- Example: $p(a, f(b)) \rightarrow q(c)$ is ground.
- Is $\forall x.p(x)$ ground? **no**

More Friendship Examples

- Given unary relation constants `student` and `atUT` and binary relation constant `friend`, how do we say the following in FOL?
  - “Every student has a friend.”
  - $\forall x.\exists y.\text{friend}(x, y)$
  - “No student has a friend unless he/she is at UT.”
  - $\forall x.((\text{student}(x) \land \neg\text{atUT}(x)) \rightarrow \neg\exists y.\text{friend}(x, y))$

Even More Friendship Examples

- “UT students only have friends at UT.”
  - $\forall x.\forall y.((\text{student}(x) \land \text{atUT}(x) \land \text{friends}(x, y)) \rightarrow \text{atUT}(y))$
  - Another way of saying this:
    - $\forall x.((\text{student}(x) \land \text{atUT}(x)) \rightarrow (\exists y.\text{friends}(x, y) \rightarrow \text{atUT}(y))$
  - These two formulas are actually semantically equivalent

More Friendship Examples

- “UT students are all friends with each other.”
  - $\forall x.\forall y. ((\text{student}(x) \land \text{atUT}(x) \land \text{student}(y) \land \text{atUT}(y)) \rightarrow \text{friends}(x, y))$

Mathematical Theorems in FOL

- **Fermat’s Last Theorem:** No three positive integers $a, b, c$ satisfy the equation $a^n + b^n = c^n$ for any integer $n$ greater than 2.
  - How do we express Fermat’s last theorem in FOL (given a function constant `^`)?
    - $\forall n. (n > 2 \rightarrow \neg\exists a, b, c. (a > 0 \land b > 0 \land c > 0 \land a^n + b^n = c^n))$
One Last Example

- Given binary relation constant `friend`, how do we say this in FOL?
- “Every pair of friends has something in common”
- This cannot be expressed in FOL because it requires quantification over relation constants!
- But it can, however, be expressed in second-order logic:
  \[ \forall x, y. (\text{friend}(x, y) \rightarrow \exists p. p(x) \land p(y)) \]

Semantics of First Order Logic

- In propositional logic, the concepts of interpretation, satisfiability, validity were all straightforward.
- In FOL, these concepts are a bit more involved . . .
- To give semantics to FOL, we need to talk about a universe of discourse (also sometimes called just “universe” or “domain”)

Universe of Discourse

- A universe of discourse is a non-empty set of objects about which we want to say something
- Universe of discourse can be finite, countably infinite, or uncountably infinite, but not empty
- Example universes:
  - Set of non-negative integers: countably infinite
  - Set of real numbers: uncountably infinite
  - The set of suits in playing cards \{♣, ♦, ♥, ♠\}: finite
  - Students in this class: finite

First-Order Interpretations

- An interpretation for a first order language \( I(C, F, R) \) is a mapping \( I \) from \( C, F, R \) to objects in universe \( U \)
- \( I \) maps every \( c \in C \) to some member of \( U \): \( I(c) \in U \)
- \( I \) maps every n-ary function constant \( f \in F \) to an n-ary function \( f^I : U^n \rightarrow U \)
- \( I \) maps every n-ary relation constant \( p \in R \) to an n-ary relation \( p^I \) such that \( p^I \subseteq U^n \)
- Observe: A first-order interpretation does not talk about variables (only constants)

An Example

- Consider the first order language containing object constants \( \{a, b, c\} \), binary function constant \( f \), and ternary relation constant \( r \).
- Universe of discourse: \( U = \{1, 2, 3\} \)
- Possible interpretation \( I \):
  \[
  I(a) = 1, I(b) = 2, I(c) = 2 \\
  I(f) = \{1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 3\} \\
  I(r) = \{(1, 2, 1), (2, 2, 1)\}
  \]
- Observe: Different object constants do not have to map to distinct objects in \( U! \)

Another Example

- Consider the universe of discourse \( U = \{\Box, \Delta\} \)
- For an object constant \( a \), what are the possible interpretations? \( I(a) = \Box \) or \( I(a) = \Delta \)
- For a unary function constant \( f \), what are the possible interpretations?
  \[
  I(f) = \{\Box \rightarrow \Box, \Delta \rightarrow \Box\}, \text{ or } I(f) = \{\Box \rightarrow \Box, \Delta \rightarrow \Delta\}, \text{ or } I(f) = \{\Box \rightarrow \Delta, \Delta \rightarrow \Delta\}
  \]
- For a unary relation constant \( r \), what are all the possible interpretations?
  \[
  I(r) = \{\}, \text{ or } I(r) = \{\Box\}, \text{ or } I(r) = \{\Delta\}, \text{ or } I(r) = \{\Box, \Delta\}\}
  \]
First-Order Structures and Variable Assignments

- A structure $S = \langle U, I \rangle$ for a first-order language consists of a universe of discourse of $U$ and an interpretation $I$.
- This is sometimes also called an algebra.
- A variable assignment $\sigma$ (or assignment) to a FOL formula $\phi$ in a structure $S = \langle U, I \rangle$ is a mapping from variables in $\phi$ to an element of $U$.
- Example: Given $U = \{\square, \triangle\}$, a possible variable assignment for $x$: $\sigma(x) = \triangle$
- Observe: $\sigma$ does not map variables to object constants but to objects in $U$!

Example: Evaluation of Terms

- Consider a first-order language containing object constants $a, b$ and binary function $f$
- Consider universe $\{1, 2\}$ and interpretation $I$:
  $I(a) = 1$  $I(b) = 2$
  $I(f) = \{\langle 1, 1 \rangle \mapsto 2, \langle 1, 2 \rangle \mapsto 2, \langle 2, 1 \rangle \mapsto 1, \langle 2, 2 \rangle \mapsto 1\}$
- Consider variable assignment $\sigma: \{x \mapsto 2, y \mapsto 1\}$
- Under $I$ and $\sigma$, what do these terms evaluate to?
  \[
  f(a, y) = 2 \\
  f(x, b) = 1 \\
  f(f(x, b), f(a, y)) = 2
  \]

Evaluation of Formulas, Bases Cases

- Base case I:
  $U, I, \sigma \models \top$  $U, I, \sigma \models \bot$
- Base case II:
  $U, I, \sigma \models p(t_1, \ldots, t_k)$  if $\langle I, \sigma \rangle(t_1), \ldots, \langle I, \sigma \rangle(t_k) \in I(p)$

Evaluation of Formulas, Notation

- We define evaluation of formula $F$ under structure $S = \langle U, I \rangle$ and variable assignment $\sigma$.
- If $F$ evaluates to true under $U, I, \sigma$, we write $U, I, \sigma \models F$
- If $F$ evaluates to false under $U, I, \sigma$, we write $U, I, \sigma \not\models F$
- We define the semantics of $\models$ inductively.

Example I: Evaluation of Formulas

- Consider a first-order language containing object constants $a, b$ and unary function $f$, and binary relation constant $p$
- Consider universe $\{\ast, \ast\}$ and interpretation $I$:
  $I(a) = \ast$  $I(b) = \ast$
  $I(f) = \{\ast \mapsto \ast, \ast \mapsto \ast\}$
  $I(p) = \{\ast, \ast\}$
- Consider variable assignment $\sigma: \{x \mapsto \ast\}$
- Under $U, I$ and $\sigma$, what do these formulas evaluate to?
  \[
  p(f(b), f(x)) = \text{false} \\
  p(f(x), f(b)) = \text{true} \\
  p(a, f(x)) = \text{false}
  \]
Evaluation of Formulas II

- Inductive semantics for boolean connectives:

  \[ U, I, \sigma \models \neg F \text{ iff } U, I, \sigma \not\models F \]
  \[ U, I, \sigma \models F_1 \land F_2 \text{ iff } U, I, \sigma \models F_1 \text{ and } U, I, \sigma \models F_2 \]
  \[ U, I, \sigma \models F_1 \lor F_2 \text{ iff } U, I, \sigma \models F_1 \text{ or } U, I, \sigma \models F_2 \]
  \[ U, I, \sigma \models F_1 \rightarrow F_2 \text{ iff } U, I, \sigma \not\models F_1 \rightarrow F_2 \]
  \[ U, I, \sigma \models F_1 \leftrightarrow F_2 \text{ iff } U, I, \sigma \not\models F_1 \leftrightarrow F_2 \]

Example II: Evaluation of Formulas

- Consider universe \{•, ⋆\}, variable assignment \(\sigma: \{x \mapsto ⋆\}\), and interpretation \(I\):

  \[ I(a) = ⋆, I(b) = • \]
  \[ I(f) = \{⋆ \mapsto •, ⋆ \mapsto ⋆\} \]
  \[ I(p) = \{\langle•, ⋆\rangle, \langle•, •\rangle\} \]

  Under \(U, I\) and \(\sigma\), what do these formulas evaluate to?

  \[ p(f(b), f(x)) \rightarrow p(f(x), f(b)) = \text{true} \]
  \[ p(f(x), f(b)) \rightarrow p(f(b), f(x)) = \text{false} \]

Variant of Variable Assignment

- We still need to evaluate formulas containing quantifiers!

- But to do that, we first define an \(x\)-variant of a variable assignment.

- An \(x\)-variant of assignment \(\sigma\), written \(\sigma[x \mapsto c]\), is the assignment that agrees with \(\sigma\) for assignments to all variables except \(x\) and assigns \(x\) to \(c\).

- Example: If \(\sigma: \{x \mapsto 1, y \mapsto 2\}\), what is \(\sigma[x \mapsto 3]\)?

  \(\sigma: \{x \mapsto 3, y \mapsto 2\}\)

Example III: Evaluation of Formulas

- Consider universe \{•, ⋆\}, variable assignment \(\sigma: \{x \mapsto ⋆\}\), and interpretation \(I\):

  \[ I(a) = ⋆, I(b) = • \]
  \[ I(f) = \{⋆ \mapsto •, ⋆ \mapsto ⋆\} \]
  \[ I(p) = \{\langle•, ⋆\rangle, \langle•, •\rangle\} \]

- Under \(U, I\) and \(\sigma\), what do these formulas evaluate to?

  \[ \forall x. p(x, a) = \text{false} \]
  \[ \forall x. p(b, x) = \text{true} \]
  \[ \exists x. p(a, x) = \text{false} \]
  \[ \forall x. (p(a, x) \rightarrow p(b, x)) = \text{true} \]
  \[ \exists x. (p(f(x), f(x)) \rightarrow p(x, x)) = \text{true} \]

Satisfiability and Validity of First-Order Formulas

- A first-order formula \(F\) is satisfiable iff there exists a structure \(S\) and variable assignment \(\sigma\) such that

  \[ S, \sigma \models F \]

  Otherwise, \(F\) is unsatisfiable.

- A structure \(S\) is a model of \(F\), written \(S \models F\), if for all variable assignments \(\sigma \in X \rightarrow U\), \(S, \sigma \models F\).

- A first-order formula \(F\) is valid, written \(\models F\) if for all structures \(S\), \(S, \sigma \models F\)
Satisfiability and Validity Examples

- Is the formula $\forall x.\exists y.p(x, y)$ satisfiable? yes
- Satisfying interpretation: $U = \{\star\}$, $I(p) = \{\langle \star, \star \rangle\}$
- Is this formula valid? no
- Falsifying interpretation: $U = \{\star\}$, $I(p) = \{\}$
- Is the formula $\forall x.(p(x, x) \rightarrow \exists y.p(x, y))$ valid? yes
- Intuition: Consider any object $o$. If $p(o, o)$ is false, then implication satisfied. If $p(o, o)$ is true, there exists a $y$ (namely $o$) s.t. $p(x, y)$ is also true.

More Satisfiability and Validity Examples

- Is the formula $(\exists x.p(x)) \rightarrow p(x)$ sat, unsat, or valid? sat
- Satisfying $U, I, \sigma$: $U = \{\star, o\}$, $I(p) = \{\}$, $\sigma(x) = o$
- Falsifying interpretation: $U = \{\star\}$, $I(p) = \{\langle \star \rangle\}$, $\sigma(x) = o$
- Is the formula $(\forall x.p(x)) \rightarrow p(x)$ sat, unsat, or valid? valid
- What about $(\forall x.(p(x) \rightarrow q(x))) \rightarrow (\exists x.(p(x) \land q(x)))$? sat
- Satisfying interpretation: $U = \{\star\}$, $I(p) = \{\langle \star \rangle\}$, $I(q) = \{\langle \star \rangle\}$
- Falsifying interpretation: $U = \{\star\}$, $I(p) = \{\}$, $I(q) = \{\langle \star \rangle\}$

Understanding Models

- Recall: A structure $S$ is a model of a formula if for all $\sigma$, $S, \sigma \models F$
- Consider a formula $F$ such that $S, \sigma \models F$. Is $S$ a model $F$? not necessarily
- Consider a sentence $F$ such that $S, \sigma \models F$. Is $S$ a model $F$? yes
- Consider a ground formula $F$ such that $S, \sigma \models F$. Is $S$ a model of $F$? yes

Summary

- Today: Syntax and formal semantics of FOL
- Next lecture:
  - Semantic argument method for FOL
  - Properties of first-order logic: decidability results, compactness