Programming Languages (COP 4020/6021) [Spring 2017]

Assignment IV

Objectives

1. To gain experience programming with recursively defined data types in ML.
2. To demonstrate an understanding of static semantics by implementing a type checker.
3. To demonstrate an understanding of dynamic semantics by implementing an interpreter.
4. To gain experience setting up deductive systems.
5. To formalize static and dynamic semantics for a new programming language.

Due Date: Monday, March 20, 2017 (at 5pm).
Late submission: You may submit any part (or both parts) of this assignment late (i.e., between 5pm on 3/20 and 5pm on 3/22) with a 15% penalty on the whole assignment.

Machine Details: *Complete this assignment by yourself*, the programming portion on the following CSEE network computers: c4lab01, c4lab02, ..., c4lab20. These machines are physically located in ENB 220. Do not use any server machines like grad, babbage, sunblast, etc. You can connect to the C4 machines from home using SSH. (Example: Host name: *c4lab01.csee.usf.edu* Login ID and Password: <your NetID username and password>) You are responsible for ensuring that your programs compile and execute properly on these machines.

Assignment Description

(1) **Programming portion:** First, correct any problems with your implementation of the `sub` function from Assignment III. Then, in a directory containing a copy of *as3.sml*, begin a new file called *as4.sml* with the command use `"as3.sml";`. Then implement the following values in *as4.sml*.

(a) `tc : expr -> typ option`
This function takes a diMLazy expression e and returns NONE iff e is an ill-typed program and SOME t iff e is a well-typed program having type t.

(b) `exception Stuck`
Section 5.2 of the Elements of ML Programming textbook has details on using exceptions in ML.

(c) `eval : expr -> expr`
This function takes a diMLazy expression e and evaluates e for as many steps as possible. If evaluation of e converges to a value v, then eval(e) returns v; if e diverges then so does eval(e). Function eval must raise exception Stuck at any point that evaluation gets "stuck" without a value being produced (but note that because diMLazy is type safe, only ill-typed expressions can get stuck before becoming values). Evaluation in diMLazy is right-to-left and call-by-name. Please note that the CBN evaluation strategy takes precedence over the right-to-left evaluation order, so the fact that diMLazy has right-to-left evaluation order really just means, on this assignment, that PlusExprs and LessExprs are evaluated right to left (i.e., they're right associative).

Throughout this assignment, you may assume that all variable names in expressions being type checked and evaluated are unique, so you never have to alpha-convert expressions.

As on Assignment III, you'll also want to make test cases (i.e., example diMLazy expressions) to try your code on. Creating good test cases is an important part of software engineering and something employers would expect you to be able to do.

Grading and Submission Notes
For full credit, your implementation must:

- be commented and formatted appropriately (as on previous assignments).
- use ML features like pattern matching when appropriate.
- not define any extra top-level values.
- compile and run on the C4 machines with no errors or warnings (except for Stuck exceptions raised at appropriate times).
- not use any library functions.
- not use any ML features that cause side effects to occur (e.g., I/O or references/pointers).
- not be significantly more complicated than is necessary.
- be reasonably efficient.

The submission process is the same as for Assignment III, except here you'll submit *as4.sml* in Canvas. Please remember to include the pledge as an initial comment; not doing so will lower your grade 50%.

(2) **Theory portion:**
Define static and dynamic semantics for the language L from Assignment III. Assume that all variable names in all expressions under consideration have been made unique through alpha-conversion; hence, you never have to consider contexts containing more than one entry for the same variable. Also, assume that capture-avoiding substitution ([e/x]e') is already defined for L, so you can just use that notation ([e/x]e') without defining it.

As always, avoid making the definitions significantly more complicated than necessary. If you get stuck at any point, please explain in prose whatever you're having trouble formalizing.

Theory-portion Submission Reminders

- Write the following pledge at the end of your submission: "I pledge my Honor that I have not cheated, and will not cheat, on this assignment." Sign your name after the pledge. Not including this pledge will lower your grade 50%.
- For full credit, turn in a hardcopy (handwritten or printed) version of your solutions.
- Late submissions may be emailed or submitted in hardcopy.
- All emailed submissions, even if sent before the deadline, will be graded as if they were submitted late, i.e., with a 15% penalty.
- If you think there's a chance you'll be absent or late for class on the date this assignment is due, you're welcome to submit solutions early by giving them to me or the TA before or after class, or during any of our office hours.