Homework 7 — ML — assigned Wednesday 31 March — due Wednesday 7 April

7.1 Formatting (100pts)

We can use the following data type declaration to introduce a language of simple arithmetic expressions, with variable names (as in Homework Exercise 4.2):

```ml
datatype expr = Num of int
             | Var of string
             | Let of {var: string, value: expr, body: expr}
             | Add of expr * expr
             | Sub of expr * expr
             | Mul of expr * expr
             | Div of expr * expr

type env = string -> int
exception Unbound of string
val emptyEnv: env = fn s => raise (Unbound s)
fun extendEnv oldEnv s n s' = if s' = s then n else oldEnv s'
exception ExprDivByZero
```

Write a formatter function (packaged appropriately using the ML module language) `format`, with type `expr -> string`, that produces a visual rendering of the expression; given an expression `e: expr`, the result of `format e: string` is an ML string containing the PostScript description of a page on which is shown the formatted expression `e`.

Here is a specification of the formatter’s actions for each possible kind of `expr`:

- for `Num`, display the number in decimal notation
- for `Var`, display the variable in 16pt Times-Italic font
- for `Let`, display the keyword `let` in 16pt Times-Bold font, the variable in 16pt Times-Italic font, the equals sign (=), and the value expression, with a required space between the `let` and the variable; then display the keyword `in` on a new line; then display the body expression, beginning on a new line and indented by 24pt to the right of the x-coordinate of the beginning of the keyword `let`; then display the keyword `end` on a new line
- for `Add`, `Sub`, `Mul`, and `Div`, display an open parenthesis, the left subexpression, the mathematical symbol for the operator (`+`, `−`, `×`, `÷`), the right subexpression, and a closed parenthesis

All text should be in 16pt Times-Roman font except as specified above. Line spacing should be 20pt. You may assume that no `Let` appears within the value expression of another `Let`. The formatter only needs to be able to accept those expressions which, when correctly formatted according to this specification, do not overflow lines or pages. A small amount of white space may be added between lexical units according to taste. The bounding box specification should be `0 0 612 792`, i.e., standard U.S. letter-size paper.

**Execution of the resulting PostScript must not leave any objects behind on the PostScript operand stack.**

The ML function `format` must not contain any embedded data about the sizes of individual glyphs (font characters).

**Example 1.** For the following expression:

Let `{var="x", value=Num 999, body=Mul (Num 12345,
Let \( \text{var} = \text{variable}, \text{value} = \text{Mul} (\text{Num} 12345, \text{Sub} (\text{Var} \ "x", \text{Num} 6789)), \)
\[ \text{body} = \text{Div} (\text{var}, \text{Var} \ "x") \]

the formatted code should look like:

\[
\begin{align*}
\textbf{let } x & = 999 \\
\textbf{in} & \quad (12345 \times \textbf{let } \text{variable} = (12345 \times (x - 6789)) \\
& \quad \textbf{in} \quad (\text{variable} \div x) \\
& \quad \textbf{end} \\
\textbf{end}
\end{align*}
\]

Example 2. For the following expression:

\[
\text{Mul} (\text{Let} (\text{var} = \text{x}, \text{value} = \text{Num} 5, \text{body} = \text{Mul} (\text{Var} \ "x", \text{Var} \ "x")), \\
\text{Let} (\text{var} = \text{z}, \text{value} = \text{Num} 5, \text{body} = \text{Mul} (\text{Var} \ "z", \text{Var} \ "z")))
\]

the formatted code should look like:

\[
\begin{align*}
\textbf{(let } x & = 5 \\
\textbf{in} & \quad (x \times x) \\
\textbf{end} \times \textbf{let } z & = 5 \\
& \quad \textbf{in} \quad (z \times z) \\
& \quad \textbf{end}
\end{align*}
\]

Hint: You will find the following PostScript operators useful in this exercise: \textbf{currentpoint}, \textbf{selectfont}, \textbf{show}, \textbf{glyphshow}.

\textbf{How to turn in}

Make sure that you have thoroughly tested your code, and include all your test runs!

Turn in your code by running

\texttt{˜clint/handin your-file}

on a regular UNM CS machine. You should use whatever filename is appropriate in place of your-file.

Include the following statement with your submission, signed and dated:

\emph{I pledge my honor that in the preparation of this assignment I have complied with the University of New Mexico Board of Regents' Policy Manual, including Section 4.8, Academic Dishonesty.}