First Order Logic– Arguments (5A)

Young W. Lim
1/11/17

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

This document was produced by using LibreOffice

Based on

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

Logic and Its Applications,
Burkey & Foxley

Arguments

An argument consists of set of formulas, called the premises
And a formula called the conclusion

The premises entail the conclusion
If every model in which all the premises are true,
The conclusion is also true.

If the premises entail the conclusion,
The argument is sound
Otherwise, it is a fallacy

A set of inference rules : a deductive system
A deductive system is sound if it only derives sound arguments
A deductive system is complete if it can derive every sound argument

Universal Instantiation

$\forall x\ A(x) \models A(t)$

t is any term

If A(x) has value T for all entities in the domain of discourse,
then it must have value T for term t

Universal Generalization

A(e) for every entity e in the domain of discourse) $\models$ $\forall x\ A(x)$

If A(e) has value T for every entity e,
Then $\forall x\ A(x)$ has value T

The rule is ordinarily applied by showing that A(e) has value T
for an aribary entity e

Existential Generalization

$A(e) \models \exists x\, A(x)$
Where e is an entity in the domain of discourse

If A(e) has value T for some entity e,
Then $\exists x\, A(x)$ has value T

Existential Instantiation

$\exists x\, A(x) \models A(e)$
Where e is an entity in the domain of discourse

If $\exists x\, A(x)$ has value T,
then A(e) has value T for some entity e

Unification

Two sentences A and B
A unification of A and B
A substitution θ of values for some of the variables in A and B
That make the sentences identical
The set of substitutions θ is called the unifier θ

Most General Unifier

If every other unifier θ' is an instance of θ in the sense that θ' can be derived by making substitutions in θ

θ

Unification Algorithm

Input : Two sentences A and B; an empty set of substitution theta
Output : a most general unifier of the sentences if they can be unified;
otherwise failure.

```
Procedure unify (A, B, var theta)          θ
Scan A and B from left to right
   Until A and B disagree on a symbol or A and B are exhausted
If A and B are exhausted
  Let x and y be the symbols where A and B disagree
  If x is a variable
    Theta = theta U {x/y}
    unify(subst(theta, A), subst(theta, B), theta)
  Else if y is a variable
    Theta = theta U {y/x}
    unify(subst(theta, A), subst(theta, B), theta)
  Else
    Theta = failure;
  Endif
endif
```

Generalized Modus Ponens

Suppose we have sentences A, B, and C, and the the sentence A $\Rightarrow$ B, which is implicitly universally quantified for all variables in the sentence.

The Generalized Modus Ponens (GMP) rule is as follows

A $\Rightarrow$ B, C, unify(A, C, θ) $\models$ subst(B, θ)

Logical Equivalences

¬, ∧, ∨

¬, ∧, ∨

∧∨⌐¬⇒

⇔≡⇛⊨

∧∨⌐¬⇒

⇔≡⇛⊨

⇒

⇔

≡

⇒

⇔

≡

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