

# Assignment 4: Declarative Programming

Due: 19:00, Sat 14 May 2016

Full marks: 100

## 1 Introduction

In this assignment, you will gain experience in Prolog and ML, two representative programming languages for logic programming and functional programming respectively.

## 2 Logic Programming

Answer Q1 and Q2 in a Prolog program with the file name assg4.pl. Your program will be graded in “SICStus Prolog 3.12.7” in the CSE Unix platform (command “sics”). The queries should be written in comments. For each code segment and query, you should clearly indicate with comments its corresponding question number.

- Recall the successor notation for representing natural numbers, and the  $\text{sum}(X, Y, Z)$  relation defined in the lecture, which is true if  $Z$  is the sum of  $X$  and  $Y$ .
  - Based on  $\text{sum}/3$ , define  $\text{product}(X, Y, Z)$  which is true if  $Z$  is the product of  $X$  and  $Y$ .
  - Give the query to compute the product of 3 and 4.
  - Give the query to compute the result of 8 divided by 4.
  - Give the query to find the factors of 6.
  - Based on  $\text{product}/3$ , define  $\text{exp}(X, Y, Z)$  which is true if  $Z$  is the result of  $X$  raised to the power  $Y$  (i.e.,  $X^Y = Z$ ).
  - Give the query to compute  $2^3$ .
  - Give the query to compute  $\log_2 8$ .
- A *finite automaton* (or *finite state machine/automaton*) is a collection of finitely many *states* and *transitions* between states due to finitely many different kinds of input. States are usually represented by circles and transitions by arrows between state circles. For the purpose of this question, states are labeled with the English alphabet and inputs are labeled by natural numbers. Figure 1 gives an example automaton. In the example, say if we start in state “c” and give the input string “01111”, then we end up in state “a”.

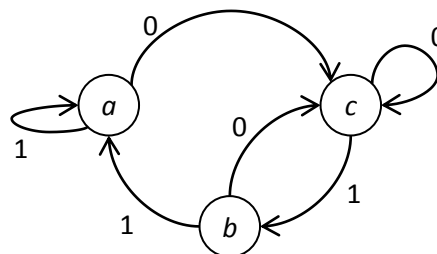


Figure 1: An Example Finite Automaton

If we consider only states that are involved in one or more transitions (that is, there would not be isolated states), then we can represent an automaton by some Prolog facts of the form  $\text{transition}(F, X, T)$ , where  $F$  and  $T$  are the “from” and “to” states of the transition respectively and  $X$  is the symbol triggering the transition.

- Encode the automaton in Figure 1 with some  $\text{transition}(F, X, T)$  facts.

- (b) Define a Prolog relation `state(N)`, which is true if `N` is a state in the automaton defined in the `transition/3` facts.
- (c) Define a Prolog relation `walk(S,B,E)`, which is true if we can go from state `B` to state `E` with input sequence `S` in the form of a list. (For example, using the automaton in Figure 1, `walk([0,1,1,1,1],c,a)` should be true.)

Your program should at least work correctly on the example automaton in Figure 1. Besides, we shall test your program using other automata.

### 3 Functional Programming

Answer Q3 and Q4 in an ML program with the file name `assg4.ml`. Your program will be graded in “Standard ML of New Jersey 110.0.7” in the CSE Unix platform (command “`sm1`”). For each code segment, you should clearly *indicate with comments its corresponding question number*.

3. Recall the (slightly re-worded) type definition of a binary tree in the lecture:

```
datatype 'a bTree = nil | bt of 'a bTree * 'a * 'a bTree;
```

where “`nil`” stands for the empty tree and “`bt`” is the tag of a non-empty binary tree. The following is an example of an `int bTree`:

```
bt(bt(nil,74,nil),
  105,
  bt(bt(nil,109,nil),
    109,
    bt(nil,121,nil)
  )
)
```

- (a) An *inorder traversal* of a binary tree traverses the left subtree, then the root and lastly the right subtree recursively. For example, an inorder traversal of the example tree above should visit the nodes in the order: 74, 105, 109, 109, and 121. Write an ML function `inorder` that accepts a binary tree and returns a list of all the elements of the tree in the order of inorder traversal.
- (b) Write a similar ML function `preorder` that produces a list of elements visited in a *preorder traversal*, which visits the root before traversing the left and then right subtrees.
- (c) Write a similar ML function `postorder` that produces a list of elements visited in a *postorder traversal*, which visits the root only *after* traversing the left and then right subtrees.

Your functions should all use pattern matching.

4. This question is about list processing.
- (a) Write an ML function `symmetric` that takes an integer  $i$ , an integer  $n$ , and a list of length  $n$  where  $1 \leq i \leq n$ , and check whether the  $i^{\text{th}}$  element of the list is identical to the  $(n + 1 - i)^{\text{th}}$  element. That is, given a list  $[a_1, a_2, \dots, a_n]$ , check whether  $a_i = a_{n+1-i}$ .
  - (b) Based on the `symmetric` function in part (a), define an ML function `palindrome` that takes an integer  $n$ , and a list of length  $n$ , and check whether the list is a palindrome. That is given a

list  $[a_1, a_2, \dots, a_n]$ , check whether  $a_1 = a_n$ ,  $a_2 = a_{n-1}$ , ..., and  $a_i = a_{n+1-i}$  for all  $1 \leq i \leq n$ . For example, the list [9,8,6,8,9] is a palindrome.

- (c) Write an ML function `rev` that takes a list of *any type* as input and returns a list containing all elements of the input list but in reverse order. You are not allowed to use any built-in functions.

## 4 Submission and Marking

- Submit the two files `assg4.pl` and `assg4.ml` in Blackboard (<https://elearn.cuhk.edu.hk/>).
- You can submit your assignment multiple times. Only the latest submission counts.
- Plagiarism is strictly monitored and heavily punished if proven. Lending your work to others is subjected to the same penalty as the copier. You have to insert the following statement as comments in your Prolog and ML programs.

CSCI3180 Principles of Programming Languages

--- Declaration ---

I declare that the assignment here submitted is original except for source material explicitly acknowledged. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website <http://www.cuhk.edu.hk/policy/academichonesty/>

Assignment 4

Name:

Student ID:

Email Addr: