Lecture 10 - More first-order logic

Vaishnavi Sundararajan

COL703/COL7203 - Logic for Computer Science

Recap: FOL Syntax

- We have a countable set of variables $x, y, z \ldots \in \mathscr{V}$
- We have a countable set of function symbols $f, g, h \ldots \in \mathscr{F}$, and a countable set of relation/predicate symbols $P, Q, R \ldots \in \mathscr{P}$
- 0-ary function symbols are constant symbols in $\mathscr{C}$
- $(\mathscr{C}, \mathscr{F}, \mathscr{P})$ is a signature Σ
- Grammar for FOL is as follows

  $\varphi, \psi \coloneqq t_1 \equiv t_2 \mid P(t_1, \ldots, t_n) \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \supset \psi \mid \exists x.\ [\varphi] \mid \forall x.\ [\varphi]$

  where P is an n-ary predicate symbol in Σ, and the term syntax is

  $$t \coloneqq x \in \mathscr{V} \mid c \in \mathscr{C} \mid f(t_1, \ldots, t_m)$$

  where f is an m-ary function symbol in Σ.

Example: Arithmetic over $+$ and $*$

- Constants: $\mathscr{C} = \{0\}$
- Functions: $\mathscr{F} = \{\mathrm{nxt}/1, (+)/2, (*)/2\}$
- Predicates: $\mathscr{P} = \emptyset$
- Adding 0 to any number does not change it

Example: Arithmetic over $+$ and $*$

- Constants: $\mathscr{C} = \{0\}$
- Functions: $\mathscr{F} = \{\mathsf{nxt}/1, (+)/2, (*)/2\}$
- Predicates: $\mathscr{P} = \emptyset$
- Adding 0 to any number does not change it: $\forall x.\ [x + 0 \equiv x]$
- Multiplying any number by 0 yields 0

Example: Arithmetic over $+$ and $*$

- Constants: $\mathscr{C} = \{0\}$
- Functions: $\mathscr{F} = \{\mathrm{nxt}/1, (+)/2, (*)/2\}$
- Predicates: $\mathscr{P} = \emptyset$
- Adding 0 to any number does not change it: $\forall x.\ [x + 0 \equiv x]$
- Multiplying any number by 0 yields 0: $\forall x.\ [x * 0 \equiv 0]$
- No number is its own successor

Example: Arithmetic over $+$ and $*$

- Constants: $\mathscr{C} = \{0\}$
- Functions: $\mathscr{F} = \{\mathsf{nxt}/1, (+)/2, (*)/2\}$
- Predicates: $\mathscr{P} = \emptyset$
- Adding 0 to any number does not change it: $\forall x.\ [x + 0 \equiv x]$
- Multiplying any number by 0 yields 0: $\forall x.\ [x * 0 \equiv 0]$
- No number is its own successor: $\forall x.\ [\neg(x \equiv \mathsf{nxt}(x))]$
- Every number is either 0 or the successor of some other number

Example: Arithmetic over $+$ and $*$

- Constants: $\mathscr{C} = \{0\}$
- Functions: $\mathscr{F} = \{\mathsf{nxt}/1, (+)/2, (*)/2\}$
- Predicates: $\mathscr{P} = \emptyset$
- Adding 0 to any number does not change it: $\forall x.\ [x + 0 \equiv x]$
- Multiplying any number by 0 yields 0: $\forall x.\ [x * 0 \equiv 0]$
- No number is its own successor: $\forall x.\ [\neg(x \equiv \mathsf{nxt}(x))]$
- Every number is either 0 or the successor of some other number: $\forall x.\ \left[x \equiv 0 \vee \left\{\exists y.\ [x \equiv \mathsf{nxt}(y)]\right\}\right]$

FOL: Expressions

- Grammar for generating the language FO_Σ is as follows
$$\varphi, \psi \coloneqq t_1 \equiv t_2 \;\Big|\; P(t_1, \dots, t_n) \;\Big|\; \neg\varphi \;\Big|\; \varphi \wedge \psi \;\Big|\; \varphi \vee \psi \;\Big|\; \varphi \supset \psi \;\Big|\; \exists x.\ [\varphi] \;\Big|\; \forall x.\ [\varphi]$$
where P is an n-ary predicate symbol in Σ, and the term syntax is
$$t \coloneqq x \in \mathscr{V} \mid c \in \mathscr{C} \mid f(t_1, \dots, t_m)$$
- Can write Abstract Syntax Trees (ASTs) for FO expressions as well
- **Main connective** labels the root of the AST; likely a quantifier!
- Define the **set of subformulae** of φ (denoted $\mathsf{sf}(\varphi)$) as follows
  - $\mathsf{sf}(\varphi) = \{\varphi\}$, if φ of the form $t_1 \equiv t_2$ or $P(t_1, \dots, t_n)$
  - $\mathsf{sf}(\neg\varphi) = \{\neg\varphi\} \cup \mathsf{sf}(\varphi)$
  - $\mathsf{sf}(\varphi \circ \psi) = \{\varphi \circ \psi\} \cup \mathsf{sf}(\varphi) \cup \mathsf{sf}(\psi)$, for $\circ \in \{\wedge, \vee, \supset\}$
  - $\mathsf{sf}(\forall x.\ [\varphi]) = \{\forall x.\ [\varphi]\} \cup \mathsf{sf}(\varphi)$
  - $\mathsf{sf}(\exists x.\ [\varphi]) = \{\exists x.\ [\varphi]\} \cup \mathsf{sf}(\varphi)$

Quantification: Scope

- How does one interpret the expression $\exists x.\ [\neg(x \equiv 0)] \wedge x \equiv 0$?

Quantification: Scope

- How does one interpret the expression $\exists x.\ [\neg(x \equiv 0)] \wedge x \equiv 0$?
- Need a notion of **scope** for quantifiers: Brackets for us
- Defined by the closest quantifier in the AST of the expression
- Is $\exists x.\ [\exists y.\ [\mathsf{nxt}(x) \equiv y \wedge \exists x.\ [\mathsf{nxt}(y) \equiv x]]]$ well-formed?

Quantification: Scope

- How does one interpret the expression $\exists x.\ [\neg(x \equiv 0)] \wedge x \equiv 0$?
- Need a notion of **scope** for quantifiers: Brackets for us
- Defined by the closest quantifier in the AST of the expression
- Is $\exists x.\ [\exists y.\ [\mathrm{nxt}(x) \equiv y \wedge \exists x.\ [\mathrm{nxt}(y) \equiv x]]]$ well-formed?
- Yes! Nothing forces us to use a different variable name every time.
- But it makes it harder to clearly interpret this expression.
- $\exists x.\ [\exists y.\ [\mathrm{nxt}(x) \equiv y \wedge \exists z.\ [\mathrm{nxt}(y) \equiv z]]]$ is an equivalent sentence.
- We will come back to this in a couple of slides.

Bound variables

- Inductively define the set of bound variables of an expression as follows.

$$\begin{aligned}
\mathsf{bv}(t_1 \equiv t_2) &= \emptyset, \text{ where } t_1, t_2 \text{ are terms} \\
\mathsf{bv}(P(t_1, \ldots, t_n)) &= \emptyset, \text{ for any } P \in \mathscr{P} \\
\mathsf{bv}(\neg\varphi) &= \mathsf{bv}(\varphi) \\
\mathsf{bv}(\varphi \circ \psi) &= \mathsf{bv}(\varphi) \cup \mathsf{bv}(\psi) \text{ where } \circ \in \{\wedge, \vee, \supset\} \\
\mathsf{bv}(Qx.\ [\varphi]) &= \{x\} \cup \mathsf{bv}(\varphi) \text{ where } Q \in \{\forall, \exists\}
\end{aligned}$$

- Can we now define the set of free (not bound) variables?
- Is it okay to say $\mathsf{fv}(\varphi) = \mathsf{vars}(\varphi) \setminus \mathsf{bv}(\varphi)$?

Free variables

- Let φ be the expression $\exists x.\ [\neg(x \equiv 0)] \land x \equiv 0$.
- Earlier proposal does not work; define free variables inductively as well.

$$\mathsf{fv}(t_1 \equiv t_2) = \mathsf{vars}(t_1) \cup \mathsf{vars}(t_2), \quad \text{where } t_1, t_2 \text{ are terms}$$

$$\mathsf{fv}(P(t_1, \ldots, t_n)) = \bigcup_{1 \leqslant i \leqslant n} \mathsf{vars}(t_i), \quad \text{for any } P \in \mathscr{P}$$

$$\mathsf{fv}(\neg\varphi) = \mathsf{fv}(\varphi)$$

$$\mathsf{fv}(\varphi \circ \psi) = \mathsf{fv}(\varphi) \cup \mathsf{fv}(\psi) \quad \text{where } \circ \in \{\land, \lor, \supset\}$$

$$\mathsf{fv}(Qx.\ [\varphi]) = \mathsf{fv}(\varphi) \setminus \{x\} \quad \text{where } Q \in \{\forall, \exists\}$$

- When we say x is free in φ, we mean that there is some free occurrence of x in φ. This is clearly not the **same** x which occurs bound!
- Better to keep fv and bv disjoint; rename **bound** variables!

Expressions, sentences, and formulae

- In PL, we used “expression” and “formula” interchangeably
- We could do this because there were no variables to worry about
- What about now? We want to make a distinction!
- An **expression** is any wff generated by our FOL grammar
- A **sentence** is an expression with **no free variables**
- A **formula** is an expression with **at least one free variable**
- Do not use these interchangeably!

FOL: Towards a semantics

- For PL, we assigned meaning via a valuation
- Defined truth values inductively over the structure of expressions
- Would like to assign meaning inductively here as well
- What is the inductive case for $\exists x.\ [\varphi]$?
- $\varphi(x)$, which is a formula with one free variable x
- How does one assign meaning to variables?
- To terms? To predicates?

FOL Semantics: Structures

- Defined syntax in terms of constant, function, and predicate **symbols**.
- So the various symbols need to be given meaning.
- Given a $\Sigma = (\mathscr{C}, \mathscr{F}, \mathscr{P})$, we define a **$\Sigma$-structure** $\mathscr{M}$ as a pair (M, ι), where M, the **domain** or **universe** of discourse, is a non-empty set, and ι is a function defined over $\mathscr{C} \cup \mathscr{F} \cup \mathscr{P}$ such that
  - for every constant symbol $c \in \mathscr{C}$, there is an element $\mathsf{c}_{\mathscr{M}} \in M$ of the domain such that $\iota(c) = \mathsf{c}_{\mathscr{M}}$
  - for every n-ary function symbol $f \in \mathscr{F}$, $\iota(f) = \mathsf{f}_{\mathscr{M}}$ such that $\mathsf{f}_{\mathscr{M}} : M^n \to M$
  - for every m-ary predicate symbol $P \in \mathscr{P}$, $\iota(P) = \mathsf{P}_{\mathscr{M}}$ such that $\mathsf{P}_{\mathscr{M}} \subseteq M^m$.
- We can omit the subscript when the structure is clear from context.
- Once we assign meaning to variables, we can assign meaning to all expressions.

Interlude: arithmetic example

- Consider our expression $\forall x.\ \left[x \equiv 0 \vee \left\{\exists y.\ [x \equiv \mathsf{nxt}(y)]\right\}\right]$
- What is the structure that gives meaning to this expression?
- We intend to interpret this over the naturals, so $M = \mathbb{N}$
- ι is the function which assigns the symbols the following meaning
  - $(+)$ is addition
  - $(*)$ is multiplication
  - nxt is successor
  - 0 is the natural number 0
- Is this enough to assign meaning to this expression?
- What meaning do x and y get? What meaning does $\mathsf{nxt}(y)$ get?

FOL Semantics

- Let $\Sigma = (\mathscr{C}, \mathscr{F}, \mathscr{P})$ be a signature.
- An **interpretation** for Σ is a pair $\mathscr{I} = (\mathscr{M}, \sigma)$, where
  - $\mathscr{M} = (M, \iota)$ is a Σ-structure, and
  - $\sigma : \mathscr{V} \to M$ is a function which assigns elements of M to variables in $\mathscr{V}$.
- We will often call $\mathscr{I}$ an interpretation "based on" the Σ-structure $\mathscr{M}$
- Once we fix an interpretation $\mathscr{I}$, each term t over Σ maps to a unique element $t^{\mathscr{I}}$ in M as follows.
  - If $t = x \in \mathscr{V}$, then $t^{\mathscr{I}} = \sigma(x)$
  - If $t = c \in C$, then $t^{\mathscr{I}} = c_{\mathscr{M}}$
  - If $t = f(t_1, \dots, t_n)$ for some n terms $t_1, \dots, t_n$ and an n-ary $f \in \mathscr{F}$, then $t^{\mathscr{I}} = \mathsf{f}_{\mathscr{M}}({t_1}^{\mathscr{I}}, \dots, {t_n}^{\mathscr{I}})$
- Think of terms as "names" for elements in the domain!

FOL Semantics

- Consider the expression $x \equiv y$ over $(\mathbb{N}, \iota)$.
- Suppose $\mathcal{I} = ((\mathbb{N}, \iota), \sigma)$ is such that $\sigma(x) = 3$ and $\sigma(y) = 5$.
- Is there anything that disallows such an interpretation?

FOL Semantics

- Consider the expression $x \equiv y$ over $(\mathbb{N}, \iota)$.
- Suppose $\mathcal{I} = ((\mathbb{N}, \iota), \sigma)$ is such that $\sigma(x) = 3$ and $\sigma(y) = 5$.
- Is there anything that disallows such an interpretation? No!
- Is the expression true under this interpretation? Obviously not.
- Much like valuations, there are interpretations and then there are interpretations.
- Interested in interpretations which **satisfy** a given expression.

Satisfaction relation

- We denote the fact that an interpretation $\mathcal{I} = (\mathcal{M}, \sigma)$ **satisfies** an expression $\varphi \in \mathsf{FO}_\Sigma$ by the familiar $\mathcal{I} \vDash \varphi$ notation.
- We define this inductively, as usual, as follows.

$\mathcal{I} \vDash t_1 \equiv t_2$ if $t_1{}^{\mathcal{I}} = t_2{}^{\mathcal{I}}$

$\mathcal{I} \vDash P(t_1, \dots, t_n)$ if $(t_1{}^{\mathcal{I}}, \dots, t_n{}^{\mathcal{I}}) \in \mathsf{P}_{\mathcal{M}}$

$\mathcal{I} \vDash \exists x.\ [\varphi]$ if there is some $m \in M$ such that $\mathcal{I}[x \mapsto m] \vDash \varphi$

$\mathcal{I} \vDash \forall x.\ [\varphi]$ if, for every $m \in M$, it is the case that $\mathcal{I}[x \mapsto m] \vDash \varphi$

where we define $\mathcal{I}[x \mapsto m]$ to be $(\mathcal{M}, \sigma')$ (where $\mathcal{I} = (\mathcal{M}, \sigma)$) such that

$$\sigma'(z) = \begin{cases} m & z = x \\ \sigma(z) & \text{otherwise} \end{cases}$$

$\mathcal{I} \vDash \neg\varphi$ if $\mathcal{I} \nvDash \varphi$

$\mathcal{I} \vDash \varphi \wedge \psi$ if $\mathcal{I} \vDash \varphi$ and $\mathcal{I} \vDash \psi$

$\mathcal{I} \vDash \varphi \vee \psi$ if $\mathcal{I} \vDash \varphi$ or $\mathcal{I} \vDash \psi$

$\mathcal{I} \vDash \varphi \supset \psi$ if $\mathcal{I} \nvDash \varphi$ or $\mathcal{I} \vDash \psi$

Satisfiability and validity

- We say that $\varphi \in \mathsf{FO}_\Sigma$ is **satisfiable** if there is an interpretation $\mathcal{I}$ based on a Σ-structure $\mathcal{M}$ such that $\mathcal{I} \models \varphi$.
- We say that $\varphi \in \mathsf{FO}_\Sigma$ is **valid** if, for every Σ-structure $\mathcal{M}$ and every interpretation $\mathcal{I}$ based on $\mathcal{M}$, it is the case that $\mathcal{I} \models \varphi$.
- A **model** of φ is an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \varphi$.
- We lift the notion of satisfiability to sets of formulas, and denote it by $\mathcal{I} \models X$, where $X \subseteq \mathsf{FO}_\Sigma$.
- We say that $X \models \varphi$ (X **logically entails** φ) for $X \cup \{\varphi\} \subseteq \mathsf{FO}_\Sigma$ if for every interpretation $\mathcal{I}$, if $\mathcal{I} \models X$ then $\mathcal{I} \models \varphi$.