

Lecture 11 - FO: Truth and models

Vaishnavi Sundararajan

COL703 - Logic for Computer Science

Recap: FOL Syntax

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

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

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

$$t := x \in \mathcal{V} \mid c \in \mathcal{C} \mid f(t_1, \dots, t_m)$$

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 \mathcal{V} is denoted by $\text{T}(\Sigma)$
- 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 $x \in \mathcal{V}$ appears both free and bound.
 - No $x \in \mathcal{V}$ is bound twice.

Recap: FOL Semantics

- Given a $\Sigma = (\mathcal{C}, \mathcal{F}, \mathcal{P})$, we define a Σ -**structure** \mathcal{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 $\mathcal{C} \cup \mathcal{F} \cup \mathcal{P}$ such that
 - for every constant symbol $c \in \mathcal{C}$, there is $c_{\mathcal{M}} \in M$ s.t. $\iota(c) = c_{\mathcal{M}}$
 - for every n -ary function symbol $f \in \mathcal{F}$, $\iota(f) = f_{\mathcal{M}}$ s.t. $f_{\mathcal{M}} : M^n \rightarrow M$
 - for every m -ary predicate symbol $P \in \mathcal{P}$, $\iota(P) = P_{\mathcal{M}}$ s.t. $P_{\mathcal{M}} \subseteq M^m$.
- An **interpretation** for Σ is a pair $\mathcal{I} = (\mathcal{M}, \sigma)$, where
 - $\mathcal{M} = (M, \iota)$ is a Σ -structure, and
 - $\sigma : \mathcal{V} \rightarrow M$ is a function which maps variables in \mathcal{V} to elements of M .
- Each term t over Σ maps to a unique element $t^{\mathcal{I}}$ in M under \mathcal{I} .
 - If $t = x \in \mathcal{V}$, then $t^{\mathcal{I}} = \sigma(x)$
 - If $t = c \in \mathcal{C}$, then $t^{\mathcal{I}} = c_{\mathcal{M}}$
 - If $t = f(t_1, \dots, t_n)$ for some n terms t_1, \dots, t_n and an n -ary $f \in \mathcal{F}$, then $t^{\mathcal{I}} = f_{\mathcal{M}}(t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}})$

Recap: Satisfaction relation

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

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

$$\mathcal{I} \models P(t_1, \dots, t_n) \text{ if } (t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}}) \in P_{\mathcal{M}}$$

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

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

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

(where $\mathcal{I} = (\mathcal{M}, \sigma)$) such that

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

$$\mathcal{I} \models \neg \varphi \text{ if } \mathcal{I} \not\models \varphi$$

$$\mathcal{I} \models \varphi \wedge \psi \text{ if } \mathcal{I} \models \varphi \text{ and } \mathcal{I} \models \psi$$

$$\mathcal{I} \models \varphi \vee \psi \text{ if } \mathcal{I} \models \varphi \text{ or } \mathcal{I} \models \psi$$

$$\mathcal{I} \models \varphi \supset \psi \text{ if } \mathcal{I} \not\models \varphi \text{ or } \mathcal{I} \models \psi$$

Recap: Satisfiability and validity

- We say that $\varphi \in \text{FO}_\Sigma$ 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 $\varphi \in \text{FO}_\Sigma$ 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 $X \subseteq \text{FO}_\Sigma$.
- We say that $X \models \varphi$ (**X logically entails φ**) for $X \cup \{\varphi\} \subseteq \text{FO}_\Sigma$ if for every interpretation \mathcal{I} , if $\mathcal{I} \models X$ then $\mathcal{I} \models \varphi$.

Satisfiability

- As usual, want to check for satisfiability of a given FO expression over a signature Σ
- Need a Σ -structure \mathcal{M} , and a model \mathcal{I} based on \mathcal{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 $\varphi := \forall x. [\forall y. [\forall z. [(Pxy \wedge Pyz) \supset Pxz]]] \in \text{FO}_{\Sigma}$ satisfiable?

Satisfiability: Example

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

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 $\mathcal{I} \models \forall x. [\forall y. [\forall z. [(Pxy \wedge Pyz) \supset Pxz]]]$?
- Need to check all possible instantiations of the universally quantified variables.
- One case:
 - Need to check if $\mathcal{I}[x \mapsto 1] \models \forall y. [\forall z. [(Pxy \wedge Pyz) \supset Pxz]]$
 - Need to check if $\mathcal{I}[x \mapsto 1, y \mapsto 1] \models \forall z. [(Pxy \wedge Pyz) \supset Pxz]$
 - Need to check if $\mathcal{I}[x \mapsto 1, y \mapsto 1, z \mapsto 1] \models (Pxy \wedge Pyz) \supset Pxz$
- Is this true?

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 $\mathcal{I} \models \forall x. [\forall y. [\forall z. [(Pxy \wedge Pyz) \supset Pxz]]]$?
- Need to check all possible instantiations of the universally quantified variables.
- One case:
 - Need to check if $\mathcal{I}[x \mapsto 1] \models \forall y. [\forall z. [(Pxy \wedge Pyz) \supset Pxz]]$
 - Need to check if $\mathcal{I}[x \mapsto 1, y \mapsto 1] \models \forall z. [(Pxy \wedge Pyz) \supset Pxz]$
 - Need to check if $\mathcal{I}[x \mapsto 1, y \mapsto 1, z \mapsto 1] \models (Pxy \wedge Pyz) \supset Pxz$
- Is this true? Yes! The precondition is false, so vacuously true.
- Many other cases are made vacuously true similarly.

Satisfiability: Example

- $\mathcal{M} = (\{1, 2, 3\}, \iota)$, 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 $\mathcal{I}[x \mapsto 1, y \mapsto 2, z \mapsto 3] \models (Pxy \wedge Pyz) \supset Pxz$? Also yes!
- So $\mathcal{I} \models \varphi$, and φ is satisfiable.

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 $\mathcal{I}[x \mapsto 1, y \mapsto 2, z \mapsto 3] \models (Pxy \wedge Pyz) \supset Pxz$? Also yes!
- So $\mathcal{I} \models \varphi$, 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 $\mathcal{F} = (\mathcal{M}, \sigma)$, 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 $\mathcal{F}[x \mapsto 1, y \mapsto 2, z \mapsto 3] \models (Pxy \wedge Pyz) \supset Pxz$? Also yes!
- So $\mathcal{F} \models \varphi$, 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 $(\mathcal{M}', \sigma') \not\models \varphi$ (for any σ' !)
- φ is true exactly when the binary relation is transitive.

Satisfiability: Example

- Is $\psi := \forall x. [\exists y. [Pxy \wedge \forall z. [Pxz \supset y \equiv z]]]$ satisfiable?

Satisfiability: Example

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

Satisfiability: Example

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

Satisfiability: Example

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

Satisfiability: Example

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

Satisfiability: Example

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

Satisfiability: Example

- $\mathcal{M}' = (\{1, 2, 3\}, \iota'), \iota'(P) = \{(1, 2), (2, 3), (3, 1)\}$
- Let's try $m = 1$.
- Need to check if $\mathcal{F}[x \mapsto 1, y \mapsto 1] \models Pxy$ and $\mathcal{F}[x \mapsto 1, y \mapsto 1, z \mapsto 1] \models Pxz \supset y \equiv z$
- Vacuously true! Interesting case is when x and z are “in the relation”
- Need to check if $\mathcal{F}[x \mapsto 1, y \mapsto 1] \models Pxy$ and $\mathcal{F}[x \mapsto 1, y \mapsto 1, z \mapsto 2] \models Pxz \supset y \equiv z$
- Not true! $(1, 2) \in \iota'(P)$, but $1 \neq 2$
- What if $m = 3$? 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 $\mathcal{J} \models \psi$, 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 $\forall x. [\exists y. [...]]$ and $\exists y. [\forall x. [...]]$
- **Exercise:** What property of the structure does ψ code up?
- **Exercise:** Is ψ valid?

Satisfiability: Example

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

Satisfiability: Example

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

Satisfiability: Example

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

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 $\mathcal{F}[y \mapsto 1] \models Pxy \wedge \neg Pyx \wedge \neg(x \equiv y)$
- Same as checking if $(\mathcal{M}, [x \mapsto 2, y \mapsto 1, _ \mapsto 1]) \models Pxy \wedge \neg Pyx \wedge \neg(x \equiv y)$

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 $\mathcal{F}[y \mapsto 1] \models Pxy \wedge \neg Pyx \wedge \neg(x \equiv y)$
- Same as checking if $(\mathcal{M}, [x \mapsto 2, y \mapsto 1, _ \mapsto 1]) \models Pxy \wedge \neg Pyx \wedge \neg(x \equiv y)$
- Other cases also work out! So $\mathcal{F} \models \chi(x)$.
- 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 $\mathcal{F}[y \mapsto 1] \models Pxy \wedge \neg Pyx \wedge \neg(x \equiv y)$
- Same as checking if $(\mathcal{M}, [x \mapsto 2, y \mapsto 1, _ \mapsto 1]) \models Pxy \wedge \neg Pyx \wedge \neg(x \equiv y)$
- Other cases also work out! So $\mathcal{F} \models \chi(x)$.
- Let $\sigma'(x) = 2$ and $\sigma'(y) = 3$ for all other $y \in \mathcal{V}$. Does $(\mathcal{M}, \sigma') \models \chi(x)$?
- Let $\sigma''(x) = 3$ and $\sigma''(y) = 1$ for all other $y \in \mathcal{V}$. Does $(\mathcal{M}, \sigma'') \models \chi(x)$?
- **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}, σ) and (\mathcal{M}, σ') were models
- Only required that σ and σ' agreed on $\text{fv}(\chi(x))$
- Recall: only considered **PL** valuations restricted to atoms of expression
- **Theorem**: Let Σ be an FO signature and $\varphi \in \text{FO}_\Sigma$. Let \mathcal{M} be a Σ -structure and σ, σ' assignments which agree on $\text{fv}(\varphi)$. 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}, σ) and (\mathcal{M}, σ') were models
- Only required that σ and σ' agreed on $\text{fv}(\chi(x))$
- Recall: only considered **PL** valuations restricted to atoms of expression
- **Theorem**: Let Σ be an FO signature and $\varphi \in \text{FO}_\Sigma$. Let \mathcal{M} be a Σ -structure and σ, σ' assignments which agree on $\text{fv}(\varphi)$. 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 $\varphi \in \text{FO}_\Sigma$ **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 $\forall x_1 x_2 \dots x_n$ as shorthand for $\forall x_1. [\forall x_2. [\dots \forall x_n. [\dots] \dots]]$
- 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 \mathcal{V} to $T(\Sigma)$, 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 $\text{vars}(t) \cap \text{dom}(\theta) = \emptyset$
- Often write $t\theta$ to mean $\theta(t)$; $t\theta$ is a “substitution instance” of t
- We often write $\theta = \{t/x \mid x\theta = t \text{ and } x \in \text{dom}(\theta)\}$
- What effect does θ have on the semantics of expressions?
- **Theorem:** Given an interpretation $\mathcal{J} = ((M, \iota), \sigma)$ for some Σ , a term $t \in T(\Sigma)$, and a substitution $\{u/x\}$ such that $u^{\mathcal{J}} = m \in M$, it is the case that $(t\{u/x\})^{\mathcal{J}} = t^{\mathcal{J}[x \mapsto m]}$. Proof: **Exercise!**
- Lift to expressions as usual; ensure distinct bound and free variables.
- A substitution θ is **admissible** for an expression φ if $\text{vars}(\text{rng}(\theta)) \cap \text{vars}(\varphi) = \emptyset$.

Back to normal forms

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

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

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

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

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

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