Automated Reasoning

Lecture 5: First-Order Logic

Jacques Fleuriot
jdf@inf.ac.uk

Recap

- Over the last three lectures, we have looked at:
  - Propositional logic, semantics and proof systems
  - Doing propositional logic proofs in Isabelle
- Today:
  - Syntax and Semantics of First-Order Logic (FOL)
  - Natural Deduction rules for FOL
  - Doing FOL proofs in Isabelle

Problem

Consider the following problem:

1. If someone cheats then everyone loses the game.
2. If everyone who cheats also loses, then I lose the game.
3. Did I lose the game?

Is **Propositional Logic** rich enough to formally represent and reason about this problem?

The finer logical structure of this problem would not be captured by the constructs we have so far encountered.

We need a richer language!

A Richer Language

First-order (predicate) logic (FOL) extends propositional logic:

- Atomic formulas are assertions about *properties* of *individual(s)*.
  e.g. an individual might have the property of being a cheat.
- We can use *variables* to denote arbitrary individuals.
  e.g. x is a cheater.
- We can *bind* variables with quantifiers $\forall$ (for all) and $\exists$ (exists).
  e.g. for all x, x is a cheater.
- We can use connectives to compose formulas:
  e.g. for all x, if x is a cheater then x loses.
- We can use quantifiers on subformulas.
  e.g. we can formally distinguish between: ”if *anyone* cheats we lose the game” and ”if *everyone* cheats, we lose the game”.

Terms of First-Order Logic

Given:

- a countably infinite set of (individual) variables $\mathcal{V} = \{x, y, z, \ldots\}$;
- a finite or countably infinite set of function letters $\mathcal{F}$ each assigned a unique arity (possibly 0)

then the set of (well-formed) terms is the smallest set such that

- any variable $v \in \mathcal{V}$ is a term;
- if $f \in \mathcal{F}$ has arity n, and $t_1, \ldots t_n$ are terms, so is $f(t_1, \ldots, t_n)$.

Remarks

- If f has arity 0, we usually write f rather than $f()$, and call f a **constant**
- In practice, we use infix notation when appropriate: e.g., $x + y$ instead of $+(x, y)$.

Formulas of First-Order Logic

Given a countably infinite set of predicates $\mathcal{P}$, each assigned a unique arity (possibly 0), the set of wffs is the smallest set such that

- if $\mathrm{A} \in \mathcal{P}$ has arity n, and $t_1, \ldots t_n$ are terms, then $\mathrm{A}(t_1, \ldots, t_n)$ is a wff;
- if P and Q are wffs, so are $\neg P$, $P \vee Q$, $P \wedge Q$, $P \rightarrow Q$, $P \leftrightarrow Q$,
- if P is a wff, so are $\exists x.\, P$ and $\forall x.\, P$ for any $x \in \mathcal{V}$;
- if P is a wff, then (P) is a wff.

Remarks

- If A has arity 0, we usually write A rather than $\mathrm{A}()$, and call A a **propositional variable**. This way, propositional logic wffs look like a subset of FOL wffs. Also, use infix notation where appropriate.
- We assume $\exists x$ and $\forall x$ bind more weakly than any of the propositional connectives.
  $\exists x.P \wedge Q$ is $\exists x.(P \wedge Q)$, not $(\exists x.P) \wedge Q$.
  (NB: H&R assume $\exists x$ and $\forall x$ bind like $\neg$.)

Example: Problem Revisited

We can now formally represent our problem in FOL:

- **Assumption 1:** If someone cheats then everyone loses the game: $(\exists x.\, \mathrm{Cheats}(x)) \to \forall x.\, \mathrm{Loses}(x)$.
- **Assumption 2:** If everyone who cheats also loses, then I lose the game : $(\forall x.\, \mathrm{Cheats}(x) \to \mathrm{Loses}(x)) \to \mathrm{Loses}(\mathit{me})$.

To answer the question *Did I lose the game?* we need to prove either $\mathrm{Loses}(\mathit{me})$ or $\neg\mathrm{Loses}(\mathit{me})$ from these assumptions.

More on this later.

Free and Bound Variables

- An occurrence of a variable x in a formula P is **bound** if it is in the scope of a $\forall x$ or $\exists x$ quantifier.
- A variable occurrence x is **in the scope of** a quantifier occurrence $\forall x$ or $\exists x$ if the quantifier occurrence is the first occurrence of a quantifier over x in a traversal from the variable occurrence position to the root of the formula tree.
- If a variable occurrence is **not** bound, it is **free**

Example

In

$$P(x) \land (\forall x.\ P(y) \rightarrow P(x))$$

The first occurrence of x and the occurrence of y are free, while the second occurrence of x is bound.

Free and Bound Variables

- An occurrence of a variable x in a formula P is **bound** if it is in the scope of a $\forall x$ or $\exists x$ quantifier.
- A variable occurrence x is **in the scope of** a quantifier occurrence $\forall x$ or $\exists x$ if the quantifier occurrence is the first occurrence of a quantifier over x in a traversal from the variable occurrence position to the root of the formula tree.
- If a variable occurrence is **not** bound, it is **free**

Example

In

$$P(\overset{\text{free}}{x}) \land (\forall x.\ P(y) \rightarrow P(x))$$

The first occurrence of x and the occurrence of y are free, while the second occurrence of x is bound.

Free and Bound Variables

- An occurrence of a variable x in a formula P is **bound** if it is in the scope of a $\forall x$ or $\exists x$ quantifier.
- A variable occurrence x is **in the scope of** a quantifier occurrence $\forall x$ or $\exists x$ if the quantifier occurrence is the first occurrence of a quantifier over x in a traversal from the variable occurrence position to the root of the formula tree.
- If a variable occurrence is **not** bound, it is **free**

Example

In

$$P(\overset{\text{free}}{\boxed{x}}) \land (\forall x.\, P(\overset{\text{free}}{\boxed{y}}) \to P(x))$$

The first occurrence of x and the occurrence of y are free, while the second occurrence of x is bound.

Free and Bound Variables

- An occurrence of a variable x in a formula P is **bound** if it is in the scope of a $\forall x$ or $\exists x$ quantifier.
- A variable occurrence x is **in the scope of** a quantifier occurrence $\forall x$ or $\exists x$ if the quantifier occurrence is the first occurrence of a quantifier over x in a traversal from the variable occurrence position to the root of the formula tree.
- If a variable occurrence is **not** bound, it is **free**

Example

In

$$P(\overset{\text{free}}{x}) \land (\forall x.\, P(\overset{\text{free}}{y}) \to P(\overset{\text{bound}}{x}))$$

The first occurrence of x and the occurrence of y are free, while the second occurrence of x is bound.

Substitution Rules

If P is a formula, s is a term and x is a variable, then

$$P[s/x]$$

is the formula obtained by **substituting s for all free occurrences of x** throughout P.

Example

$$\begin{array}{lcl}(\exists x.\, P(x,y))\,[3/y] & \equiv & \exists x.\, P(x,3) \\ (\exists x.\, P(x,y))\,[2/x] & \equiv & \exists x.\, P(x,y).\end{array}$$

If necessary, bound variables in P must be **renamed** to avoid capture of free variables in s.

$$(\exists x.\, P(x,y))\,[f(x)/y] = \exists z.\, P(z, f(x))$$

Semantics of First-Order Logic Formulas

(Recall that an *interpretation* for propositional logic maps atomic propositions to truth values.)

Informally, an **interpretation** of a formula maps its function letters to actual functions, and its predicate symbols to actual predicates.

The interpretation also **specifies some domain of discourse** $\mathcal{D}$ (a non-empty set or universe) on which the functions and relations are defined.

A formal definition requires some work!

Semantics of FOL Formulas (II)

Definition (Interpretation)

Let $\mathcal{F}$ be a set of function symbols and $\mathcal{P}$ be a set of predicate symbols.
An **interpretation** $\mathcal{I}$ consists of a **non-empty set** $\mathcal{D}$ of concrete values, called the domain of the intepretation, together with the following mappings

1. each **predicate symbol** $P \in \mathcal{P}$ of arity $n > 0$ is assigned to a subset $P^{\mathcal{I}} \subseteq D^n$ of n-tuples of D. Each **nullary predicate** is assigned either **T** or **F**.
2. Each **function symbol** $f \in \mathcal{F}$ of arity $n > 0$ is assigned to a concrete function $f^{\mathcal{I}} : \mathcal{D}^n \to \mathcal{D}$. Each **nullary function (constant)** is assigned to a concrete value in $\mathcal{D}$.

Example of Interpretation

Consider the following statement, containing constant a:

$$P(a) \wedge \exists x.\, Q(a, x) \qquad (*)$$

In one possible interpretation $\mathcal{I}$:

- the domain $\mathcal{D}$ is the set of natural numbers $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$;
- Mappings:
  - 2 to a i.e. $a^{\mathcal{I}} \equiv 2$,
  - the property of being even to P i.e. $P^{\mathcal{I}} \equiv \{0, 2, 4, \ldots\}$, and
  - the relation of being greater than to Q, i.e. the set of pairs $Q^{\mathcal{I}} \equiv \{(1, 0), \ldots, (2, 0), (2, 1), \ldots, (89, 27), \ldots\}$;
- under this interpretation: $(*)$ affirms that 2 is even and there exists a natural number that 2 is greater than. Is $(*)$ satisfied under this interpretation?

Example of Interpretation

Consider the following statement, containing constant a:

$$P(a) \land \exists x.\, Q(a, x) \qquad (*)$$

In one possible interpretation $\mathcal{I}$:

- the domain $\mathcal{D}$ is the set of natural numbers $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$;
- Mappings:
  - 2 to a i.e. $a^{\mathcal{I}} \equiv 2$,
  - the property of being even to P i.e. $P^{\mathcal{I}} \equiv \{0, 2, 4, \ldots\}$, and
  - the relation of being greater than to Q, i.e. the set of pairs $Q^{\mathcal{I}} \equiv \{(1, 0), \ldots, (2, 0), (2, 1), \ldots, (89, 27), \ldots\}$;
- under this interpretation: $(*)$ affirms that 2 is even and there exists a natural number that 2 is greater than. Is $(*)$ satisfied under this interpretation? — Yes.
- Such a satisfying interpretation is sometimes known as a **model**. Note: In H&R, a model is *any* interpretation (so be careful).

Semantics of FOL Formulas (III)

Definition (Assignment)

Given an interpretation $\mathcal{I}$, an *assignment* s assigns a value from the domain $\mathcal{D}$ to each variable in $\mathcal{V}$ i.e. $s : \mathcal{V} \to \mathcal{D}$.

We extend this assignment s to all terms inductively by saying that

1. if $\mathcal{I}$ maps the n-ary function letter f to the function $f^{\mathcal{I}}$, and
2. if terms $t_1, \ldots, t_n$ have been assigned concrete values $a_1, \ldots, a_n \in \mathcal{D}$

then we can assign value $f^{\mathcal{I}}(a_1, \ldots, a_n) \in \mathcal{D}$ to the term $f(t_1, \ldots, t_n)$.

An assignment s of values to **variables** is also commonly known as an **environment** and we denote by $s[x \mapsto a]$ the environment that maps $x \in \mathcal{V}$ to a (and any other variable $y \in \mathcal{V}$ to $s(y)$).

Remark: The interpretation I is fixed before we interpret a formula, but the assignment s will vary as we interpret the quantifiers.

Semantics of FOL Formulas (IV)

Definition (Satisfaction)

Given an interpretation $\mathcal{I}$ and an assignment $s : \mathcal{V} \to \mathcal{D}$

1. any wff which is a nullary predicate letter A is satisfied if and only if the interpretation in $\mathcal{I}$ of A is **T**;
2. suppose we have a wff P of the form $\mathrm{A}(t_1 \ \ldots \ t_n)$, where A is interpreted as relation $A^{\mathcal{I}}$ and $t_1, \ldots, t_n$ have been assigned concrete values $a_1, \ldots, a_n$ by s. Then P is satisfied if and only if $(a_1, \ldots, a_n) \in A^{\mathcal{I}}$;
3. any wff of the form $\forall x.P$ is satisfied if and only if P is satisfied with respect to assignment $s[x \mapsto a]$ for all $a \in \mathcal{D}$;
4. any wff of the form $\exists x.P$ is satisfied if and only if P is satisfied with respect to assignment $s[x \mapsto a]$ for some $a \in \mathcal{D}$;
5. any wffs of the form $P \vee Q$, $P \wedge Q$, $P \to Q$, $P \leftrightarrow Q$, $\neg P$ are satisfied according to the truth-tables for each connective (e.g. $P \vee Q$ is satisfied if and only if P is satisfied or Q is satisfied).

Example: Assignment and Satisfaction

Consider the wff ϕ:

$$R(f(x), g(y, a))$$

where $x, y \in \mathcal{V}$ i.e. are variables and a is a nullary function symbol i.e. a constant.

Given the interpretation $\mathcal{I}$ where

- the domain $\mathcal{D}$ is the set of integers $\mathbb{Z}$
- $a^{\mathcal{I}} \equiv -5$
- $R^{\mathcal{I}} \equiv <$ (less than)
- $f^{\mathcal{I}} \equiv -$ (minus)
- $g^{\mathcal{I}} \equiv +$ (addition)

and the environment $s[x \mapsto 3, y \mapsto 2]$ then under this interpretation and assignment:

Example: Assignment and Satisfaction

Consider the wff ϕ:

$$R(f(x), g(y, a))$$

where $x, y \in \mathcal{V}$ i.e. are variables and a is a nullary function symbol i.e. a constant.

Given the interpretation $\mathcal{I}$ where

- the domain $\mathcal{D}$ is the set of integers $\mathbb{Z}$
- $a^{\mathcal{I}} \equiv -5$
- $R^{\mathcal{I}} \equiv <$ (less than)
- $f^{\mathcal{I}} \equiv -$ (minus)
- $g^{\mathcal{I}} \equiv +$ (addition)

and the environment $s[x \mapsto 3, y \mapsto 2]$ then under this interpretation and assignment:

$$\phi^{\mathcal{I}} \equiv -3 < 2 + -5 = -3 < -3$$

is not satisfied.

Example: Satisfaction & Validity

Consider the following statement:

$$\forall x\ y.\ R(x, y) \rightarrow \exists z. R(x, z) \land R(z, y)$$

1. Is it satisfiable?
2. Is it valid?

Example: Satisfaction & Validity

Consider the following statement:

$$\forall x\ y.\ R(x, y) \to \exists z. R(x, z) \land R(z, y)$$

1. Is it satisfiable?
2. Is it valid?

Answers:

Example: Satisfaction & Validity

Consider the following statement:

$$\forall x\ y.\ R(x, y) \rightarrow \exists z. R(x, z) \wedge R(z, y)$$

1. Is it satisfiable?
2. Is it valid?

Answers:

1. Yes: Domain is the real numbers and R is interpreted as the less-than relation.

Example: Satisfaction & Validity

Consider the following statement:

$$\forall x\ y.\ R(x, y) \rightarrow \exists z. R(x, z) \wedge R(z, y)$$

1. Is it satisfiable?
2. Is it valid?

Answers:

1. Yes: Domain is the real numbers and R is interpreted as the less-than relation.
2. No: Domain? Intepretation for R?

Semantics of FOL Formulas (V)

Definition (Entailment)

We write $I \models_s P$ to mean that wff P is satisfied by interpretation I and assignment s.

We say that the wffs $P_1, P_2, \ldots, P_n$ **entail** wff Q and write

$$P_1, P_2, \ldots, P_n \models Q$$

if, for any interpretation I and assignment s for which $I \models_s P_i$ for all i, we also have $I \models_s Q$.

As with propositional logic, we must ensure that our inference rules are *valid*. That is, if

$$\frac{P_1 \qquad P_2 \qquad \ldots \qquad P_n}{Q}$$

then we must have $P_1, P_2, \ldots, P_n \models Q$.

More Introduction Rules

We now consider the additional natural deduction rules for FOL.

The introduction rules for the quantifiers are:

- Universal quantification: Provided that x_0 is **not** free in the assumptions,

$$\frac{P[x_0/x]}{\forall x.\ P}\ (\text{allI})$$

- Existential quantification:

$$\frac{P[t/x]}{\exists x. P}\ (\text{exI})$$

Existential Elimination

$$\dfrac{\exists x.P \qquad \begin{array}{c}[P[x_0/x]] \\ \vdots \\ Q\end{array}}{Q}\ (\text{exE})$$

Provided x_0 does **not** occur in Q or any assumption other than $P[x_0/x]$ on which the derivation of Q from $P[x_0/x]$ depends.

Universal Elimination

Specialisation rule:

$$\frac{\forall x. P}{P[t/x]} \ (\text{spec})$$

An alternative universal elimination rule is `allE`:

$$\frac{\forall x. P \qquad \begin{array}{c} [P[t/x]] \\ \vdots \\ Q \end{array}}{Q} \ (\text{allE})$$

Example Proof

Prove that $\exists y.\, P(y)$ is true, given that $\forall x.\, P(x)$ holds.

$$\dfrac{\dfrac{\forall x. P(x)}{P(a)}\ \text{(spec)}}{\exists y. P(y)}\ \text{(exI)}$$

Example Proof

Prove that $\exists y.\, P(y)$ is true, given that $\forall x.\, P(x)$ holds.

$$\dfrac{\dfrac{\forall x. P(x)}{P(a)}\ (\text{spec})}{\exists y. P(y)}\ (\text{exI})$$

Side note: semantically, we implicitly use the fact that our domain of discourse is non-empty. It doesn't matter what a is, but we have to have something.

Why the side conditions on `allI` and `exE`?

A (non-)proof of: $\vdash x > 5 \to \forall x.\, x > 5$:

$$\dfrac{\dfrac{\dfrac{}{x > 5 \vdash x > 5}\ \text{(assumption)}}{x > 5 \vdash \forall x.\, x > 5}\ \text{(allI)}}{\vdash x > 5 \to \forall x.\, x > 5}\ \text{(impI)}$$

But it is clearly false that if a particular x is greater than 5, then every x is greater than 5. We have "proven" that $x > 5$, but not for an **arbitrary** x, only for the **particular** x we had already made an assumption about.

Exercise: Give a non-proof for `exE`.

Machine assistance: Isabelle keeps track of which variable names are allowed where, so we can only apply the rules in a sound way.

Example Proof (II)

Prove that $\forall x.\, Q(x)$ is true, given $\forall x.\, P(x)$ and $(\forall x.\, P(x) \rightarrow Q(x))$.

$$
\cfrac{\forall x.P(x) \rightarrow Q(x) \qquad \cfrac{\forall x.P(x) \qquad \cfrac{[P(y) \rightarrow Q(y)]_1 \qquad [P(y)]_2}{Q(y)}}{Q(y)}\;(\mathrm{allE}_2)}{\cfrac{Q(y)}{\forall x.Q(x)}\;(\mathrm{allI})}\;(\mathrm{allE}_1)
$$

Note: Subscripts are attached to rules being applied and the assumption(s) that they introduce e.g. allE_1 and $[P(y) \rightarrow Q(y)]_1$.

Problem (III)

Prove that $Loses(me)$ given that

1. $(\exists x.\, \mathrm{Cheats}(x)) \to \forall x.\, \mathrm{Loses}(x)$
2. $(\forall x.\, \mathrm{Cheats}(x) \to \mathrm{Loses}(x)) \to \mathrm{Loses}(me)$

$$
\dfrac{\textit{assumption2} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\textit{assumption1} \qquad \dfrac{[\mathrm{Cheats}(y)]_1}{\exists x.\ \mathrm{Cheats}(x)}\ (\mathrm{exI})}{\forall x.\ \mathrm{Loses}(x)}\ (\mathrm{mp})}{\mathrm{Loses}(y)}\ (\mathrm{spec})}{\mathrm{Cheats}(y) \to \mathrm{Loses}(y)}\ (\mathrm{implI}_1)}{\forall x.\ \mathrm{Cheats}(x) \to \mathrm{Loses}(x)}\ (\mathrm{allI})}{\mathrm{Loses}(me)}\ (\mathrm{mp})
$$

FOL in Isabelle/HOL

Isabelle's HOL object logic is richer than the FOL so far presented. One difference is that all variables, terms and formulas have **types**.

The type language is built using

- **base types** such as *bool* (the type of truth values) and *nat* (the type of natural numbers).
- **type constructors** such as *list* and *set* which are written postfix, *e.g., nat list* or *nat set.*
- **function types** written using $\Rightarrow$; e.g.

  $$nat \times nat \Rightarrow nat$$

  which is a function taking two arguments of type *nat* and returning an object of type *nat.*
- **type variables** such as $'a$, $'b$ etc. These give rise to polymorphic types such as $'a \Rightarrow 'a$.

FOL in Isabelle/HOL (II)

- Consider the mathematical predicate $a = b \bmod n$. We could formalise this operator as:

  $$\texttt{definition}\ \mathrm{mod}\ ::\ \texttt{"}\mathit{int} \Rightarrow \mathit{int} \Rightarrow \mathit{int} \Rightarrow \mathit{bool}\texttt{"}$$
  $$\texttt{where}\ \texttt{"}\mathrm{mod}\ a\ b\ n \equiv \exists k.\ a = k * n + b\texttt{"}$$

- Isabelle performs **type inference**, allowing us to write:

  $$\forall x\ y\ n.\ \mathrm{mod}\ x\ y\ n \to \mathrm{mod}\ y\ x\ n$$

  instead of

  $$\forall (x :: \mathit{int})\ (y :: \mathit{int})\ (n :: \mathit{int}).\ \mathrm{mod}\ x\ y\ n \to \mathrm{mod}\ y\ x\ n$$

FOL L-System Sequent Calculus Rules

$$\frac{\Gamma \vdash P[x_0/x]}{\Gamma \vdash \forall x.\ P}\ (\text{allI}) \qquad \frac{\Gamma, P[t/x] \vdash Q}{\Gamma, \forall x.P \vdash Q}\ (\text{e allE t}) \qquad \frac{\Gamma, \forall x.P, P[t/x] \vdash Q}{\Gamma, \forall x.P \vdash Q}\ (\text{f spec t})$$

$$\frac{\Gamma \vdash P[t/x]}{\Gamma \vdash \exists x.P}\ (\text{r exI t}) \qquad \frac{\Gamma, P[x_0/x] \vdash Q}{\Gamma, \exists x.P \vdash Q}\ (\text{e exE}) \qquad \frac{\Gamma, \forall x.\neg P \vdash \bot}{\Gamma \vdash \exists x.P}\ (\text{exCIF})$$

- Rule prefixes: e = erule, f = frule, r = rule
- x_0 is some variable **not** free in hypotheses or conclusion. Isabelle automatically picks fresh names (to ensure soundness!)
- When t suffix is used above (e.g., as in "e allE t"), then the term t can be explicitly specified in Isabelle method using a variant of the existing method. *e.g.*, `apply (erule_tac x="t" in allE)`.
- Rule `exCIF` is a variation on the standard Isabelle rule `exCI` introduced in the `FOL.thy` file on the course webpage. It does not exist as an explicit Isabelle inference rule but can be derived (see `FOL.thy`).

Example II as a FOL Sequent Proof

$$
\dfrac{\dfrac{\dfrac{\dfrac{\overline{P(y) \vdash P(y)} \qquad \overline{P(y), Q(y) \vdash Q(y)}}{P(y) \to Q(y), P(y) \vdash Q(y)}\ \text{(e impE)}}{P(y) \to Q(y), \forall x.\ P(x) \vdash Q(y)}\ \text{(e allE y)}}{\forall x.\ P(x) \to Q(x), \forall x.\ P(x) \vdash Q(y)}\ \text{(e allE y)}}{\forall x.\ P(x) \to Q(x), \forall x.\ P(x) \vdash \forall x.\ Q(x)}\ \text{(allI)}
$$

Summary

- Introduction to First-Order Logic (H&R 2.1-2.4)
  - Syntax and Semantics
  - Substitution
  - Natural Deduction rules for quantifiers
- Isabelle and First-Order Logic
  - Defining predicates
  - A brief look at types
  - Try `FOL.thy` on the course webpage in Isabelle.