First Order Logic (a.k.a. Predicate Calculus)

Andrew W. Moore

16-731 Fall 1998

Handout AWM2

See also chapter 7 of R&N

DISASTER!!!!: Most of the slides HAND-DRAWN, not on line. Copies will be outside AWM's office (Smith 221)

First Order Logic (with Equality)

Propositional logic can talk about: atomic facts.

First order logic can talk about: objects, properties of objects, and relations between objects (including functions).

Examples of ..

- Objects: people, numbers, states, days-of-week, Dan Quayle, the moon, the smallest onion in Chatanooga.
- Relations: comes before, stupider than, brother of, bigger than, beats
- Properties: stupid, tall, blue
- Functions: father of, inverse of, favorite wild flower of.

First Order Logic Syntax

Sentence ::= AtomicSentence | ComplexSentence

AtomicSentence ::= Predicate (Term , ...) | Term = Term

Term ::= Function (Term , ...) | Constant | Variable

ComplexSentence ::= (Sentence) |

Sentence Connective Sentence |

~Sentence |

Quantifier Variable Sentence

Connective ::= $\wedge$ | $\vee$ | <=> | =>

Quantifier ::= “ | $

Constant ::= A | B | .. | X | X1 | X2 .. | Andrew | Pat...

Variable ::= a | b | c | ... x | y | z

Predicate ::= Before | HasColor | Raining | AreBrothers ...

Function ::= Mother | LeftLegOf | log | Height ...

First Order Logic: examples of legal sentences

$\forall m. \forall c. Mother(c) = m <=> Female(m) \wedge Parent(m,c)$

$\forall w. \forall h. Husband(h,w) <=> Male(h) \wedge Spouse(h,w)$

Father(Andrew) = Michael

~SitsInChair(Lecturer)

$\forall x. Female(x) <=> \sim Male(x)$

Female(Mary)

$\forall x. \forall y. Sibling(x,y) <=> \sim(x = y) \wedge \exists p . Parent(p ,x) \wedge Parent(p ,y)$

Now let’s do some semantics. Remember how we did this for Propositional logic.

- First we define what is an interpretation
- Then we describe how to evaluate a sentence under a given interpretation
- Then we say the meaning (ie semantics) of a sentence is the set of interpretations in which the sentence evaluates to True.

Constant Symbols in Interpretations

e.g. *Andrew* , *Michael*, *Lecturer* , *Mary*

An interpretation does the following:

With each Constant symbol, it associates an object. Some constant symbols may be given the same object.

One interpretation of the Constant Symbols *Andrew Michael* and *Lecturer* would tie *Andrew* and *Lecturer* to the same object, *Michael* to a different object and *Mary* to yet another object.

I'm going to teach you something that's wrong

Sorry.

When we look at examples of interpretations in the next few slides, you'll notice that I implicitly assume that each constant symbol is associated with its own distinct object.

So we won't consider, say, interpretations where Andrew and Lecturer are bound to the same object.

What are the consequences?

Good thing: It means we can look at examples of interpretations without getting utterly swallowed up in complexity.

Bad thing: It won't emphasize just how many different interpretations there can be.

Predicate Symbols

$\forall m. \forall c.\ Mother(c) = m \Leftrightarrow Female(m) \wedge Parent(m,c)$

$\forall w. \forall h.\ Husband(h,w) \Leftrightarrow Male(h) \wedge Spouse(h,w)$

Father(Andrew) = Michael

~SitsInChair(Lecturer)

$\forall x.\ Female(x) \Leftrightarrow \sim Male(x)$

Female(Mary)

OlderThan(Andrew, Mary)

$\forall x. \forall y.\ Sibling(x,y) \Leftrightarrow \sim(x = y) \wedge \exists p.\ Parent(p,x) \wedge Parent(p,y)$

In an interpretation, each Predicate symbol is specified by a set of Tuples. (a Predicate with N arguments is a set of N-tuples).

Example. One interpretation could have:

Female = { (Mary) }

OlderThan = { (Andrew, Mary), (Michael, Andrew), (Michael, Lecturer), (Lecturer, Mary) }

Sibling = { }

Function Symbols

Father(Andrew) = Michael

In an interpretation, each function symbol is specified by a set of tuples. A function with N arguments is a set of N+1-tuples in which each possible combination of N constant symbols appears exactly once among the first N elements.

Example. One interpretation could have:

Father = {(Andrew, Michael), (Michael, Mary), (Mary, Michael), (Lecturer, Michael)}

With four constant symbols, a function symbol that took two arguments would need sixteen 3-tuples in the interpretation.

E.G: AgeDifference(Andrew, Mary) = 1

AgeDifference = {(Andrew, Andrew, 0), (Andrew, Mary, 1), (Andrew, Michael, 25), (Andrew, Lecturer, 0), (Mary, Andrew, 1), (Mary, Mary, 0), ... }

Terms

Term ::= Function (Term , ...) | Constant | Variable

In a given interpretation a Term is evaluated as follows.

- If it's a constant, merely return the object associated with that constant in the interpretation.
- If it's a variable ....see later
- If it's a function applied to arguments,
  - Evaluate all the arguments in the current interpretation.
  - Search (among the tuples associated this function symbol) for the tuple which has the same first N elements as the N argument values.
  - Once you've found the one that matches (there must be precisely one, according to the definition of the interpretation of a function symbol) return the N+1'th element of that tuple.

Example

Assume in interpretation I that

Father = {(Andrew,Michael), (Michael,Mary), (Mary, Michael), (Lecturer,Michael)}

AgeDifference = {(Andrew,Andrew,0), (Andrew,Mary,1), (Andrew,Michael,25), (Andrew,Lecturer,0), (Mary,Andrew,1), (Mary,Mary,0), ... }

Suppose you are asked to evaluate

AgeDifference(Andrew,Father(Andrew))

in interpretation I. What would you reply?

`WARNING: REMAINDER OF PAGES CONTINUE AS HAND-WRITTEN SLIDES........`