CS 136, Fall 2009 - Midterm Review Session

October 28, 2009

1. [Analysis]

Analyse the running time of the following code:

```
(define (find-max lst)
  (cond [(empty? lst) false]
        [(empty? (rest lst)) (first lst)]
        [(empty? (filter (lambda(x) (> x (first lst))) (rest lst)))
(first lst)]
        [else (find-max (rest lst))]))
```

The running time for filter is $O(n \times T)$, where n is the length of the list, and T is the running time of the function applied to each element. In this case, $T = O(1)$.

How can you make it run in $O(n)$ time?

2. [Mutation and Memory Model]

We define these state variables:

```
(define x 10)
(define y 's)
(define lst1 (list x y))
(define lst2 (cons (box lst1) lst1))
```

What will be the values of these variables after the following expressions are evaluated?

```
(set! y x)
(set! x 20)
(set-box! (first lst2) 'symbol)
(set! lst1 empty)
```

Draw a diagram and the memory layout to help your understanding.

3. [Data Abstraction]

When you declare a structure, constructor and selectors are automatically created for it. For example, executing

```
(define-struct pc (name arch os))
```

creates functions make-pc, pc-name, pc-arch, and pc-os. In fact, the data structure pc is defined exactly by these functions.

Implement the structure pc by implementing these functions from scratch. How can you make the structure mutable?

4. [Analysis 2 (from assignment 5)]

One implementation of the function reverse is as follows:

```
;; Contract: my-reverse: (listof any) -> (listof any)
;; Purpose: Produces the result of reversing lst
;;
(define (my-reverse lst)
         (cond
           [(empty? lst) lst]
           [else (append (my-reverse (rest lst)) (cons (first lst) empty))]))
```

Let N be the length of lst. Analyze the running time of my-reverse in terms of N.

There is a way to improve the running time of my-reverse. Write a function my-reverse2 which consumes a list and produces the same result as the function my-reverse but is more e cient. Analyze the running time of the function in terms of N, the length of the list.