Homework 5: First-Order Logic

Due: 5:00 PM, Mar 16, 2009

Contents

1 Foolish Apples **1**

2 Unification **2**

3 List Axioms **2**

4 Multiple Choice **2**

5 You've Got A Friend **3**

6 Prove It If You Can **3**

7 B & F Chaining **3**

Objectives

By the end of this homework, you will be able to:

1. express English sentences in first-order logic
2. unify two expressions of first-order logic
3. convert sentences of first-order logic to normal form
4. prove or disprove sentences of first-order logic using resolution

Practice

1 Foolish Apples

(a) Translate the following sentence of first-order logic into English:

$$(\forall d \ \text{EatApple}(d)) \rightarrow (\neg \exists d \ \text{SeeDoctor}(d))$$

(b) Translate the following English sentence into first-order logic: You can fool some of the people all the time, and all the people some of the time, but you can't fool all the people all the time.

2 Unification

Task: Unify the following pairs of expressions, or state why no unifier exists:

1. foo(x,y), foo(y,x)
2. mother(x,y), mother(y,father(x))
3. $p(x, y, z)$, $p(q(y), r(z), \text{foo})$

3 List Axioms

Characterize the following list axioms in first-order logic by defining the predicates LIST, MEMBER, APPEND, REVERSE, and LENGTH. Construct terms using the constant NIL and the function CONS.

1. A LIST is either NIL or the result of applying CONS to element x and list l.
2. The FIRST element in a non-NIL list is precisely the first element in the list. *i.e.*, the parameter of the CONS operation by which the list is created.
3. The REST of a list is what remains after the FIRST element is removed.
4. An element is a MEMBER of a non-NIL list iff it appears in the list.
5. The APPEND predicate relates two lists to a third, their concatenation.
6. The REVERSE of a list is a list in which the order of the elements is reversed.
7. The LENGTH of a list is its number of elements.

Problems

4 Multiple Choice

Task: State whether the following logical statements are valid, merely satisfiable, or altogether unsatisfiable. If the sentence is satisfiable, give a model: *i.e.*, a satisfying interpretation.

- P
- $(P \rightarrow \neg P) \wedge (\neg P \rightarrow P)$
- $(P \rightarrow Q) \wedge (Q \rightarrow R) \wedge (R \rightarrow P)$
- $P(f(a)) \rightarrow \exists x\, P(f(x))$

- $P(f(a)) \rightarrow \forall x \, P(f(x))$

5 You've Got A Friend

Let $\mathcal{L}$ be a first-order language that contains the predicates, $A(x)$, $C(x)$, $D(x)$, to say x is an animal, cat, and dog, respectively, and $L(x, y)$ and $F(x, y)$, to say x loves y and y is a friend of x, respectively.

(a) Translate the following knowledge base into the language $\mathcal{L}$.

1. Cats and dogs are animals.
2. Everyone loves either a cat or a dog.
3. Anyone who loves an animal has a friend.
4. Everyone has a friend.

(b) Convert your formulas into normal form, negating the last beforehand.

1. Cats and dogs are animals.
2. Everyone loves either a cat or a dog.
3. Anyone who loves an animal has a friend.
4. Someone does not have a friend.

(c) Prove that everyone has a friend, using resolution and proof-by-refutation. (Strategic Hint: Resolve with the negated query last.)

6 Prove It If You Can

(a) Convert the negation of this sentence to normal form: $\exists y \forall x \; R(x, y) \rightarrow \forall x \exists y \; R(x, y)$. Prove this sentence using proof-by-refutation and generalized resolution, or state why it cannot be proven.

(b) Convert the negation of this sentence to normal form: $\forall x \exists y \; R(x, y) \rightarrow \exists y \forall x \; R(x, y)$. Prove this sentence using proof-by-refutation and generalized resolution, or state why it cannot be proven.

7 B & F Chaining

Let $\mathcal{L}$ be a first-order language with two integer constants -1 and 1, the unary function $s(x)$—denoting the successor of x—and the binary predicate less(x, y)—representing x is less than y. Consider the following Horn database, with two rules labeled A and B and fact C:

A. $\text{less}(x, y) \rightarrow \text{less}(x, s(y))$
B. $\text{less}(s(x), y) \rightarrow \text{less}(x, y)$
C. $\text{less}(s(-1), 1)$

(a) Prove less$(-1, s(1))$ by backward chaining (with substitution) on rules A and B and fact C.

(b) Using forward chaining (and substitution), begin enumerating all facts implied by fact C, given rules A and B.