HW 2 – Unification, Type Derivation, Regular Expressions and Finite State Automata

CS 421 – Spring 2007
Revision 1.0

Assigned Wednsday, February 28, 2007
Due Wednesday, March 7, 2007, 9:00 AM - in class
Extension 48 hours (20% penalty)

1 Change Log

1.0 Initial Release.

2 Turn-In Procedure

Your answers to the following questions are to be hand-written, or printed, neatly on one or more sheets of paper, each with your name in the upper right corner. The homework is to be turned in in class at the start of class. Alternately, you may hand it to Prof. Elsa Gunter in person before the deadline.

3 Objectives and Background

The purpose of this HW is to test your understanding of

- How to unify a system of equations
- How to perform type derivations in simplified OCaml
- How to use regular expressions and finite state automata to formally express sets of strings (called *languages*) given by an English language description

Another purpose of HW2 is to provide you with experience answering non-programming written questions of the kind you may experience on the second midterm and final.
Caution: It is strongly advised that you know how to do these problems before the second midterm.

4 Problems

1. (15 points) Give a most general unifier for the following set of equations (unification problem). Capital letters (A, B, C, D) denote variables of unification. The lower-case letters (f, l, n, p) are constants or term constructors. (f and p have arity 2 - i.e., take 2 arguments, l has arity 1, and n has arity 0 - i.e. it is a constant.) Show your work by listing the operations performed in each step of the unification and the result of that step.

$$\{(f(A, B) = f(l(C), f(l(D), l(p(C, D))))); \ (l(C) = l(n)); \ (D = p(n, A))\}$$

2. (25 points) Give a complete type derivation for the following typing judgment's:

$\{\} \vdash$ `(let rec square = fun n ->`
`if n = 0 then 0 else let sq = square (n - 1) in sq + ((2 * n) - 1)`
`in square 5) : int`

3. (20 points) For each of the following languages (ie, sets of strings), write a regular expression generating the set, and draw a deterministic finite state automaton accepting the set:

   1. (10 points) The set of all strings of `a`'s and `b`'s of even length having exactly one `b`.
   2. (10 points) The set of all strings of `0`'s, `1`'s, and `2`'s, such that before and after each `2`, the combined number of `0`'s and `1`'s is always even.

4. (Extra Credit) (6 points) Write an ocaml program that implements the DFSA in Problem 3, part 1. It's type should be `: string list -> bool`. The strings in the list can be assumed to be either `"a"` or `"b"`.