

# CS228 Logic for Computer Science 2022

## Lecture 13: First-order logic

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2022-08-07

## Topic 13.1

### First-order logic (FOL) syntax

# First-order logic(FOL)

First-order logic(FOL)

=

propositional logic + quantifiers over individuals + functions/predicates

“First” comes from this property

## Example 13.1

*Consider argument: Humans are mortal. Socrates is a human. Therefore, Socrates is mortal.*

*In symbolic form,*

$\forall x.(H(x) \Rightarrow M(x)) \wedge H(s) \Rightarrow M(s)$

- ▶  $H(x)$  =  $x$  is a human
- ▶  $M(x)$  =  $x$  is mortal
- ▶  $s$  = Socrates

## A note on FOL syntax

The FOL syntax may appear **non-intuitive** and **cumbersome**.

FOL requires **getting used to it** like many other concepts such as **complex numbers**.

# Connectives and variables

**An** FOL consists of three disjoint kinds of symbols

- ▶ variables
- ▶ logical connectives
- ▶ non-logical symbols : function and predicate symbols

# Variables

We assume that there is a set  $\text{Vars}$  of variables, which is countably infinite in size.

- ▶ Since  $\text{Vars}$  is countable, we assume that variables are **indexed**.

$$\text{Vars} = \{x_1, x_2, \dots, \}$$

- ▶ The variables are just **names/symbols** without any inherent meaning
- ▶ We may also sometimes use  $x, y, z$  to denote the variables

Now forget all the definitions of the propositional logic. We will redefine everything and the new definitions will subsume the PL definitions.

## Logical connectives

The following are a finite set of symbols that are called **logical connectives**.

formal name	symbol	read as	
true	$\top$	top	} 0-ary
false	$\perp$	bot	
negation	$\neg$	not	} unary
conjunction	$\wedge$	and	} binary
disjunction	$\vee$	or	
implication	$\Rightarrow$	implies	
exclusive or	$\oplus$	xor	
equivalence	$\Leftrightarrow$	iff	} binary predicate
equality	$=$	equals	
existential quantifier	$\exists$	there is	} quantifiers
universal quantifier	$\forall$	for each	
open parenthesis	(		} punctuation
close parenthesis	)		
comma	,		

# Non-logical symbols

FOL is a parameterized logic

The parameter is a signature  $\mathbf{S} = (\mathbf{F}, \mathbf{R})$ , where

- ▶  $\mathbf{F}$  is a set of **function symbols** and
- ▶  $\mathbf{R}$  is a set of **predicate symbols** (aka **relational symbols**).

Each symbol is associated with an arity  $\geq 0$ .

We write  $f/n \in \mathbf{F}$  and  $P/k \in \mathbf{R}$  to explicitly state the arity

## Example 13.2

We may have  $\mathbf{F} = \{c/0, f/1, g/2\}$  and  $\mathbf{R} = \{P/0, H/2, M/1\}$ .

## Example 13.3

We may have  $\mathbf{F} = \{+/2, -/2\}$  and  $\mathbf{R} = \{</2\}$ .

**Commentary:** We are familiar with predicates, which are the things that are either true or false. However, the functions are the truly novel concept.



## Non-logical symbols (contd.)

**F** and **R** may either be finite or infinite.

Each **S** defines an FOL. We say, consider an FOL with signature  $\mathbf{S} = (\mathbf{F}, \mathbf{R})$  ...

We may not mention **S** if from the context the signature is clear.

### Example 13.4

*In the propositional logic,  $\mathbf{F} = \emptyset$  and*

$$\mathbf{R} = \{p_1/0, p_2/0, \dots\}.$$

# Constants and Propositional variable

There are special cases when the arity is zero.

$f/0 \in \mathbf{F}$  is called a **constant**.

$P/0 \in \mathbf{R}$  is called a **propositional variable**.

# Building FOL formulas

Let us use the ingredients to build the FOL formulas.

It will take a few steps to get there.

- ▶ terms
- ▶ atoms
- ▶ formulas

# Syntax : terms

Commentary: Terms are defined using grammar notation. If unfamiliar with the notation Please look [https://en.wikipedia.org/wiki/Formal\\_grammar](https://en.wikipedia.org/wiki/Formal_grammar)

## Definition 13.1

For signature  $\mathbf{S} = (\mathbf{F}, \mathbf{R})$ , **S-terms**  $T_{\mathbf{S}}$  are given by the following grammar:

$$t ::= x \mid f(\underbrace{t, \dots, t}_n),$$

where  $x \in \text{Vars}$  and  $f/n \in \mathbf{F}$ .

## Example 13.5

Consider  $\mathbf{F} = \{c/0, f/1, g/2\}$ . Let  $x_i$ s be variables. The following are terms.

- ▶  $f(x_1)$
- ▶  $g(f(c), g(x_2, x_1))$
- ▶  $c$
- ▶  $x_1$

You may be noticing some similarities between variables and constants

## Infix notation

We may write some functions and predicates in infix notation.

### Example 13.6

*we may write  $+(a, b)$  as  $a + b$  and similarly  $<(a, b)$  as  $a < b$ .*

## Syntax: atoms

### Definition 13.2

**S-atoms**  $A_S$  are given by the following grammar:

$$a ::= P(\underbrace{t, \dots, t}_n) \mid t = t \mid \perp \mid \top,$$

where  $P/n \in \mathbf{R}$ .

### Exercise 13.1

Consider  $\mathbf{F} = \{s/0\}$  and  $\mathbf{R} = \{H/1, M/1\}$ . Which of the following are atom?

▶  $H(x)$

▶  $M(s)$

▶  $s$

▶  $H(M(s))$

## Equality within logic vs. equality outside logic

We have an equality = within logic and the other when we use to talk about logic.

Both are distinct objects.

Some notations use same symbols for both and the others do not to avoid confusion.

Whatever is the case, we must be very clear about this.

## Syntax: formulas

### Definition 13.3

**S-formulas**  $P_S$  are given by the following grammar:

$$F ::= a \mid \neg F \mid (F \wedge F) \mid (F \vee F) \mid (F \Rightarrow F) \mid (F \Leftrightarrow F) \mid (F \oplus F) \mid \forall x.(F) \mid \exists x.(F)$$

where  $x \in \text{Vars}$ .

### Example 13.7

Consider  $\mathbf{F} = \{s/0\}$  and  $\mathbf{R} = \{H/1, M/1\}$

The following is a  $(\mathbf{F}, \mathbf{R})$ -formula:

$$\forall x.(H(x) \Rightarrow M(x)) \wedge H(s) \Rightarrow M(s)$$



## Unique parsing

For FOL we will ignore the issue of unique parsing,

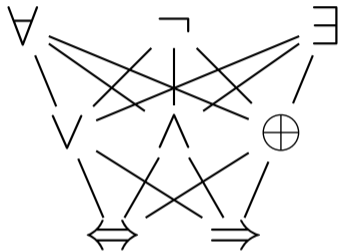
and assume

all the necessary precedence and associativity orders are defined

for ensuring human readability and unique parsing.

## Precedence order

We will use the following precedence order in writing the FOL formulas



### Example 13.8

*The following are the interpretation of the formulas after dropping parenthesis*

- ▶  $\forall x.H(x) \Rightarrow M(x) = \forall x.(H(x)) \Rightarrow M(x)$
- ▶  $\exists z\forall x.\exists y.G(x, y, z) = \exists z.(\forall x.(\exists y.G(x, y, z)))$

## Topic 13.2

### FOL - semantics

# Semantics : models

## Definition 13.4

For signature  $\mathbf{S} = (\mathbf{F}, \mathbf{R})$ , a **S-model**  $m$  is a

$$(D_m; \{f_m : D_m^n \rightarrow D_m \mid f/n \in \mathbf{F}\}, \{P_m \subseteq D_m^n \mid P/n \in \mathbf{R}\}),$$

where  $D_m$  is a nonempty set. Let **S-Mods** denotes the set of all **S-models**.

Some terminology

- ▶  $D_m$  is called **domain** of  $m$ .
- ▶  $f_m$  assigns meaning to  $f$  under model  $m$ .
- ▶ Similarly,  $P_m$  assigns meaning to  $P$  under model  $m$ .

**Commentary:** Models are also known as interpretations/structures.

## Example: model

### Example 13.9

Consider  $\mathbf{S} = (\{c/0, f/1, g/2\}, \{H/1, M/2\})$ .

Let us suppose our model  $m$  has domain  $D_m = \{\bullet, \bullet, \bullet\}$ .

We need to assign value to each function.

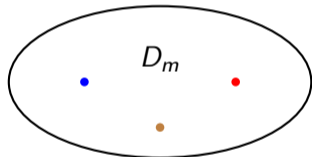
▶  $c_m = \bullet$

▶  $f_m = \{\bullet \mapsto \bullet, \bullet \mapsto \bullet, \bullet \mapsto \bullet\}$

▶  $g_m = \{(\bullet, \bullet) \mapsto \bullet, (\bullet, \bullet) \mapsto \bullet, (\bullet, \bullet) \mapsto \bullet,$   
 $(\bullet, \bullet) \mapsto \bullet, (\bullet, \bullet) \mapsto \bullet, (\bullet, \bullet) \mapsto \bullet,$   
 $(\bullet, \bullet) \mapsto \bullet, (\bullet, \bullet) \mapsto \bullet, (\bullet, \bullet) \mapsto \bullet\}$

We also need to assign values to each predicate.

▶  $H_m = \{\bullet, \bullet\}$        $M_m = \{(\bullet, \bullet), (\bullet, \bullet)\}$



### Exercise 13.2

a. How many models are there for the signature with the above domain?

b. Suppose  $P/0 \in \mathbf{R}$ , give a value to  $P_m$ .

# Semantics: assignments

Recall, We also have variables. Who will assign to the variables?

## Definition 13.5

An *assignment* is a map  $\nu : \text{Vars} \rightarrow D_m$

## Example 13.10

In our running example the domain is  $\mathbb{N}$ . We may have the following assignment.

$$\nu = \{x \mapsto 2, y \mapsto 3, \dots\}$$

**Commentary:**  $\nu$  is a function. It needs to map each variable. However, we only care about the mappings for variables that are relevant to our context. Therefore, in our slides we write mapping for only those variables that are important. For others, we assume there is some mapping.

## Semantics: term value

### Definition 13.6

For a model  $m$  and assignment  $\nu$ , we define  $m^\nu : T_{\mathbf{S}} \rightarrow D_m$  as follows.

$$\begin{aligned} m^\nu(x) &\triangleq \nu(x) && x \in \text{Vars} \\ m^\nu(f(t_1, \dots, t_n)) &\triangleq f_m(m^\nu(t_1), \dots, m^\nu(t_n)) \end{aligned}$$

### Example 13.11

Consider  $\mathbf{S} = (\{s/1, +/2\}, \{\})$  and term  $s(x) + y$

Consider model  $m = (\mathbb{N}; \text{succ}, +^{\mathbb{N}})$  and assignment  $\nu = \{x \mapsto 3, y \mapsto 2\}$

$$m^\nu(s(x) + y) = m^\nu(s(x)) +^{\mathbb{N}} m^\nu(y) = \text{succ}(m^\nu(x)) +^{\mathbb{N}} 2 = \text{succ}(3) +^{\mathbb{N}} 2 = 6$$

# Semantics: satisfaction relation

## Definition 13.7

We define the *satisfaction relation*  $\models$  among models, assignments, and formulas as follows

- ▶  $m, \nu \models \top$
- ▶  $m, \nu \models P(t_1, \dots, t_n)$       if  $(m^\nu(t_1), \dots, m^\nu(t_n)) \in P_m$
- ▶  $m, \nu \models t_1 = t_2$               if  $m^\nu(t_1) = m^\nu(t_2)$
- ▶  $m, \nu \models \neg F$                     if  $m, \nu \not\models F$
- ▶  $m, \nu \models F_1 \vee F_2$             if  $m, \nu \models F_1$  or  $m, \nu \models F_2$   
   skipping other propositional connectives
- ▶  $m, \nu \models \exists x.(F)$                 if there is  $u \in D_m : m, \nu[x \mapsto u] \models F$
- ▶  $m, \nu \models \forall x.(F)$                 if for each  $u \in D_m : m, \nu[x \mapsto u] \models F$



## Example: satisfiability

### Example 13.12

Consider  $\mathbf{S} = (\{s/1, +/2\}, \{\})$  and formula  $\exists z.s(x) + y = s(z)$

Consider model  $m = (\mathbb{N}; \text{succ}, +^{\mathbb{N}})$  and assignment  $\nu = \{x \mapsto 3, y \mapsto 2\}$

We have seen  $m^{\nu}(s(x) + y) = 6$ .

$$m^{\nu[z \mapsto 5]}(s(x) + y) = m^{\nu}(s(x) + y) = 6.$$

//Since  $z$  does not occur in the term

$$m^{\nu[z \mapsto 5]}(s(z)) = 6$$

Therefore,  $m, \nu[z \mapsto 5] \models s(x) + y = s(z)$ .

$$m, \nu \models \exists z.s(x) + y = s(z).$$

# Satisfiable, true, valid, and unsatisfiable

We say

- ▶  $F$  is *satisfiable* if there are  $m$  and  $\nu$  such that  $m, \nu \models F$
- ▶ Otherwise,  $F$  is called unsatisfiable (written  $\not\models F$ )
- ▶  $F$  is *true* in  $m$  ( $m \models F$ ) if for all  $\nu$  we have  $m, \nu \models F$
- ▶  $F$  is *valid* ( $\models F$ ) if for all  $\nu$  and  $m$  we have  $m, \nu \models F$

## Exercise: model

Consider  $\mathbf{S} = (\{c/0, f/1\}, \{H/1, M/2\})$ . Let us suppose model  $m$  has  $D_m = \{\bullet, \bullet, \bullet\}$  and the values of the symbols in  $m$  are

▶  $c_m = \bullet$

▶  $f_m = \{\bullet \mapsto \bullet, \bullet \mapsto \bullet, \bullet \mapsto \bullet\}$

▶  $H_m = \{\bullet, \bullet\}$

▶  $M_m = \{(\bullet, \bullet), (\bullet, \bullet)\}$

### Exercise 13.3

*Which of the following hold?*

▶  $m, \{x \mapsto \bullet\} \models M(f(x), x)$

▶  $m, \{\} \models \exists x.H(x)$

▶  $m, \{\} \models \exists x.H(f(x))$

▶  $m, \{x \mapsto \bullet\} \models H(x)$

▶  $m, \{\} \models \forall x.H(x)$

▶  $m, \{\} \models H(c)$

## Extended satisfiability (repeat from propositional logic)

We extend the usage of  $\models$ . Let  $\Sigma$  be a (possibly infinite) set of formulas.

### Definition 13.8

$m, \nu \models \Sigma$  if  $m, \nu \models F$  for each  $F \in \Sigma$ .

### Definition 13.9

$\Sigma \models F$  if for each model  $m$  and assignment  $\nu$  if  $m, \nu \models \Sigma$  then  $m, \nu \models F$ .

$\Sigma \models F$  is read  $\Sigma$  implies  $F$ . If  $\{G\} \models F$  then we may write  $G \models F$ .

### Definition 13.10

Let  $F \equiv G$  if  $G \models F$  and  $F \models G$ .

### Definition 13.11

Formulas  $F$  and  $G$  are *equisatisfiable* if

$F$  is sat iff  $G$  is sat.

**Commentary:** These definitions are identical to the propositional case.

# Topic 13.3

## Problems

## Exercise 13.4

*Give the restrictions on FOL such that it becomes the propositional logic. Give an example of FOL model of a non-trivial propositional formula.*

# Valid formulas

## Exercise 13.5

*Prove/Disprove the following formulas are valid.*

- ▶  $\forall x.P(x) \Rightarrow P(c)$
- ▶  $\forall x.(P(x) \Rightarrow P(c))$
- ▶  $\exists x.(P(x) \Rightarrow \forall x.P(x))$
- ▶  $\exists y\forall x.R(x, y) \Rightarrow \forall x\exists y.R(x, y)$
- ▶  $\forall x\exists y.R(x, y) \Rightarrow \exists y\forall x.R(x, y)$

# Properties of FOL

## Exercise 13.6

Show the validity of the following formulas.

1.  $\neg\forall x. P(x) \Leftrightarrow \exists x. \neg P(x)$
2.  $\neg\exists x. P(x) \Leftrightarrow \forall x. \neg P(x)$
3.  $(\forall x. (P(x) \wedge Q(x))) \Leftrightarrow \forall x. P(x) \wedge \forall x. Q(x)$
4.  $(\exists x. (P(x) \vee Q(x))) \Leftrightarrow \exists x. P(x) \vee \exists x. Q(x)$

## Exercise 13.7

Show  $\forall$  does not distribute over  $\vee$ .

Show  $\exists$  does not distribute over  $\wedge$ .



## Example: non-standard models

Consider  $\mathbf{S} = (\{0/0, s/1, +/2\}, \{\})$  and formula  $\exists z.s(x) + y = s(z)$

**Unexpected model:** Let  $m = (\{a, b\}^*; \epsilon, \text{append}_a, \text{concat})$ .

- ▶ The domain of  $m$  is the set of all strings over alphabet  $\{a, b\}$ .
- ▶  $\text{append}_a$ : appends  $a$  in the input and
- ▶  $\text{concat}$ : joins two strings.

Let  $\nu = \{x \mapsto ab, y \mapsto ba\}$ .

Since  $m, \nu[z \mapsto abab] \models s(x) + y = s(z)$ , we have  $m, \nu \models \exists z.s(x) + y = s(z)$ .

### Exercise 13.8

- ▶ Show  $m, \nu[y \mapsto bb] \not\models \exists z.s(x) + y = s(z)$
- ▶ Give an assignment  $\nu$  s.t.  $m, \nu \models x \neq 0 \Rightarrow \exists y. x = s(y)$ .  
Show  $m \not\models \forall x. (x \neq 0 \Rightarrow \exists y. x = s(y))$ .

# Find models

## Exercise 13.9

For each of the following formula give a model that satisfies the formula. If there is no model that satisfies a formula, then report that the formula is unsatisfiable.

1.  $\forall x. \exists y R(x, y) \wedge \neg \exists x. \forall y R(x, y)$
2.  $\neg \forall x. \exists y R(x, y) \wedge \exists x. \forall y R(x, y)$
3.  $\neg \forall x. \exists y R(x, y) \wedge \neg \exists x. \forall y R(x, y)$
4.  $\forall x. \exists y R(x, y) \wedge \exists x. \forall y R(x, y)$

# Similar quantifiers

## Exercise 13.10

Show using FOL fol semantics.

- ▶  $\exists x.\exists x. F \equiv \exists x. F$
- ▶  $\exists x.\exists y. F \equiv \exists y.\exists x. F$
- ▶  $\forall x.\forall x. F \equiv \forall x. F$
- ▶  $\forall x.\forall y. F \equiv \forall y.\forall x. F$

## Exercise : compact notation for terms

Since we know arity of each symbol, we need not write “,” “(”, and “)” to write a term unambiguously.

### Example 13.13

$f(g(a, b), h(x), c)$  can be written as  $fgabhxc$ .

### Exercise 13.11

Consider  $\mathbf{F} = \{f/3, g/2, h/1, c/0\}$  and  $x, y \in \text{Vars}$ .

Insert parentheses at appropriate places in the following if they are valid term.

▶  $hc =$

▶  $fhxhyhc =$

▶  $gxc =$

▶  $fx =$

### Exercise 13.12

Give an algorithm to insert the parentheses

## Exercise: DeBruijn index of quantified variables

DeBruijn index is a method for representing formulas without naming the quantified variables.

### Definition 13.12

Each *DeBruijn index* is a natural number that represents an occurrence of a variable in a formula, and denotes the number of quantifiers that are in scope between that occurrence and its corresponding quantifier.

### Example 13.14

We can write  $\forall x.H(x)$  as  $\forall.H(1)$ . **1** is indicating the occurrence of a quantified variable that is bounded by the closest quantifier. More examples.

- ▶  $\exists y\forall x.M(x, y) = \exists\forall.M(1, 2)$
- ▶  $\exists y\forall x.M(y, x) = \exists\forall.M(2, 1)$
- ▶  $\forall x.(H(x) \Rightarrow \exists y.M(x, y)) = \forall.(H(1) \Rightarrow \exists.M(2, 1))$

### Exercise 13.13

Give an algorithm that translates FOL formulas into DeBurjin indexed formulas.

# Drinker paradox

## Exercise 13.14

*Prove*

*There is someone  $x$  such that if  $x$  drinks, then everyone drinks.*

Let  $D(x) \triangleq x$  drinks. Formally

$$\exists x.(D(x) \Rightarrow \forall x. D(x))$$

[https://en.wikipedia.org/wiki/Drinker\\_paradox](https://en.wikipedia.org/wiki/Drinker_paradox)

## Exercise: satisfaction relation

### Exercise 13.15

Consider  $\mathbf{S} = (\{\cup/2\}, \{\in /2\})$  and formula  $F = \exists x. \forall y. \neg y \in x$  (what does it say to you!)

Consider  $\mathbf{S}$ -model  $m = (\mathbb{N}; \cup_m = \max, \in_m = \{(i, j) \mid i < j\})$  and  $\nu = \{x \mapsto 2, y \mapsto 3\}$ .

$m, \nu \models F?$

## Exercise: implication

### Exercise 13.16

Let us suppose the following formula is valid and  $\Sigma$  does not refer to  $c$ .

$$\Sigma \Rightarrow H(f(c)) \wedge \neg H(f(a))$$

Prove that  $\Sigma$  is unsatisfiable.



## Topic 13.4

Extra slides: some properties of models

# Homomorphisms of models

## Definition 13.13

Consider  $\mathbf{S} = (\mathbf{F}, \mathbf{R})$ . Let  $m$  and  $m'$  be  $\mathbf{S}$ -models.

A function  $h : D_m \rightarrow D_{m'}$  is a **homomorphism** of  $m$  into  $m'$  if the following holds.

- ▶ for each  $f/n \in \mathbf{F}$ , for each  $(d_1, \dots, d_n) \in D_m^n$

$$h(f_m(d_1, \dots, d_n)) = f_{m'}(h(d_1), \dots, h(d_n))$$

- ▶ for each  $P/n \in \mathbf{R}$ , for each  $(d_1, \dots, d_n) \in D_m^n$

$$(d_1, \dots, d_n) \in P_m \quad \text{iff} \quad (h(d_1), \dots, h(d_n)) \in P_{m'}$$

## Definition 13.14

A homomorphism  $h$  of  $m$  into  $m'$  is called **isomorphism** if  $h$  is one-to-one.

$m$  and  $m'$  are called **isomorphic** if an  $h$  exists that is also onto.

## Example : homomorphism

### Example 13.15

Consider  $\mathbf{S} = (\{+/2\}, \{\})$ .

Consider  $m = (\mathbb{N}, +^{\mathbb{N}})$  and  $m' = (\mathcal{B}, \oplus^{\mathcal{B}})$ ,

$h(n) = n \bmod 2$  is a homomorphism of  $m$  into  $m'$ .

# Homomorphism theorem for terms and quantifier-free formulas without =

## Theorem 13.1

Let  $h$  be a homomorphism of  $m$  into  $m'$ . Let  $\nu$  be an assignment.

1. For each term  $t$ ,  $h(m^\nu(t)) = m'^{\nu \circ h}(t)$
2. If formula  $F$  is quantifier-free and has no symbol “=”

$$m^\nu \models F \quad \text{iff} \quad m'^{\nu \circ h} \models F$$

Proof.

Simple structural induction. □

## Exercise 13.17

For a quantifier-free formula  $F$  that may have symbol “=”, show

$$\text{if } m^\nu \models F \quad \text{then} \quad m'^{\nu \circ h} \models F$$

Why the reverse direction does not work?

## Homomorphism theorem with =

### Theorem 13.2

Let  $h$  be a homomorphism of  $m$  into  $m'$ . Let  $\nu$  be an assignment. If  $h$  is isomorphism then the reverse implication also holds for formulas with “=”.

### Proof.

Let us suppose  $m^{(\nu \circ h)} \models s = t$ .

Therefore,  $m^{(\nu \circ h)}(s) = m^{(\nu \circ h)}(t)$ .

Therefore,  $h(m^\nu(s)) = h(m^\nu(t))$ .

Due to the one-to-one condition of  $h$ ,  $m^\nu(s) = m^\nu(t)$ .

Therefore,  $m^\nu \models s = t$ . □

### Exercise 13.18

For a formula  $F$  (with quantifiers) without symbol “=”, show

$$\text{if } m^{(\nu \circ h)} \models F \text{ then } m^\nu \models F.$$

**Commentary:** Note that that implication direction is switched from the previous exercise.

# Homomorphism theorem with quantifiers

## Theorem 13.3

Let  $h$  be an isomorphism of  $m$  into  $m'$  and  $\nu$  be an assignment.

If  $h$  is also onto, the reverse direction also holds for the quantified formulas.

### Proof.

Let us assume,  $m^\nu \models \forall x.F$ .

Choose  $d' \in D_{m'}$ .

Since  $h$  is onto, there is a  $d$  such that  $d = h(d')$ .

Therefore,  $m^{\nu[x \mapsto d]} \models F$ .

Therefore,  $m'^{\nu[x \mapsto d']} \models F$ .

Therefore,  $m'^{(\nu \circ h)} \models \forall x.F$ . □

## Theorem 13.4

If  $m$  and  $m'$  are isomorphic then for all sentences  $F$ ,  $m \models F$  iff  $m' \models F$ .

Commentary: The reverse direction of the above theorem is not true.

## Expressive/distinguishing power of FOL

If two models are isomorphic, then no two formulas can separate them.

This is not a limitation of FOL.

It is not the case that if we add more features in the logic, we can distinguish the models.

Therefore, one may view isomorphic models as same models.

End of Lecture 13