CSCI3180 Principles of Programming Languages Spring 2014

Teacher: Prof. Jimmy Lee

Assignment 4: Declarative Programming Due: April 20, 2014

1. A *binary tree* is either empty or composed of a root and two successors, which are binary trees themselves. A binary tree consists of a set of *nodes* and lines connecting parent with children. The nodes are depicted by little circles with the element written inside.

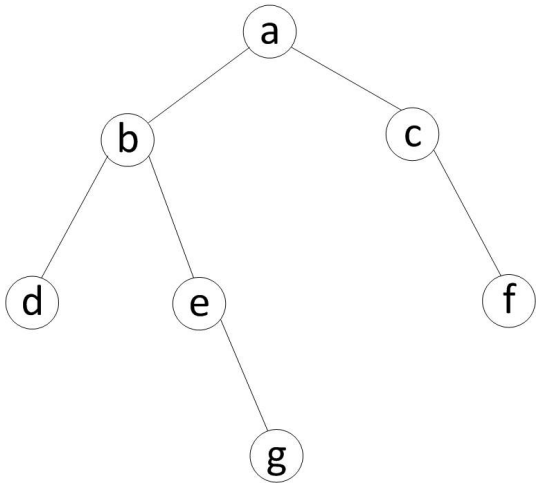

Figure 1: An example binary tree.

In Prolog, we represent the empty tree by the atom "`nil`" and a non-empty tree by the term `bt(X,L,R)`, where `X` denotes the root node and `L` and `R` denotes the left and right subtrees respectively.

(a) Write down the example tree as a Prolog term.

(b) The *height* of a binary tree is the length of the path from the root to the deepest leaf node (which must be an empty tree) in the tree. We define an empty tree to have a height of zero. Thus a tree with a single element node (with both children as empty trees) will have a height of one, and so on.

Give the rule(s) to define the predicate `height(tree,H)`, where "`tree`" is a Prolog term representing a binary tree and `H` is the height of the tree. You might want to define a max/3 predicate first.

(c) Give the rule(s) to define `size(tree,S)` to count the number of nodes in a binary tree. Note that the empty binary tree has 0 nodes.

2. Recall the successor notation for representing natural numbers, and the `sum(X,Y,Z)` relation defined in the lecture, which is true if `Z` is the sum of `X` and `Y`.

   (a) Based on `sum/3`, define `product(X,Y,Z)` which is true if `Z` is the product of `X` and `Y`.

   (b) Give the query to compute the product of 2 and 3.

   (c) Give the query to compute the results of 8 divided by 2.

   (d) Give the query to find the factors of 6.

   (e) Based on `product(X,Y,Z)`, define `exp(X,Y,Z)` which is true if `Z` is the result of raising `X` to the power of `Y`.

   (f) Give the query to compute 2^3.

   (g) Give the query to compute $\log_2 8$.

3. In mathematics, Fibonacci numbers are numbers following the integer sequence:

   1, 1, 2, 3, 5, 8, 13, ...

   To express it in a rigorous mathematical form, this sequence $\{a_n\}_{n=1}^{\infty}$is defined by the following recurrence relation:

   $a_1 = 1$
   $a_2 = 1$
   $a_n = a_{n-1} + a_{n-2}\ \forall n \geq 3$

   Write a ML function `fibonacci` that takes an integer `n` as input and returns the `n-th` number in this Fibonacci number sequence. For example, `fibonacci(6)` should return the integer `8`. Your function should make use of pattern matching.

4. A *binary tree* is a data structure in which each node has a value and at most two children which are referred to as the left child and right child. Recall that the type definition of a binary tree:

   ```
   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.

   (a) A *leaf node* of a binary tree is the node without any child nodes, which means the leaf node has two children which are both `nil`. For example, an empty tree, namely a `nil` tree, has 0 leaf nodes. A single element tree is a tree with only one element and it has two children, which are both `nil` trees. This single element tree has 1 leaf node, which is just the root node. We want you to write a function that could count the number of leaf nodes for a given binary tree.

   Write a ML function `noOfLeaves` which can take a binary tree of *any type* as input and return the number of leaf nodes. Your function `noOfLeaves` should make use of pattern matching.

(b) The *height* of a binary tree is the length of the path from the root to the deepest leaf node. For example, an empty tree has a height of zero. A single element tree has a height of one, and so on so forth.

Write a ML function `height` which can take a binary tree of *any type* as input and return the height of the tree. Your function `height` should make use of pattern matching.

(c) The *size* of a binary tree is the number of the elements in the tree. For example, the size of an empty tree is 0 and the size of a single element tree is 1.

Write a ML function `size` which can take a binary tree of *any type* as input and return its size. Your function `size` should make use of pattern matching.

Submission Guidelines

Please read the guidelines CAREFULLY. If you fail to meet the deadline because of submission problem on your side, marks will still be deducted. **Note that NO LATE SUBMISSION would be allowed in Assignment 4.** So please start your work early!

1. In the following, **SUPPOSE**

   your name is *Chan Tai Man*,
   your student ID is *1155234567*,
   your username is *tmchan*, and
   your email address is *tmchan@cse.cuhk.edu.hk*.

2. In your source files, insert the following header.

```
/*
 * 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 : Chan Tai Man
 * Student ID : 1155234567
 * Email Addr : tmchan@cse.cuhk.edu.hk
 */
```

The sample file header is available at

`http://www.cse.cuhk.edu.hk/~csci3180/resource/header.txt`

3. You should put all Prolog code in file "asg4.pl" and all ML code in "asg4.ml". **Please write the queries in comments. For each code segment and query, you should clearly indicate its corresponding question number.**

4. Make sure you compile and run the programs without any problem.

5. `Tar` your source files to `username.tar` by

```
tar cvf tmchan.tar asg4.pl asg4.ml
```

6. `Gzip` the `tarred` file to `username.tar.gz` by

```
gzip tmchan.tar
```

7. `Uuencode` the `gzipped` file and send it to the course account with the email title "HW4 *studentID yourName*" by

```
uuencode tmchan.tar.gz tmchan.tar.gz \
| mailx -s "HW4 1155234567 Chan Tai Man" csci3180@cse.cuhk.edu.hk
```

8. Please submit your assignment using your Unix accounts.

9. An acknowledgement email will be sent to you if your assignment is received. **DO NOT** delete or modify the acknowledgement email. You should contact your TAs for help if you do not receive the acknowledgement email within 5 minutes after your submission. **DO NOT** re-submit just because you do not receive the acknowledgement email.

10. You can check your submission status at

`http://www.cse.cuhk.edu.hk/~csci3180/submit/hw4.html`.

11. You can re-submit your assignment, but we will only grade the latest submission.

12. Enjoy your work :>