

Homework Assignment #3

Note

This assignment is due 2:10PM Wednesday, October 17, 2012. Please write or type your answers on A4 (or similar size) paper. Drop your homework by the due time in Yih-Kuen Tsay's mail box on the first floor of Management College Building 2. Late submission will be penalized by 20% for each working day overdue. You may discuss the problems with others, but copying answers is strictly forbidden.

Problems

There are five problems in this assignment, each accounting for 20 points unless otherwise marked.

1. Specify the following two quilts (i.e., write two "symbolic" expressions that denote respectively these two quilts) in Little Quilt.



2. Each evaluation of a function body is called an invocation of the function. Consider a function g defined as follows.

```
let rec g n =
  if n = 1 then 1
  else g (if even n then n/2 else 3 * n + 1)
```

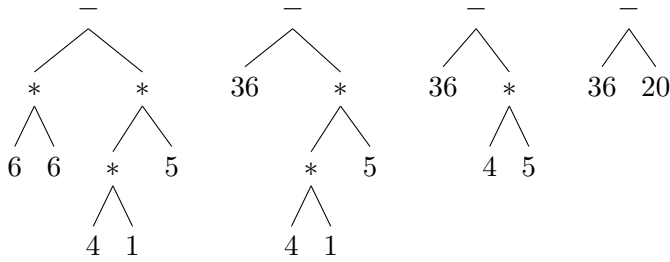
where *even* n returns true if n is even. How many invocations of g are there in the evaluation of g 12? What is the sequence of numbers formed by the values of the actuals in these invocations? (Note: the sequence is precisely the Collatz sequence from 12. The Collatz conjecture states that, no matter what positive integer such a sequence starts with, the sequence will always eventually reach 1.)

3. Sketch the innermost evaluation of f 98, where f is the 91-function given below.

```
let rec f n = if n > 100 then n - 10 else f (f (n + 11))
```

You may assume that f 100 evaluates to 91 (which we obtained in class).

4. (10 points) Innermost evaluation corresponds to tree rewriting. Below is a sequence of trees that correspond to the expressions encountered during an innermost evaluation of $6 * 6 - 4 * 1 * 5$. For instance, the subtree for $6 * 6$ is rewritten as a node for 36, representing that the immediate result of evaluating $6 * 6$ is 36.



Draw a sequence of trees that correspond to the innermost evaluation of each of the following expressions.

(a) $1=2 \ || \ 1=1$

(b) $1=1 \ || \ 1=2$

5. Function *fib* computes Fibonacci numbers:

```

let rec fib n =
  if n = 0 || n = 1 then 1
  else fib (n - 2) + fib (n - 1)

```

(a) (10 points) Draw an abstract syntax tree for the function body.

(b) (20 points) Draw a sequence of trees that correspond to the innermost evaluation of *fib* 2. A subtree for *fib* *m* should be rewritten as another subtree for the function body of *fib* with *m* substituted for each occurrence of the formal.