First Order Logic – Semantics (3A)

Copyright (c) 2016 - 2017 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice

Contemporary Artificial Intelligence, R.E. Neapolitan & X. Jiang

Logic and Its Applications, Burkey & Foxley

A Signature and a Language

First specify a **signature**

Constant Symbols Predicate Symbols Function Symbols

$$\{C_{1}, C_{2}, \dots C_{n}\} = D$$

$$\{P_{1}, P_{2}, \dots P_{m}\}$$

$$\{f_{1}, f_{2}, \dots f_{l}\}$$

Determines the language

Given a language A **model** is specified A **domain of discourse** a set of entities An **interpretation** constant assignments function assignments truth value assignments

 $\{\text{entity}_1, \text{entity}_2, \dots \text{entity}_n\}$

$$\{C_{1}, C_{2}, \dots C_{n}\} = D$$

$$f_{1}(), f_{2}(), \dots f_{l}()$$

$$P_{1}(), P_{2}(), \dots P_{m}()$$

First Order Logic (3A) Semantics Young Won Lim 6/6/17

Model – domain of discourse

- 1. a nonempty set D of **entities** called a **domain of discourse**
 - this domain is a <u>set</u>
 - each <u>element</u> in the set : <u>entity</u>
 - each constant symbol : one entity in the domain

If we considering all individuals in a class, The constant symbols might be

- 'Mary', an entity
- 'Fred', an entity
- 'John', an entity
- 'Tom' an entity

2. an interpretation

(a) an <u>entity</u> in D is assigned to each of the <u>constant symbols</u>.

Normally, every entity is assigned to a constant symbol.

(b) for each function,

an <u>entity</u> is assigned to each possible <u>input of entities</u> to the **function**

(c) the predicate 'True' is always assigned the value T

The predicate 'False' is always assigned the value F

(d) for every other predicate,

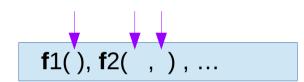
the value T or F is assigned

to each possible input of entities to the predicate

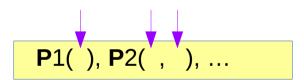


{entity₁, entity₂, ... entity_n} { $c_1, c_2, ... c_n$ } = D

Function assignments

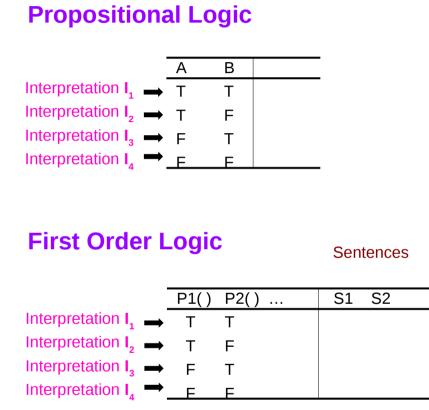


Truth value assignments

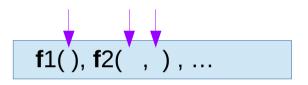


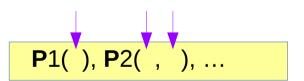
always return T / F

Interpretation



{entity₁, entity₂, ... entity_n} { $c_1, c_2, ... c_n$ } = D

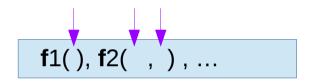




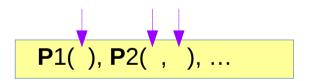
always return T / F

Arity one:C(n, 1)Arity two:C(n, 2)Arity three:C(n, 3)

{entity₁, entity₂, ... entity_n} { $c_1, c_2, ... c_n$ } = D



Arity one functions & predicates: Arity two: Arity three: C(n, 1) C(n, 2) C(n, 3)



always return T / F

• • •

. . .

Constant assignments

(a) an <u>entity</u> \rightarrow the <u>constant symbols</u>.

Function assignments

(b) an <u>entity</u> \rightarrow each possible <u>input of entities</u> to the **function**

Truth value assignments

(c) the value $T \rightarrow$ the predicate '**True**' the value $F \rightarrow$ the predicate '**False**'

(d) for every other **predicate**,

the value T or F is assigned \rightarrow every other predicate to each possible <u>input of entities</u> to the **predicate**

Signature Model Examples A - (1)

Signature

- 1. <u>constant symbols</u> = { Mary, Fred, Sam }
- 2. predicate symbols = { married, young }
 married(x, y) : arity two
 young(x) : arity one

Model

- 1. domain of discourse D : the set of three particular individuals
 - this domain is a <u>set</u>
 - each <u>element</u> in the set : <u>entity (= individuals)</u>
 - each <u>constant symbol</u> : one <u>entity</u> in the domain <u>(= one individual)</u>

2. interpretation

(a) a different individual is assigned to each of the constant symbols

(a) an <u>entity</u> in D is assigned to each of the <u>constant symbols</u>. Normally, every entity is assigned to a constant symbol.

Signature Model Examples A - (2)

(b) for each **function**, an <u>entity</u> is assigned to each possible <u>input of entities</u> to the **function**

(c) the predicate '**True**' is always assigned the value T The predicate '**False**' is always assigned the value F

(d) the truth value assignments for every predicate

young(Mary) = F, young(Fred) = F, young(Sam) = T

married(Mary, Mary) = F, married(Mary, Fred) = T, married(Mary, Sam) = F married(Fred, Mary) = T, married(Fred, Fred) = F, married(Fred, Sam) = F married(Sam, Mary) = F, married(Sam, Fred) = F, married(Sam, Sam) = F

 (d) for every other predicate, the value T or F is assigned to each possible <u>input of entities</u> to the predicate

> (Mary, Mary), (Mary, Fred), (Mary, Sam) (Fred, Mary), (Fred, Fred), (Fred, Sam) (Sam, Mary), (Sam, Fred), (Sam, Sam)

Signature Model Examples B – (1)

Signature

- 1. <u>constant symbols</u> = { Fred, Mary, Sam }
- 2. <u>predicate symbols</u> = { love } love(x, y) : arity two
- 3. <u>function symbols</u> = { mother } mother(x) : arity one

Model

- 1. domain of discourse D : the set of three particular individuals
- 2. interpretation

(a) a different individual is assigned to each of the constant symbols

(b) the truth value assignments for every predicate love(Fred, Fred) = F, love(Fred, Mary) = F, love(Fred, Ann) = F love(Mary, Fred) = T, love(Mary, Mary) = F, love(Mary, Ann) = T love(Ann, Fred) = T, love(Ann, Mary) = T, love(Ann, Ann) = F

(c) the function assignments mother(Fred) = Mary, mother(Mary) = Ann, mother(Ann) = - (no assignment)

Signature Model Examples B – (2)

2. interpretation

(a) a different individual is assigned to each of the constant symbols

(a) an <u>entity</u> in D is assigned to each of the <u>constant symbols</u>. Normally, every entity is assigned to a constant symbol.

(b) the truth value assignments

(b) for each function,

an entity is assigned to each possible input of entities to the function

love(Fred, Fred) = F, love(Fred, Mary) = F, love(Fred, Ann) = F love(Mary, Fred) = T, love(Mary, Mary) = F, love(Mary, Ann) = T love(Ann, Fred) = T, love(Ann, Mary) = T, love(Ann, Ann) = F

(c) the function assignments

 (d) for every other predicate, the value T or F is assigned to each possible input of entities to the predicate

mother(Fred) = Mary, mother(Mary) = Ann, mother(Ann) = - (no assignment)

The truth values of **all sentences** are assigned :

1. the truth values for sentences developed with the symbols \neg , \land , \lor , \Rightarrow , \Leftrightarrow are assigned as in propositional logic.

2. the truth values for two terms connected by the = symbol is T if both terms refer to the same entity; otherwise it is F

3. the truth values for $\forall x p(x)$ has value T if p(x) has value T for every assignment to x of an entity in the domain D; otherwise it has value F

4. the truth values for $\exists x p(x)$ has value T if p(x) has value T for at least one assignment to x of an entity in the domain D; otherwise it has value F

5. the operator **precedence** is as follows \neg , =, \land , \lor , \Rightarrow , \Leftrightarrow

6. the **quantifiers** have precedence over the operators

7. **parentheses** change the order of the precedence

Formulas and Sentences

An formula

- A atomic formula
- The operator ¬ followed by a **formula**
- Two formulas separated by Λ , \forall , \Rightarrow , \Leftrightarrow
- A quantifier following by a variable followed by a formula

A sentence

- A formula with no free variables.
- $\forall x \text{ tall}(x)$: no free variable : a sentence
- $\forall x \text{ love}(x, y)$: free variable y : not a sentence

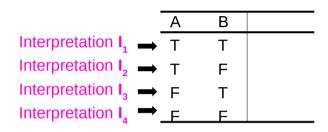
Finding the truth value

Find the truth values of all sentences

- 1. ¬, Λ , V, \Rightarrow , \Leftrightarrow
- 2. = symbol
- 3. ∀x p(x)
- 4. ∃x p(x)
- 5. the operator precedence is as follows \neg , =, \land , \lor , \Rightarrow , \Leftrightarrow
- 6. the **quantifiers** (\forall , \exists) have precedence over the **operators**
- 7. parentheses change the order of the precedence

Truth values of sentences

Propositional Logic



First Order Logic

Sentences

	P1()	P2()	S1	S2
Interpretation $I_1 \longrightarrow$	Т	Т		
Interpretation $I_2 \implies$	Т	F		
Interpretation $I_3 \rightarrow$	F	т		
Interpretation $I_4 \rightarrow$	F	F		

1. ¬, Λ , V, \Rightarrow , \Leftrightarrow

- 2. = symbol
- 3. ∀x p(x)
- 4. ∃x p(x)
- 5. operator precedence
- 6. quantifiers (\forall, \exists) high recedence
- 7. parentheses change the order

An formula

- A atomic formula
- The operator ¬ followed by a formula
- Two formulas separated by Λ , V, \Rightarrow , \Leftrightarrow
- A quantifier following by a variable followed by a formula

A sentence

A formula with no free variables

First Order Logic (3A) Semantics

Sentence Examples (1)

Signature

```
Constant Symbols = {Socrates, Plato, Zeus, Fido}
Predicate Symbols = {human, mortal, legs} all arity one
```

Model

D: the set of these four particular individuals

Interpretation

(a) a different individual is assigned to each of the constant symbols

(b) the truth value assignment

human(Socrates)=T, human(Plato)=T, human(Zeus)=F, human(Fido)=F mortal(Socrates)=T, mortal(Plato)=T, mortal(Zeus)=F, mortal(Fido)=T legs(Socrates)=T, legs(Plato)=T, legs(Zeus)=T, legs(Fido)=T

Sentence Examples (2)

Sentence 1: human(Zeus) Ahuman(Fido) Vhuman(Socrates) = T F Т Sentence 2: human(Zeus) human(Zeus) human(Socrates)) = F F Sentence 3: $\forall x human(x) = F$ human(Zeus)=F, human(Fido)=F Sentence 4: $\forall x \text{ mortal}(x) = F$ mortal(Zeus)=F Sentence 5: $\forall x \text{ legs}(x) = T$ legs(Socrates)=T, legs(Plato)=T, legs(Zeus)=T, legs(Fido)=T Sentence 6: $\exists x human(x) = T$ human(Socrates)=T, human(Plato)=T

Sentence 7: $\forall x (human(x) \Rightarrow mortal(x)) = T$

Sentence Examples (3)

```
Sentence 7: \forall x (human(x) \Rightarrow mortal(x)) = T
```

human(Socrates)=T,	mortal(Socrates)=T,	$T \Rightarrow T$: T
human(Plato)=T,	mortal(Plato)=T,	$T \Rightarrow T$: T
human(Zeus)=F,	mortal(Zeus)=F,	$F \Rightarrow F$: T
human(Fido)=F	mortal(Fido)=T	$F \Rightarrow T$: T

References

- [1] en.wikipedia.org
- [2] en.wiktionary.org
- [3] U. Endriss, "Lecture Notes : Introduction to Prolog Programming"
- [4] http://www.learnprolognow.org/ Learn Prolog Now!
- [5] http://www.csupomona.edu/~jrfisher/www/prolog_tutorial
- [6] www.cse.unsw.edu.au/~billw/cs9414/notes/prolog/intro.html
- [7] www.cse.unsw.edu.au/~billw/dictionaries/prolog/negation.html
- [8] http://ilppp.cs.lth.se/, P. Nugues,`An Intro to Lang Processing with Perl and Prolog