1 Instructions

Begin by downloading the file hw2-base.ml from the course website, and renaming it to hw2.ml. Then fill in your answers to the problems, adding or modifying definitions as you see fit. Make sure to answer the problems in both Part 2 (Grammars and Abstract Syntax) and Part 3 (Type Checking). Submit your completed hw2.ml via Gradescope. As always, please don’t hesitate to ask for help on Piazza (https://piazza.com/class/ksknvqg6ogb2kc).

2 Grammars and Abstract Syntax

The type stmt corresponds to this grammar for part of the C language:

\[ S ::= \langle name \rangle = \langle int \rangle ; \mid \text{if} (\langle name \rangle) \{ S \} \text{else} \{ S \} \]

1. (3 points) What program in the language of the grammar is represented by the value IfStmt ("x", AssignStmt ("y", 5), AssignStmt ("z", 2))? Write your answer in the comments in the space provided.

2. (3 points) Suppose we wanted to extend the grammar with a case

\[ \text{do} \{ S \} \text{while} (\langle name \rangle) \]

Extend the OCaml definition of stmt with a case that represents this syntax.

3. (4 points) Define a value stmt2 : stmt representing the following C program:

\[ \text{do} \{ \text{if} (x) \{ x = 0; \} \text{else} \{ x = 1; \} \} \text{while} (y) \]

3 Type Checking

The function typecheck contains the incomplete typechecker for expressions that we wrote in class.
4. (5 points) Add a case for If, according to the following typing rule:

\[
\frac{e : \text{bool} \quad e_1 : \tau \quad e_2 : \tau}{\text{if } e \text{ then } e_1 \text{ else } e_2 : \tau}
\]

Note that the types of \(e_1\) and \(e_2\) must match the type of the overall if expression. Once you have added this case, the examples in the code should return the indicated values. You may want to add some test cases of your own as well.

5. (for graduate students) Add a case for Eq, according to the following typing rule:

\[
\frac{e_1 : \tau \quad e_2 : \tau}{e_1 = e_2 : \text{bool}}
\]

Hint: The type \(\tau\) is either int or bool.