

## Homework 2

**Due:** The deadline (for both the paper and the electronic submission) was extended until Sunday, June 10, 5pm. Because of the extension, no homeworks will be taken later than this. However, students who prefer to hand in according to the initial deadline, Friday, June 8, 5pm, will get a bonus of 5% of their mark.

**Lateness reminder:** penalty 30% if handed by Sunday, 5 pm; not accepted later.

**Note:** The names of your functions should be the ones required in this assignment. The name of the submitted file should be **h2.scm**. Otherwise you will lose marks. In this assignment you should write pure functional code, i.e., you should not use `loop`, `do`, `set!` and other functions ended in `!`. You can use the functions mentioned in class, except `let` and `let*`.

### 1 Question 1

Define functions for arithmetic operations with complex numbers. Assume we represent the complex numbers as a list (real-part, imaginary-part). For example  $2 - 3i$  is represented as `(2 -3)`. Let's denote two complex numbers as  $a+bi$  and  $c+di$ . **Don't use** the built-in complex numbers that are part of Scheme.

**a.** Define a function named **magn** which computes the magnitude of a complex number. Remember that  $|a + bi| = \sqrt{a^2 + b^2}$ . Example:

```
1 ]=> (magn '(-1 2))
;Value: 2.236
```

**b.** Define a function named **add** which adds to complex numbers. Remember that  $(a+bi) + (c+di) = (a+c) + (b+d)i$ . Example:

```
1 ]=> (add '(-1 2) '(2 3))
;Value 1: (1 5)
```

**c.** Define a function named **subtract** which has two complex numbers as arguments, and subtracts the second argument from the first argument. Remember that  $(a+bi) - (c+di) = (a-c) + (b-d)i$ . Example:

```
1 ]=> (subtract '(-1 2) '(2 3))
;Value 1: (-3 -1)
```

**d.** Define a function named **multiply** which multiplies two complex numbers. Remember that  $(a+bi) * (c+di) = (ac-bd) + (ad+bc)i$ . Example:

```
1 ]=> (multiply '(-1 2) '(2 3))
;Value 1: (-8 1)
```

### 2 Question 2

**a.** Define a function called **repl** which replaces all the numbers on the superficial level of a list with their squares. Non-numeric elements are kept unchanged. Examples:

```
1 ]=> (repl '(1 a 2 b 4))
```

```
;Value 1: (1 a 4 b 16)
```

```
1 ]=> (repl '(1 a (2 (3)) b 4))  
;Value 1: (1 a (2 (3)) b 16)
```

**b.** Define a function called **nested-repl** which replaces all the numbers on all the levels of a list with their squares. Non-numeric elements are kept unchanged. Examples:

```
1 ]=> (nested-repl '(1 a 2 b 4))  
;Value 1: (1 a 4 b 16)
```

```
1 ]=> (nested-repl '(1 a (2 (3) a) b 4))  
;Value 1: (1 a (4 (9) a) b 16)
```

### 3 Question 3

Implement functions for binary search trees. We represent a tree as a list (key left right), and an empty tree or subtree as empty list (). You can assume that the keys in the nodes are integers. A search tree contains in each node a key that is bigger than all keys in its left subtree and lower than all keys in the right subtree. Examples:

```
empty tree: ()
```

```
      3      (3 (1 () ()) (6 (4 () ()) ()))  
     / \  
    1   6  
     /  
    4
```

**a.** Implement a function called **insert**, which takes an integer and a binary search tree, and returns a new binary search tree with the integer inserted in the right place. Assume that when the inserted key is already in the tree, the tree will be unchanged. Example:

```
1 ]=> (insert 2 '(3 (1 () ()) (6 (4 () ()) ())))  
;Value 1: (3 (1 () (2 () ())) (6 (4 () ()) ()))
```

```
      3      after inserting 2 becomes:      3  
     / \      / \      / \      / \  
    1   6    1   6    1   6    1   6  
     /      \ / \      \ / \      \ / \  
    4        2 4      2 4      2 4
```

**b.** Implement a function called **list2tree** which transforms a list of integers in a binary search tree. The first element of the list will be the root of the tree (the keys are inserted in the order they come in the list). Example:

```
1 ]=> (list2tree '(5 3 8 6))  
;Value 1: (5 (3 () ()) (8 (6 () ()) ()))
```

```
      5  
     / \
```

```
3 8
 /
6
```

## 4 What to hand in?

Put all definitions of the functions above, as well as any helper function, in a file called **h2.scm**. Add comments to explain what each function is doing, what is the expected input (preconditions), and the expected output (postconditions). The comments should also indicate the question number the function in an answer to.

### 4.1 On paper

The paper submission has to be put in 324 drop box (SF, 2nd floor, near bridge to LP).

Please use an unsealed envelope, having on top the cover page provided at the end of this file. Put inside:

- A printout of your code (the file **h2.scm**).
- A printout of one or more script files containing tests run for each function. (See `man script` if you are unsure of how to create a script file.) The tests for each function you implemented should include several relevant runs, degenerated cases for the input (such as empty list).

Example of generating a script file named file1:

```
> script file1
> scheme
1 ]=> (+ 1 2)
1 ]=> ^D
^D
```

### 4.2 Electronically

In addition to your paper submission, you must submit your code electronically.

- The file **h2.scm** containing all your functions.

Submit the file **h2.scm** using the following command:

```
submit -N homework2 csc324h h2.scm
```

See `man submit` if you need more information. Note that if you already submitted a file and you wish to replace it with a different version, you can submit it again. But you must use the `-f` option to overwrite the old version.

## Cover Page for Homework 2

Family name: \_\_\_\_\_ Student #: \_\_\_\_\_

First name: \_\_\_\_\_ CDF id: \_\_\_\_\_

TA name:        HaiTao Zhang        Ramona Truta        Cosmin Truta        Tristan Miller

I declare that this assignment is my own work, and it is in accordance with the University of Toronto Code of Behaviour on Academic Matters and the Code of Student Conduct.

I discussed this assignment with the following people:

Name: \_\_\_\_\_ Name: \_\_\_\_\_

Name: \_\_\_\_\_ Name: \_\_\_\_\_

Signature \_\_\_\_\_