CSIS0270 Artificial Intelligence, 2002–2003
Assignment 3
Deadline Apr 23, 2003, 5:00pm.

This assignment contains both written parts (Questions 1–5) and programming parts (Questions 6). For the written parts, hand-in your answers to the assignment box (R2). For programming parts, hand-in the programs that you write via the hand-in system of the department.

1. ***Resolution in Propositional logic (15%)*** In p. 217 of the textbook, it claims that a particular assignment of variables produces a model whenever resolution fails to produce a refutation proof. Give a proof to that statement.

2. ***Writing FOL sentences (20%)*** (Adapted from question 8.6.) Express each the follow sentences in FOL. You should use simple predicate symbols and compose the sentence out of it, so as to reveal relationships between components in FOL. Define all your predicate symbols. If they are not in CNF, transform it into CNF as well.

   a. Some students took French in spring 2001.

   b. Every student who takes French passes it.

   c. Only one student took Greek in spring 2001.

   d. The best score in Greek is always higher than the best score in French (i.e., for any semester).

   e. Every person who buys a policy is smart.

   f. No person buys an expensive policy.

   g. There is an agent who sells policies only to people who are not insured.

   h. There is a barber who shaves all men in town who do not shave themselves.

   i. A person born in the UK, each of whose parents is a UK citizen or a UK resident, is a UK citizen by birth.

   j. A person born outside the UK, one of whose parent is a UK citizen by birth, is a UK citizen by descent.

   k. Politicians can fool some of the poeple all of the time, and they can fool all the people some of the time, but they can't fool all the people all the time.

3. ***Arithematics in FOL (15%)*** (Adapted from question 8.12.) The textbook explains that FOL can be used to represent the Peano axioms of arithematics, except mathematical induction. This question reveals what is lost by leaving out mathematical induction.

   a. Given the FOL sentences that defines the domain as every natural number:

$$
\begin{aligned}
&\forall x \quad x = 0 \vee (\exists y \quad x = S(y))\\
&\forall x \quad S(x) \neq 0\\
&\forall x, y \quad x \neq y \Rightarrow S(x) \neq S(y)\\
&\forall x \quad +(0, x) = x\\
&\forall x, y \quad +(S(y), x) = S(+(y, x))
\end{aligned}
$$

Show that $\forall x \ \ +(x, 0) = x$ is not provable, by giving a model under which this is not true, although all the above axioms are true.

b. Why mathematical induction would avoid the difficulty?

4. ***Normalization and Unification (10%)*** (Textbook question 9.5) For each pair of atomic sentences, give the most general unifier if it exists. Show all recursive steps involved when running the *Unify* function.

   a. $P(A, B, B)$, $P(x, y, z)$.

   b. $Q(y, G(A, B))$, $Q(G(x, x), y)$.

   c. $Older(Father(y), y)$, $Older(Father(x), John)$.

   d. $Knows(Father(y), y)$, $Knows(x, x)$.

5. ***Prolog warm-up (15%)*** (Textbook question 9.13) The following Prolog code defines a predicate `P`:

```
p(X, [X|Y]).
p(X, [Y|Z]) :- p(X,Z).
```

   a. Show proof trees and (all) solutions for the queries `P(A, [1,2,3])` and `P(2,[1,A,3])`.

   b. What standard list operation does `P` represent?

6. ***Sorting in Prolog (25%)*** (Textbook question 9.14)

   a. Write Prolog clauses that define the predicate `sorted(L)`, which is true if and only if list `L` is sorted in ascending order. (You may assume `L` is bounded.)

   b. Write a Prolog definition for the predicate `perm(L,M)`, which is true if and only if `L` is a permutation of `M`. (You should allow `L` to be unbounded.)

   c. Define `sort(L,M)` (`M` is a sorted version of `L`) using `perm` and `sorted`.

   d. Run `sort` on longer and longer lists until you lose patient. What is the time complexity of your program?

   e. Implement insertion sort in Prolog.