Lecture 11 - FO: Truth and models

Vaishnavi Sundararajan

COL703 - 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.

Recap: Expressions, sentences, and formulae

- Notation: For a given Σ
  - the set of all expressions over Σ is denoted by FO_Σ
  - the set of all terms over Σ and $\mathscr{V}$ is denoted by $\mathsf{T}(\Sigma)$
- Defined notions of bound and free variables
- 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**
- Rename bound variables to keep bound and free variables distinct!
- Keep variable names distinct within the same set (bound/free) also.
- We will assume this in whatever follows to simplify the presentation.
  - No $x \in \mathscr{V}$ appears both free and bound.
  - No $x \in \mathscr{V}$ is bound twice.

Recap: FOL Semantics

- 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 $\mathsf{c}_{\mathscr{M}} \in M$ s,t, $\iota(c) = \mathsf{c}_{\mathscr{M}}$
  - for every n-ary function symbol $f \in \mathscr{F}$, $\iota(f) = \mathsf{f}_{\mathscr{M}}$ s.t. $\mathsf{f}_{\mathscr{M}} : M^n \to M$
  - for every m-ary predicate symbol $P \in \mathscr{P}$, $\iota(P) = \mathsf{P}_{\mathscr{M}}$ s.t. $\mathsf{P}_{\mathscr{M}} \subseteq M^m$.
- 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 maps variables in $\mathscr{V}$ to elements of M.
- Each term t over Σ maps to a unique element $t^{\mathscr{I}}$ in M under $\mathscr{I}$.
  - 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 $t^{\mathscr{I}} = \mathsf{f}_{\mathscr{M}}(t_1{}^{\mathscr{I}}, \dots, t_n{}^{\mathscr{I}})$

Recap: 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 $\mathcal{I} \vDash \psi$

Recap: 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 $\mathcal{I} \vDash \varphi$.
- 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 $\mathcal{I} \vDash \varphi$.
- A **model** of φ is an interpretation $\mathcal{I}$ such that $\mathcal{I} \vDash \varphi$.
- We lift the notion of satisfiability to sets of formulas, and denote it by $\mathcal{I} \vDash X$, where $X \subseteq \mathsf{FO}_\Sigma$.
- We say that $X \vDash \varphi$ (X **logically entails** φ) for $X \cup \{\varphi\} \subseteq \mathsf{FO}_\Sigma$ if for every interpretation $\mathcal{I}$, if $\mathcal{I} \vDash X$ then $\mathcal{I} \vDash \varphi$.

Satisfiability

- As usual, want to check for satisfiability of a given FO expression over a signature Σ
- Need a Σ-structure $\mathscr{M}$, and a model $\mathscr{I}$ based on $\mathscr{M}$
- In general, Σ will allow us to (somewhat) narrow down the expected application (arithmetic, graphs etc)
- But sometimes, unexpected models can come to light!

Satisfiability: Example

- Consider a signature $\Sigma = (\emptyset, \emptyset, P/2)$.
- Is $\varphi \coloneqq \forall x.\ [\forall y.\ [\forall z.\ [(Pxy \wedge Pyz) \supset Pxz]]] \in \mathsf{FO}_\Sigma$ satisfiable?

Satisfiability: Example

- Consider a signature $\Sigma = (\emptyset, \emptyset, P/2)$.
- Is $\varphi \coloneqq \forall x.\ [\forall y.\ [\forall z.\ [(Pxy \wedge Pyz) \supset Pxz]]] \in \mathsf{FO}_\Sigma$ satisfiable?
- We define a candidate structure $\mathscr{M} = (M, \iota)$, where
  - $M = \{1, 2, 3\}$
  - $\iota(P) = \{(1, 2), (2, 3), (1, 3)\}$
- Fix $\mathscr{I} = (\mathscr{M}, \sigma)$, where $\sigma(x) = 1$ for every $x \in \mathscr{V}$.
- Does $\mathscr{I} \vDash \forall x.\ [\forall y.\ [\forall z.\ [(Pxy \wedge Pyz) \supset Pxz]]]$?

Satisfiability: Example

- $\mathcal{M} = (\{1,2,3\}, \iota)$, with $\iota(P) = \{(1,2),(2,3),(1,3)\}$
- Fix $\mathcal{I} = (\mathcal{M}, \sigma)$, where $\sigma(x) = 1$ for every $x \in \mathcal{V}$. (More on this later)
- Does $\mathcal{I} \models \forall x.\ [\forall y.\ [\forall z.\ [(Pxy \land Pyz) \supset Pxz]]]$?
- Need to check all possible instantiations of the universally quantified variables.
- One case:
  - Need to check if $\mathcal{I}[x \mapsto 1] \models \forall y.\ [\forall z.\ [(Pxy \land Pyz) \supset Pxz]]$
  - Need to check if $\mathcal{I}[x \mapsto 1, y \mapsto 1] \models \forall z.\ [(Pxy \land Pyz) \supset Pxz]$
  - Need to check if $\mathcal{I}[x \mapsto 1, y \mapsto 1, z \mapsto 1] \models (Pxy \land Pyz) \supset Pxz$
- Is this true?

Satisfiability: Example

- $\mathscr{M} = (\{1, 2, 3\}, \iota)$, with $\iota(P) = \{(1, 2), (2, 3), (1, 3)\}$
- Fix $\mathscr{I} = (\mathscr{M}, \sigma)$, where $\sigma(x) = 1$ for every $x \in \mathscr{V}$. (More on this later)
- Does $\mathscr{I} \vDash \forall x.\ [\forall y.\ [\forall z.\ [(Pxy \wedge Pyz) \supset Pxz]]]$?
- Need to check all possible instantiations of the universally quantified variables.
- One case:
  - Need to check if $\mathscr{I}[x \mapsto 1] \vDash \forall y.\ [\forall z.\ [(Pxy \wedge Pyz) \supset Pxz]]$
  - Need to check if $\mathscr{I}[x \mapsto 1, y \mapsto 1] \vDash \forall z.\ [(Pxy \wedge Pyz) \supset Pxz]$
  - Need to check if $\mathscr{I}[x \mapsto 1, y \mapsto 1, z \mapsto 1] \vDash (Pxy \wedge Pyz) \supset Pxz$
- Is this true? Yes! The precondition is false, so vacuously true.
- Many other cases are made vacuously true similarly.

Satisfiability: Example

- $\mathscr{M} = (\{1,2,3\}, \iota)$, with $\iota(P) = \{(1,2),(2,3),(1,3)\}$
- Fix $\mathscr{I} = (\mathscr{M}, \sigma)$, where $\sigma(x) = 1$ for every $x \in \mathscr{V}$.
- Interesting case is when (m_1, m_2) and (m_2, m_3) are in $P_{\mathscr{M}}$.
- Could be a problem if $(m_1, m_3) \notin P_{\mathscr{M}}$
- Does $\mathscr{I}[x \mapsto 1, y \mapsto 2, z \mapsto 3] \vDash (Pxy \land Pyz) \supset Pxz$? Also yes!
- So $\mathscr{I} \vDash \varphi$, and φ is satisfiable.

Satisfiability: Example

- $\mathscr{M} = (\{1, 2, 3\}, \iota)$, with $\iota(P) = \{(1, 2), (2, 3), (1, 3)\}$
- Fix $\mathscr{I} = (\mathscr{M}, \sigma)$, where $\sigma(x) = 1$ for every $x \in \mathscr{V}$.
- Interesting case is when (m_1, m_2) and (m_2, m_3) are in $P_{\mathscr{M}}$.
- Could be a problem if $(m_1, m_3) \notin P_{\mathscr{M}}$
- Does $\mathscr{I}[x \mapsto 1, y \mapsto 2, z \mapsto 3] \models (Pxy \wedge Pyz) \supset Pxz$? Also yes!
- So $\mathscr{I} \models \varphi$, and φ is satisfiable. Is φ valid?

Satisfiability: Example

- $\mathcal{M} = (\{1, 2, 3\}, \iota)$, with $\iota(P) = \{(1, 2), (2, 3), (1, 3)\}$
- Fix $\mathcal{I} = (\mathcal{M}, \sigma)$, where $\sigma(x) = 1$ for every $x \in \mathcal{V}$.
- Interesting case is when (m_1, m_2) and (m_2, m_3) are in $P_{\mathcal{M}}$.
- Could be a problem if $(m_1, m_3) \notin P_{\mathcal{M}}$
- Does $\mathcal{I}[x \mapsto 1, y \mapsto 2, z \mapsto 3] \vDash (Pxy \wedge Pyz) \supset Pxz$? Also yes!
- So $\mathcal{I} \vDash \varphi$, and φ is satisfiable. Is φ valid?
- As always, easier to prove **invalidity**.
- $\mathcal{M}' = (\{1, 2, 3\}, \iota')$, with $\iota'(P) = \{(1, 2), (2, 3), (3, 1)\}$
- **Exercise**: Show that $(\mathcal{M}', \sigma') \nvDash \varphi$ (for any σ'!)
- φ is true exactly when the binary relation is transitive.

Satisfiability: Example

- Is $\psi \coloneqq \forall x.\ [\exists y.\ [Pxy \land \forall z.\ [Pxz \supset y \equiv z]]]$ satisfiable?

Satisfiability: Example

- Is $\psi \coloneqq \forall x.\ [\exists y.\ [Pxy \wedge \forall z.\ [Pxz \supset y \equiv z]]]$ satisfiable?
- $\mathcal{I} = (\mathcal{M}', \sigma)$ exactly as in the previous example.
- Does $\mathcal{I} \models \psi$? Consider a "first" case.
- Need to check if $\mathcal{I}[x \mapsto 1] \models \exists y.\ [Pxy \wedge \forall z.\ [Pxz \supset y \equiv z]]$
- Need to check if there is some $m \in \{1, 2, 3\}$ such that $\mathcal{I}[x \mapsto 1, y \mapsto m] \models Pxy \wedge \forall z.\ [Pxz \supset y \equiv z]$
- Need to check if there is some $m \in \{1, 2, 3\}$ such that $\mathcal{I}[x \mapsto 1, y \mapsto m] \models Pxy$ and $\mathcal{I}[x \mapsto 1, y \mapsto m] \models \forall z.\ [Pxz \supset y \equiv z]$
- Which m? Not sure yet.

Satisfiability: Example

- Is $\psi \coloneqq \forall x.\ [\exists y.\ [Pxy \wedge \forall z.\ [Pxz \supset y \equiv z]]]$ satisfiable?
- $\mathcal{I} = (\mathcal{M}', \sigma)$ exactly as in the previous example.
- Does $\mathcal{I} \vDash \psi$? Consider a "first" case.
- Need to check if $\mathcal{I}[x \mapsto 1] \vDash \exists y.\ [Pxy \wedge \forall z.\ [Pxz \supset y \equiv z]]$
- Need to check if there is some $m \in \{1, 2, 3\}$ such that $\mathcal{I}[x \mapsto 1, y \mapsto m] \vDash Pxy \wedge \forall z.\ [Pxz \supset y \equiv z]$
- Need to check if there is some $m \in \{1, 2, 3\}$ such that $\mathcal{I}[x \mapsto 1, y \mapsto m] \vDash Pxy$ and $\mathcal{I}[x \mapsto 1, y \mapsto m] \vDash \forall z.\ [Pxz \supset y \equiv z]$
- Which m? Not sure yet. **But same m for both!**

Satisfiability: Example

- $\mathscr{M}' = (\{1,2,3\}, \iota'), \iota'(P) = \{(1,2),(2,3),(3,1)\}$
- Let's try $m = 1$.
- Need to check if $\mathcal{I}[x \mapsto 1, y \mapsto 1] \vDash Pxy$ and $\mathcal{I}[x \mapsto 1, y \mapsto 1, z \mapsto 1] \vDash Pxz \supset y \equiv z$

Satisfiability: Example

- $\mathscr{M}' = (\{1, 2, 3\}, \iota'), \iota'(P) = \{(1, 2), (2, 3), (3, 1)\}$
- Let's try $m = 1$.
- Need to check if $\mathcal{I}[x \mapsto 1, y \mapsto 1] \vDash Pxy$ and $\mathcal{I}[x \mapsto 1, y \mapsto 1, z \mapsto 1] \vDash Pxz \supset y \equiv z$
- Vacuously true! Interesting case is when x and z are "in the relation"
- Need to check if $\mathcal{I}[x \mapsto 1, y \mapsto 1] \vDash Pxy$ and $\mathcal{I}[x \mapsto 1, y \mapsto 1, z \mapsto 2] \vDash Pxz \supset y \equiv z$

Satisfiability: Example

- $\mathscr{M}' = (\{1,2,3\}, \iota'), \iota'(P) = \{(1,2),(2,3),(3,1)\}$
- Let's try $m = 1$.
- Need to check if $\mathscr{I}[x \mapsto 1, y \mapsto 1] \models Pxy$ and $\mathscr{I}[x \mapsto 1, y \mapsto 1, z \mapsto 1] \models Pxz \supset y \equiv z$
- Vacuously true! Interesting case is when x and z are "in the relation"
- Need to check if $\mathscr{I}[x \mapsto 1, y \mapsto 1] \models Pxy$ and $\mathscr{I}[x \mapsto 1, y \mapsto 1, z \mapsto 2] \models Pxz \supset y \equiv z$
- Not true! $(1,2) \in \iota'(P)$, but $1 \neq 2$
- What if $m = 3$?

Satisfiability: Example

- $\mathscr{M}' = (\{1, 2, 3\}, \iota'), \iota'(P) = \{(1, 2), (2, 3), (3, 1)\}$
- Let's try $m = 1$.
- Need to check if $\mathcal{I}[x \mapsto 1, y \mapsto 1] \vDash Pxy$ and $\mathcal{I}[x \mapsto 1, y \mapsto 1, z \mapsto 1] \vDash Pxz \supset y \equiv z$
- Vacuously true! Interesting case is when x and z are "in the relation"
- Need to check if $\mathcal{I}[x \mapsto 1, y \mapsto 1] \vDash Pxy$ and $\mathcal{I}[x \mapsto 1, y \mapsto 1, z \mapsto 2] \vDash Pxz \supset y \equiv z$
- Not true! $(1, 2) \in \iota'(P)$, but $1 \neq 2$
- What if $m = 3$? Also does not work. $(1, 2) \in \iota'(P)$, but $3 \neq 2$

Satisfiability: Example

- Taking m to be 2 works. (Work it out!)
- So $\mathcal{I} \models \psi$, and ψ is satisfiable.
- For each value u assigned to x, take m to be v such that $(u, v) \in \iota'(P)$
- Value of m is a function of the value assigned to x (This will be important later!)
- **Important**: The value of m changes with the value assigned to x
- Essentially the difference between $\forall x.\ [\exists y.\ [\ldots]]$ and $\exists y.\ [\forall x.\ [\ldots]]$
- **Exercise**: What property of the structure does ψ code up?
- **Exercise**: Is ψ valid?

Satisfiability: Example

- Is $\chi(x) \coloneqq \forall y. \, [\neg(x \equiv y) \supset (Pxy \wedge \neg Pyx)]$ satisfiable?

Satisfiability: Example

- Is $\chi(x) \coloneqq \forall y.\ [\neg(x \equiv y) \supset (Pxy \wedge \neg Pyx)]$ satisfiable?
- We define a candidate structure $\mathscr{M} = (M, \iota)$, where
  - $M = \{1, 2, 3\}$
  - $\iota(P) = \{(2, 1), (2, 3), (3, 3)\}$
- Fix $\mathscr{I} = (\mathscr{M}, \sigma)$, where

Satisfiability: Example

- Is $\chi(x) \coloneqq \forall y.\ [\neg(x \equiv y) \supset (Pxy \wedge \neg Pyx)]$ satisfiable?
- We define a candidate structure $\mathscr{M} = (M, \iota)$, where
  - $M = \{1, 2, 3\}$
  - $\iota(P) = \{(2,1), (2,3), (3,3)\}$
- Fix $\mathscr{I} = (\mathscr{M}, \sigma)$, where $\sigma(x) = 2$ and $\sigma(y) = 1$ for all **other** $y \in \mathscr{V}$.
- Does $\mathscr{I} \vDash \forall y.\ [\neg(x \equiv y) \supset (Pxy \wedge \neg Pyx)]$?
- “First” case: Need to check if $\mathscr{I}[y \mapsto 1] \vDash \neg(x \equiv y) \supset (Pxy \wedge \neg Pyx)$

Satisfiability: Example

- $\mathscr{M} = (\{1, 2, 3\}, \iota)$ with $\iota(P) = \{(2, 1), (2, 3), (3, 3)\}$
- $\sigma(x) = 2$ and $\sigma(y) = 1$ for all **other** $y \in \mathscr{V}$.
- "First" case: Need to check if $\mathscr{I}[y \mapsto 1] \models \neg(x \equiv y) \supset (Pxy \wedge \neg Pyx)$
- Same as checking if
  $(\mathscr{M}, [x \mapsto 2, y \mapsto 1, \_ \mapsto 1]) \models \neg(x \equiv y) \supset (Pxy \wedge \neg Pyx)$

Satisfiability: Example

- $\mathcal{M} = (\{1, 2, 3\}, \iota)$ with $\iota(P) = \{(2, 1), (2, 3), (3, 3)\}$
- $\sigma(x) = 2$ and $\sigma(y) = 1$ for all **other** $y \in \mathcal{V}$.
- "First" case: Need to check if $\mathcal{I}[y \mapsto 1] \models \neg(x \equiv y) \supset (Pxy \wedge \neg Pyx)$
- Same as checking if $(\mathcal{M}, [x \mapsto 2, y \mapsto 1, \_ \mapsto 1]) \models \neg(x \equiv y) \supset (Pxy \wedge \neg Pyx)$
- Other cases also work out! So $\mathcal{I} \models \chi(x)$.
- Let $\sigma'(x) = 2$ and $\sigma'(y) = 3$ for all other $y \in \mathcal{V}$. Does $(\mathcal{M}, \sigma') \models \chi(x)$?

Satisfiability: Example

- $\mathcal{M} = (\{1, 2, 3\}, \iota)$ with $\iota(P) = \{(2, 1), (2, 3), (3, 3)\}$
- $\sigma(x) = 2$ and $\sigma(y) = 1$ for all **other** $y \in \mathcal{V}$.
- "First" case: Need to check if $\mathcal{I}[y \mapsto 1] \vDash \neg(x \equiv y) \supset (Pxy \wedge \neg Pyx)$
- Same as checking if $(\mathcal{M}, [x \mapsto 2, y \mapsto 1, \_ \mapsto 1]) \vDash \neg(x \equiv y) \supset (Pxy \wedge \neg Pyx)$
- Other cases also work out! So $\mathcal{I} \vDash \chi(x)$.
- Let $\sigma'(x) = 2$ and $\sigma'(y) = 3$ for all other $y \in \mathcal{V}$. Does $(\mathcal{M}, \sigma') \vDash \chi(x)$?
- Let $\sigma''(x) = 3$ and $\sigma''(y) = 1$ for all other $y \in \mathcal{V}$. Does $(\mathcal{M}, \sigma'') \vDash \chi(x)$?
- **Exercise**: Is $\chi(x)$ valid? What would it mean for $\chi(x)$ to be valid?

Satisfiability: Example

- Can talk about satisfiability for a set of sentences (called a **theory**)
- Fix a signature $\Sigma = (\{\varepsilon\}, \{f/2\}, \emptyset)$
- Consider the following sentences:

$$\forall x.\ [\forall y.\ [\forall z.\ [f(f(x,y),z) \equiv f(x,f(y,z))]]]$$

$$\forall x.\ [f(x,\varepsilon) \equiv x]$$

$$\forall x.\ [\exists y.\ [f(x,y) \equiv \varepsilon]]$$

- Is there an interpretation that is a model for all three?

Satisfiability of formulae and sentences

- Earlier example with $\chi(x)$: Both $(\mathcal{M}, \sigma)$ and $(\mathcal{M}, \sigma')$ were models
- Only required that σ and σ' agreed on $\mathsf{fv}(\chi(x))$
- Recall: only considered PL valuations restricted to atoms of expression
- **Theorem**: Let Σ be an FO signature and $\varphi \in \mathsf{FO}_\Sigma$. Let $\mathcal{M}$ be a Σ-structure and σ, σ' assignments which agree on $\mathsf{fv}(\varphi)$. Then $(\mathcal{M}, \sigma) \models \varphi$ iff $(\mathcal{M}, \sigma') \models \varphi$. Proof: **Exercise!**
- Can we now say something about the satisfiability of **sentences**?

Satisfiability of formulae and sentences

- Earlier example with $\chi(x)$: Both $(\mathcal{M}, \sigma)$ and $(\mathcal{M}, \sigma')$ were models
- Only required that σ and σ' agreed on $\mathsf{fv}(\chi(x))$
- Recall: only considered PL valuations restricted to atoms of expression
- **Theorem**: Let Σ be an FO signature and $\varphi \in \mathsf{FO}_\Sigma$. Let $\mathcal{M}$ be a Σ-structure and σ, σ' assignments which agree on $\mathsf{fv}(\varphi)$. Then $(\mathcal{M}, \sigma) \models \varphi$ iff $(\mathcal{M}, \sigma') \models \varphi$. Proof: **Exercise!**
- Can we now say something about the satisfiability of **sentences**?
- **Corollary**: Let Σ be an FO signature and $\varphi \in \mathsf{FO}_\Sigma$ **be a sentence**. Let $\mathcal{M}$ be a Σ-structure. Then, for any assignments σ, σ', it is the case that $(\mathcal{M}, \sigma) \models \varphi$ iff $(\mathcal{M}, \sigma') \models \varphi$.

Satisfiability in general

- Recall what we did for satisfiability and validity in PL
- Cast PL expression into CNF, then did resolution
- If a PL expression is in DNF, checking for satisfiability is easy
- Normal forms are useful in general from an automation perspective!
- Easier to handle for algorithms
    - Especially if one can algorithmically obtain the normal form also!
- What does a normal form look like for FO? Are there many such?
- First, some notational shorthand going forward.
- Use $\forall x_1 x_2 \ldots x_n$ as shorthand for $\forall x_1.\ [\forall x_2.\ [\ldots \forall x_n.\ [\ldots] \ldots]]$
- Omit brackets when clear from context.

Towards a normal form

- Handling nested quantifiers took some doing, maybe get rid of that?
- Cannot get rid of quantifiers entirely without assignment
- So what is the next best thing we might try?

Towards a normal form

- Handling nested quantifiers took some doing, maybe get rid of that?
- Cannot get rid of quantifiers entirely without assignment
- So what is the next best thing we might try?
- Push all quantifiers out into one “block” at the head of the expression
- Do all instantiations upfront; then evaluate the resultant expression
- Recall: Can always push negation inside the quantifier
- Can we do this for other connectives also?
- But first, we need to talk about **substitutions**

Substitutions

- A **substitution** θ is a partial map from $\mathscr{V}$ to $\mathrm{T}(\Sigma)$, with a finite domain
- We can lift this to terms, inductively as usual (**Exercise!**)
- $\theta(t) = t$ for a term t in the language, if $\mathsf{vars}(t) \cap \mathsf{dom}(\theta) = \emptyset$
- Often write $t\theta$ to mean $\theta(t)$; $t\theta$ is a "substitution instance" of t
- We often write $\theta = \{t/x \mid x\theta = t \text{ and } x \in \mathsf{dom}(\theta)\}$
- What effect does θ have on the semantics of expressions?
- **Theorem**: Given an interpretation $\mathscr{I} = ((M, \iota), \sigma)$ for some Σ, a term $t \in \mathrm{T}(\Sigma)$, and a substitution $\{u/x\}$ such that $u^{\mathscr{I}} = m \in M$, it is the case that $(t\{u/x\})^{\mathscr{I}} = t^{\mathscr{I}[x \mapsto m]}$. Proof: **Exercise!**
- Lift to expressions as usual; ensure distinct bound and free variables.
- A substitution θ is **admissible** for a term t (resp. an expression φ) if $\mathsf{vars}(\mathsf{rng}(\theta)) \cap \mathsf{vars}(t) = \emptyset$ (resp. $\mathsf{vars}(\mathsf{rng}(\theta)) \cap \mathsf{vars}(\varphi) = \emptyset$).

Back to normal forms

- Want to move quantifiers into one block at the head of the expression
- **Theorem**: Let $z \notin \mathsf{fv}(\varphi) \cup \mathsf{fv}(\psi) \cup \{x_1, \ldots, x_n\}$, where $n \geqslant 0$. For $Q_i \in \{\forall, \exists\}$ for every $1 \leqslant i \leqslant n$, the following equivalences hold.

$$Q_1 x_1 \ldots Q_n x_n.\ [\neg Qy.\ [\varphi]] \Leftrightarrow Q_1 x_1 \ldots Q_n x_n.\ \overline{Q}y.\ [\neg\varphi]$$

$$Q_1 x_1 \ldots Q_n x_n.\ [\psi \circ Qy.\ [\varphi]] \Leftrightarrow Q_1 x_1 \ldots Q_n x_n.\ Qz.\ [\psi \circ \varphi\{z/y\}]$$

$$Q_1 x_1 \ldots Q_n x_n.\ [Qy.\ [\varphi] * \psi] \Leftrightarrow Q_1 x_1 \ldots Q_n x_n.\ Qz.\ [\varphi\{z/y\} * \psi]$$

$$Q_1 x_1 \ldots Q_n x_n.\ [Qy.\ [\varphi] \supset \psi] \Leftrightarrow Q_1 x_1 \ldots Q_n x_n.\ \overline{Q}z.\ [\varphi\{z/y\} \supset \psi]$$

where $\circ \in \{\wedge, \vee, \supset\}$, and $* \in \{\wedge, \vee\}$, and $\overline{Q} = \begin{cases} \exists & \text{if } Q = \forall \\ \forall & \text{if } Q = \exists \end{cases}$