Optional: Homework # 14
due Tuesday, May 10, 3:30 PM

This is an optional homework. You are under no obligation to do it. The grade on this homework will take the place of a lower homework score (if any).

1 Denotational Semantics

Implement the denotational semantics from lecture in ML:

$$\mathcal{E}[\![n]\!]m = n \qquad \mathcal{E}[\![e_1 \texttt{ + } e_2]\!]m = (\mathcal{E}[\![e_1]\!]m) + (\mathcal{E}[\![e_2]\!]m)$$

$$\mathcal{E}[\![e_1 \texttt{ * } e_2]\!]m = (\mathcal{E}[\![e_1]\!]m)\,(\mathcal{E}[\![e_2]\!]m) \qquad \mathcal{E}[\![x]\!]m = mx$$

$$\mathcal{S}[\![\texttt{skip}]\!]m = (m, ()) \qquad \mathcal{S}[\![\texttt{print } e]\!]m = (m, (\mathcal{E}[\![e]\!]m))$$

$$\mathcal{S}[\![x \texttt{ := } e]\!]m = (m\,[x \mapsto \mathcal{E}[\![e]\!]m]\,, ())$$

$$\mathcal{S}[\![s_1\texttt{; } s_2]\!]m_0 = (m_2, (o_1 \ldots, o_2 \ldots)) \text{where } \begin{array}{rcl}(m_1, o_1) & = & \mathcal{S}[\![s_1]\!]m_0 \\ (m_2, o_2) & = & \mathcal{S}[\![s_2]\!]m_1\end{array}$$

$$\mathcal{S}[\![\texttt{if } e_0 \texttt{ > 0 then } s_1 \texttt{ else } s_2]\!]m = \begin{cases} \mathcal{S}[\![s_1]\!]m & \text{if } \mathcal{E}[\![e_0]\!]m > 0 \\ \mathcal{S}[\![s_2]\!]m & \text{otherwise}\end{cases}$$

$$\mathcal{S}[\![\texttt{until } e \texttt{ > 0 do } s]\!] = \mathcal{S}[\![\texttt{if } e \texttt{ > 0 then skip else } s\texttt{; until } e \texttt{ > 0 do } s]\!]$$

You should use the following algebraic data types for expressions and statements:

```
datatype expr = LIT of int
              | PLUS of expr * expr
              | TIMES of expr * expr
              | VAR of string
              ;

datatype stmt = SKIP
              | ASSIGN of string * expr
              | SEQ of stmt * stmt
              | PRINT of expr
              | IFPOS of expr * stmt * stmt
              | UNTILPOS of expr * stmt
              ;

type memory = string -> int;
type output = int list;
```

You should name the denotational functions $\mathcal{E}$ and $\mathcal{S}$ `sExpr` and `sStmt` respectively. they should have type:

```
sExpr : expr -> memory -> int
sStmt : stmt -> memory -> memory * output
```

Put your solution in `homework14.sml`.

2 Axiomatic Semantics

Give the weakest precondition at the beginning of this code that enables us to prove that $y \geq 0$ at the end of the following code:

```
y = x;
if (y < 0) y = 0 - y; else skip
```

Show your work! Simplify the final logical expression.

Put your answer in a PLAIN ASCII TEXT file named homework14.txt.

3 Submitting Your Work

You submit your program work by putting it in the `homework14` directory in your AFS class volume.

The `homework14` directory should include the following:

- `homework14.sml`
- `homework14.txt`