Homework 5: SASL

This homework is due March 22, 1991, (by 5:00 PM).

Test your programs using the `sasl` command on Unix, which is the interpreter described in Appendix F. Put your SASL code for the standard problems in `hw5.l`. Hand in a print-out of the code along with your answers to other problems, and please put your name on both. (If you do extra-credit problems, please write the names of the file(s) where the source code for your solutions can be found on your print-out.)

1. (15 points) The lambda calculus (see section 5.8) is a much simpler language than our version of SASL. What, if any, are the reasons for adding all the features of SASL to the lambda calculus, given that with the lambda calculus one is able to define all the functions that one can define with SASL? (Make your arguments as precise as you can.)

2. (extra credit only) In a language like SASL or Scheme with static scoping, is it possible to simulate some of the effects of dynamic scoping? For example, is it possible to modify the print-num function (and how it is used in exercise 9 of page 151 of Kamin's book) to accomplish the same effect has having dynamically scoped variables? Can you do it without changing the interface to print-num (that is, without changing how many arguments print-num takes)?

3. (6 points) Do problem 1, parts (a) and (b) on page 199 of Kamin's book. For this problem, define a global variable containing the requested list.

4. (30 points) Do problem 2, parts (a) through (f) on pages 199–201 of Kamin's book.

5. (extra credit only) Do problem 2, part (g) on page 201.

6. (15 points) Do problem 3, part (a) on page 201.

7. (extra credit only) Do problem 3, parts (b) and/or (c) on page 201.

8. (10 points) Do problem 4 on page 201, parts (a) and (b).

9. (extra credit only) Do problem 4 on page 201, parts (c) and/or (d).

10. (extra credit only) Do (one or more of) the problems 5 or 6 in section 5.11.2 of Kamin's book. You will find the source for the sasl interpreter in

    `/home/cs342/public/src/chap.5/sasl.p`

    Make a copy using the Unix `cp` command and then edit your copy. Turn in differences (using the Unix command `diff`) as before.

11. (extra credit only) Modify the sasl interpreter so that closures and thunks are displayed in a form similar to that used in the text. Since environments can be circular structures of pointers, don't print the global environment (note that Kamin usually writes {} for the global environment), or only print at most one level of environment. Exact details of the output format are up to you, but try to reuse as much of the existing interpreter's code as you can (e.g., for printing lists).