Assignment 1

CPSC 449

September 16, 2016

Abstract

The purpose of this assignment is to assess your ability to solve problems using the logical programming paradigm.

Question 1

Translate the following from Enlish into Prolog.

1. The cat sat on the mat.
2. Mary had a little lamb.
3. The lamb's fleece was white as snow.
4. If Mary went there, then so did the lamb.
5. If a thing is both a human and totally consistent, then the thing is dead.

Question 2

Translate the following Prolog into English.

1. scaredyCat(scooby_doo).
2. lessThan(1,3).
3. likes(ice_crean,bill).
4. has_red_hair(bill).
5. red(X) :- rose(X).
6. even(zero).
7. even(suc(suc(N))) :- even(N).

Question 3

Define a predicate odd, that is true precisely when a number (encoded using zero and suc) is odd.

Question 4

A binary relation, in Prolog, is a symbol of arity 2. Suppose the symbol lt is used to encode a binary relation. Use Prolog give extensions to lt, so that the relation becomes:

1. reflexive

2. symmetric

3. transitive

Could you make the relation antisymmetric?

Question 5

1. Describe a predicate app($L1, X, L2$) of arity 3, in English, that holds precisely when $L2$ is the list $L1$ with X appended onto the end of it. Define this predicate in Prolog.

2. Describe a predicate reversed($L1, L2$) of arity 2, in English, that holds precisely when $L2$ is $L1$ written backwards. Define this predicat in Prolog.

3. Describe a predicate palindromic(L) of arity 1, in English, that holds precisely when L is a palindromic list. Define this predicate in Prolog.

4. Describe a predicate subset(L, Ls) that holds precisely when Ls is the list of sublists of elements of the list L. Define this predicate in Prolog.

5. Describe a predicate subset(L, Ls) that holds precisely when Ls is the list of permutations of L.

6. Can you use the builtin setof to do the last two?

Question 6

Bonus Describe a predicate gcd(m, n, d) of arity 3 that is true when d is the greatest common divisor of m and n.

1. Write an inefficient Prolog program that can be used to determine if a number is prime.

Bonus Goldbach's conjecture says that every even number is the sum of two primes. Write (an inefficient) Prolog program that can be used, given an even number n to find the prime numbers p_1, p_2 with $n = p_1 + p_2$.

Question 7

1. A labelled binary tree in Prolog can be modelled using an arity 3 predicate, $\mathsf{node}(l, t1, t2)$, with leaves being given by a unary predicate $\mathsf{l}(x)$. For example, a labelled tree representing the expression $1+(3*2)$ is represented by $\mathsf{node}(+, \mathsf{l}(1), \mathsf{node}(*, \mathsf{l}(3), \mathsf{l}(2)))$. Write a Prolog program that can be used to change all occurrences of a label X to a label Y. For example, $\mathsf{change}(+, *, \mathsf{node}(+, \mathsf{l}(1), \mathsf{l}(2)), \mathsf{node}(*, \mathsf{l}(1), \mathsf{l}(2)))$.

2. Let $p(m, n)$ be some binary symbol. Write a predicate $\mathsf{assoc}(M, N)$ that can be used to determine if N is a reassociation of p in M. For example, $\mathsf{assoc}(p(1, p(2, 3), p(p(1, 2), 3))$. Use the builtin setof to collect all possible reassociations of p in some term.

Bonus Write a predicate $\mathsf{deriv}(M, N)$ that is true when N is the derivative of M. You may assume that you have only the 1 variable case. You may assume any function symbols you want. For example, $\mathsf{deriv}(\sin(N), \cos(N) * T)$ should hold when $\mathsf{deriv}(N, T)$.

Not for points An antiderivative F of f is a term whose derivative is f. Can you use the above to define antiderivatives?

Question 8

Explain the solution to Prolog problems number 09 and 19. `http://www.ic.unicamp.br/~meidanis/courses/mc336/2009s2/prolog/problemas/p09.pl`

Question 9

Family relations may encoded in Prolog as a list of facts using only two predicates $\mathsf{mother}(X, Y)$ and $\mathsf{father}(X, Y)$. Use these two basic relations to (as Prolog predicates):

1. Define the brother, sister, uncle, grandparent, and cousin relation.

2. Define a more general relation: relative_of.

3. Allow for the list of facts to contain one more kind of predicate: $\mathsf{married}(X, Y)$. Use these three basic predicates to define an even more general relation: extended_relative.

4. Suppose finally that the list of facts contains one more kind of predicate, $\mathsf{wealthy}(X)$. Use this to define a program that can be used to find wealthy members of your extended family.