## 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, three constants:
  1. object constants
  2. function constants
  3. relation constants

## Object Constants

- **Object constants** refer to objects in a universe of discourse.

- **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: `motherOf`, `ageOf`, `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)

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.

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!

Free vs. Bound Variable Example

- Consider the formula: $\forall y.((\forall x.p(x)) \rightarrow q(x, y))$
- Is variable $y$ bound or free?
- Is first occurrence of $x$ bound or free?
- What about second occurrence of $x$?
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?
- Is the formula \( \forall y.((\forall x.p(x)) \rightarrow (\exists x.q(x, y))) \) a sentence?
- 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?

FOL example #1

- 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.
- Assuming universe is integers, how do we express this theorem in FOL using function constant \( ^\cdot \) and relation constants \( >, = \)?

FOL example #2

- Consider the axiom schema of unrestricted comprehension in naive set theory:
  “There exists a set whose members are precisely those objects that satisfy predicate \( P \)”
- Using predicates \( IsSet, \in, P \), express this in FOL

FOL Example #3

- Consider the statement “CS389L is taken only by those students who do not take CS388L”
- Express this sentence in FOL using binary relation constant \( takes \), and unary relation constant \( student \)

One Last Example

- Given binary relation friend, how do we say this in FOL?
- “Every pair of friends has something in common”

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
  - Set of real numbers
  - The set of suits in playing cards
  - Students in this class

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 \( U = \{\Box, \triangle\} \), a possible variable assignment for \( x \): \( \sigma(x) = \triangle \)

Evaluation of Terms

- We define how to evaluate a term \( t \) under interpretation \( I \) and assignment \( \sigma \), written \( \langle I, \sigma \rangle(t) \)
  - Object constants: \( \langle I, \sigma \rangle(a) = I(a) \)
  - Variable terms: \( \langle I, \sigma \rangle(v) = \sigma(v) \)
  - Function terms:
    \[ \langle I, \sigma \rangle(f(t_1, \ldots, t_k)) = I(f)(\langle I, \sigma \rangle(t_1), \ldots, \langle I, \sigma \rangle(t_k)) \]
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 \( U, I, \sigma \not\models F \).

Example I: Evaluation of Formulas

Consider a first-order language containing object constants \( a, b \) and unary function \( f \).

\[
\begin{align*}
I(a) & = 1 \\
I(b) & = 2 \\
I(f) & = \{ (1, 1) \mapsto 2, (1, 2) \mapsto 2, (2, 1) \mapsto 1, (2, 2) \mapsto 1 \}
\end{align*}
\]

Consider variable assignment \( \sigma : \{ x \mapsto 2, y \mapsto 1 \} \).

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

\[
\begin{align*}
f(a, y) & = \\
f(x, b) & = \\
f(f(x, b), f(a, y)) & =
\end{align*}
\]

Evaluation of Formulas II

- Inductive semantics for boolean connectives:

\[
\begin{align*}
U, I, \sigma \models \neg F & \iff U, I, \sigma \not\models F \\
U, I, \sigma \models F_1 \land F_2 & \iff U, I, \sigma \models F_1 \text{ and } U, I, \sigma \models F_2 \\
U, I, \sigma \models F_1 \lor F_2 & \iff U, I, \sigma \models F_1 \text{ or } U, I, \sigma \models F_2 \\
U, I, \sigma \models F_1 \rightarrow F_2 & \iff U, I, \sigma \not\models F_1 \text{ or } U, I, \sigma \models F_2 \\
U, I, \sigma \models F_1 \leftrightarrow F_2 & \iff U, I, \sigma \models F_1 \text{ and } U, I, \sigma \models F_2, \\
& \quad \text{or } U, I, \sigma \not\models F_1 \text{ and } I, \sigma \not\models F_2
\end{align*}
\]

Example: Evaluation of Terms

Consider a first-order language containing object constants \( a, b \) and binary function \( f \).

\[
\begin{align*}
\text{Example: If } & \\
\text{Variation of Variable Assignment:} &
\end{align*}
\]

- 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 \( \sigma[x \mapsto 3] \)?
We can now give semantics to quantifiers:

- **Universal quantifier:**
  \[ U, I, \sigma \models \forall x.F \text{ iff for all } o \in U, U, I, \sigma[x \mapsto o] \models F \]

- **Existential quantifier:**
  \[ U, I, \sigma \models \exists x.F \text{ iff there exists } o \in U \text{ s.t. } U, I, \sigma[x \mapsto o] \models F \]

**Example III: Evaluation of Formulas**

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

\[
\begin{align*}
I(a) &= \star \\
I(b) &= \bullet \\
I(f) &= \{ \star \mapsto \bullet, \bullet \mapsto \star \} \\
I(p) &= \{ (\bullet, \star), (\bullet, \bullet) \}
\end{align*}
\]

- Under \( U, I \) and \( \sigma \), what do these formulas evaluate to?
  
  \[
  \begin{align*}
  \forall x.p(x, a) &= \text{false} \\
  \forall x.p(b, x) &= \text{false} \\
  \exists x.p(a, x) &= \text{true} \\
  \exists x.(p(a, x) \rightarrow p(b, x)) &= \text{false} \\
  \exists x.(p(f(x), f(x)) \rightarrow p(x, x)) &= \text{false}
  \end{align*}
  \]

**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 of \( F \)?
  
  Consider a sentence \( F \) such that \( S, \sigma \models F \). Is \( S \) a model of \( F \)?
  
  Consider a ground formula \( F \) such that \( S, \sigma \models F \). Is \( S \) a model of \( F \)?
Summary

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