HW 5 **Prof. Caldwell**
COSC 4780

Assignment:

1. Read pages 1 – 14 of the Schmidt text.
2. Using the style of proof presented in the notes below, prove that the example in figure 1.5 (pp. 10) is well-typed.
3. Prove the following commands are (or are not) well-typed.

$$\begin{array}{l} loc_1 := 0; \textbf{while}\ \neg(@loc_1 = 0)\ \textbf{do}\ loc_1 := @loc_2 + 1\ \textbf{od} : comm \\ @loc_1 := 0 \\ loc_1 := @loc_1 = 0 \end{array}$$

1 The Core Language

We are using the core language from Schmidt.

1.1 Syntax

The syntax for the core language is

$$\begin{array}{ll} C ::= L{:=}E \mid C_1;C_2 \mid \textbf{if}\ E\ \textbf{then}\ C_1\ \textbf{else}\ C_2\ \textbf{fi} \mid \textbf{while}\ E\ \textbf{do}\ C\ \textbf{od} \mid \textbf{skip} & \\ E ::= N \mid @L \mid E_1 + E_2 \mid \neg E \mid E_1 = E_2 & \\ L ::= loc_i & \text{if } i > 0 \\ N ::= n & \text{if } n \in \mathbb{Z} \end{array}$$

1.2 Types

The types are *int*, *intloc*, *intexp*, *boolexp* and *comm.*

1.3 Typing Rules

The typing rules can be sued to build derivations showing a syntactically well-formed phrase is well-typed. There are four styles of rules here.

For the purposes of the following discussion, let $P, P_1, P_2, P_3 \in \{C, E, L, N\}$ be meta-variables denoting syntactic classes and let $T, T_1, T_2, T_3 \in \{int, intloc, intexp, boolexp, comm\}$ be meta-variables denoting types.

Rules of the following form are axiom rules.

$$\frac{}{P{:}T}\ \text{if } C$$

They may (or may not) have a side condition C, which if it holds, means there is nothing more to prove.

Rules of the form

$$\frac{P_1{:}T_1}{P{:}T} \qquad \frac{P_1{:}T_1 \quad P_2{:}T_2}{P{:}T} \qquad \frac{P_1{:}T_1 \quad P_2{:}T_2 \quad P_3{:}T_3}{P : T}$$

These rules have hypotheses which are typing goals that remain to be shown. Rules used here have one, two or three hypotheses.

The typing rules for the core languages are as follows:

For Location: For Numeral

$$\frac{}{loc_i : intloc} \quad \text{if } i > 0 \qquad\qquad \frac{}{N : int} \quad \text{if } n \in \mathbb{Z}$$

For Expression:

$$\frac{N : int}{N : intexp} \qquad \frac{L : intloc}{@L : intexp} \qquad \frac{E_1 : intexp \quad E_2 : intexp}{E_1 + E_2 : intexp}$$

$$\frac{E : boolexp}{\neg E : boolexp} \qquad \frac{E_1 : \tau exp \quad E_2 : \tau exp}{E_1 = E_2 : boolexp} \quad \text{if } \tau \in \{int, bool\}$$

For Command:

$$\frac{L : intloc \quad E : intexp}{L{:=}E : comm} \qquad \frac{C_1 : comm \quad C_2 : comm}{C_1\,; C_2 : comm} \qquad \frac{}{\mathbf{skip} : comm}$$

$$\frac{E : boolexp \quad C_1 : comm \quad C_2 : comm}{\mathbf{if}\ E\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2\ \mathbf{fi} : comm} \qquad \frac{E : boolexp \quad C : comm}{\mathbf{while}\ E\ \mathbf{do}\ C\ \mathbf{od} : comm}$$

1.4 Building type derivations

My style of writing typing derivations differs a bit from Schmidt's. You can see examples of his style in figure 1.3 on page 4. The following examples will show how I do it.

Example 1.1. As a first example, I present a derivation that the phrase $loc_1 := 5 + @loc_2$ is a well-typed command. Thus, the goal is to build a derivation justifying the following typing assertion.

$$loc_1 := 5 + @loc_2 : comm$$

Find a rule that matches by considering the syntax of the phrase and the type and matching against the bottom line of a typing rules. In this case, the assignment rule matches with a substitution of the following form:

$$\{L := loc_1, E := 5 + @loc_1\}$$

The way to see that this works is to apply the substitution to the goal of the typing rule and notice that the result is exactly the goal we are trying to show. Note that, for example, the addititiion rules does not match this goal – there is no substitution that can be applied to the goal of that rule $E_1 + E_2 : intexp$ that will yield the assertion we are trying to show. To build the subgoals we must derive, replace L in the left hypotheses of the rule by loc_1 and replace E in the right hypothesis by the expression $5 + @loc_2$. This yields the following partial derivation.

$$\dfrac{@\mathit{loc}_1 : \mathit{intloc} \qquad 5 + @\mathit{loc}_2 : \mathit{intexp}}{\mathit{loc}_1 := 5 + @\mathit{loc}_2 : \mathit{comm}}$$

Now, on the left branch, since $1 > 0$ we know the loc_1 is well-typed. This closes the left branch.

$$\dfrac{\dfrac{}{\mathit{loc}_1 : \mathit{intloc}} \qquad 5 + @\mathit{loc}_2 : \mathit{intexp}}{\mathit{loc}_1 := 5 + @\mathit{loc}_2 : \mathit{comm}}$$

On the right side, the rule for typing int-expressions defined by addition applies using the following match.

$$\{E_1 := 5, E_2 := @\mathit{loc}_2$$

Apply the rule by writing down a line ober the open subgoal and writing down the result of applying the matching substitution to the two subgoals $E_1 : \mathit{intexp}$ and $E_2 : \mathit{intexp}$ above the line. This yields the following partial derivation.

$$\dfrac{\dfrac{}{\mathit{loc}_1 : \mathit{intloc}} \qquad \dfrac{5 : \mathit{intexp} \qquad @\mathit{loc}_2 : \mathit{intexp}}{5 + @\mathit{loc}_2 : \mathit{intexp}}}{\mathit{loc}_1 := 5 + @\mathit{loc}_2 : \mathit{comm}}$$

On the leftmost open branch, the rule for showing a numeral is an int-expression applies and then the axiom rule for numerals as *int* holds because the side condition holds *i.e.* that 5 is an integer. Applying these two rules to the left open branch yields the following.

$$\dfrac{\dfrac{}{\mathit{loc}_1 : \mathit{intloc}} \qquad \dfrac{\dfrac{\dfrac{}{5 : \mathit{int}}}{5 : \mathit{intexp}} \qquad @\mathit{loc}_2 : \mathit{intexp}}{5 + @\mathit{loc}_2 : \mathit{intexp}}}{\mathit{loc}_1 := 5 + @\mathit{loc}_2 : \mathit{comm}}$$

For the right open branch, we match the rule for typing the dereference operator. This rule says, to show that an assertion of the form $@L : \mathit{intexp}$ holds, show that the assertion $L : \mathit{intloc}$. There is an axiom rule for assertions of this form. In this case, L is loc_2 and $2 > 0$ so so the side condition holds. Applying both rules apply closes the open branch of the derivation yielding a completed type derivation.

$$\dfrac{\dfrac{}{\mathit{loc}_1 : \mathit{intloc}} \qquad \dfrac{\dfrac{\dfrac{}{5 : \mathit{int}}}{5 : \mathit{intexp}} \qquad \dfrac{\dfrac{}{\mathit{loc}_2 : \mathit{intloc}}}{@\mathit{loc}_2 : \mathit{intexp}}}{5 + @\mathit{loc}_2 : \mathit{intexp}}}{\mathit{loc}_1 := 5 + @\mathit{loc}_2 : \mathit{comm}}$$

Example 1.2. Here's a derivation that the command

$$\mathbf{if}\ @\mathit{loc}_1 = 0\ \mathbf{then}\ \mathit{loc}_1 := @\mathit{loc}_2\ \mathbf{else\ skip\ fi}$$

is a well-typed command *i.e.* that it has type *comm*.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathit{loc}_1 : \mathit{intloc}}}{@\mathit{loc}_1 : \mathit{intexp}} \quad \dfrac{\dfrac{}{0 : \mathit{int}}}{0 : \mathit{intexp}}}{@\mathit{loc}_1 = 0 : \mathit{boolexp}} \quad \dfrac{\dfrac{}{\mathit{loc}_1 : \mathit{intloc}} \quad \dfrac{\dfrac{}{\mathit{loc}_2 : \mathit{intloc}}}{@\mathit{loc}_2 : \mathit{intexp}}}{\mathit{loc}_1 := @\mathit{loc}_2 : \mathit{comm}} \quad \dfrac{}{\mathbf{skip} : \mathit{comm}}}{\mathbf{if}\ @\mathit{loc}_1 = 0\ \mathbf{then}\ \mathit{loc}_1 := @\mathit{loc}_2\ \mathbf{else\ skip\ fi}\text{: comm}}$$