Commands

Concrete syntax:

$$\begin{array}{lcl} com & ::= & \texttt{SKIP} \\ & | & string\ \texttt{::=}\ aexp \\ & | & com\ \texttt{;;}\ com \\ & | & \texttt{IF}\ bexp\ \texttt{THEN}\ com\ \texttt{ELSE}\ com \\ & | & \texttt{WHILE}\ bexp\ \texttt{DO}\ com \end{array}$$

Commands

Abstract syntax:

$$\begin{array}{lcl} \textbf{datatype}\ com & = & SKIP \\ & \mid & Assign\ string\ aexp \\ & \mid & Seq\ com\ com \\ & \mid & If\ bexp\ com\ com \\ & \mid & While\ bexp\ com \end{array}$$

Big-step semantics

Concrete syntax:

$$(com,\ initial\text{-}state) \Rightarrow final\text{-}state$$

Intended meaning of $(c,\ s) \Rightarrow t$:

Command c started in state s terminates in state t

"$\Rightarrow$" here not type!

Big-step rules

$$(SKIP,\ s) \Rightarrow s$$

$$(x ::= a,\ s) \Rightarrow s(x := aval\ a\ s)$$

$$\frac{(c_1,\ s_1) \Rightarrow s_2 \qquad (c_2,\ s_2) \Rightarrow s_3}{(c_1;;\ c_2,\ s_1) \Rightarrow s_3}$$

Big-step rules

$$\frac{bval\ b\ s \qquad (c_1,\ s) \Rightarrow t}{(IF\ b\ THEN\ c_1\ ELSE\ c_2,\ s) \Rightarrow t}$$

$$\frac{\neg\ bval\ b\ s \qquad (c_2,\ s) \Rightarrow t}{(IF\ b\ THEN\ c_1\ ELSE\ c_2,\ s) \Rightarrow t}$$

Big-step rules

$$\frac{\neg\ bval\ b\ s}{(WHILE\ b\ DO\ c,\ s) \Rightarrow s}$$

$$\frac{bval\ b\ s_1 \qquad (c,\ s_1) \Rightarrow s_2 \qquad (WHILE\ b\ DO\ c,\ s_2) \Rightarrow s_3}{(WHILE\ b\ DO\ c,\ s_1) \Rightarrow s_3}$$

Well-typed commands

Notation:

$$\Gamma \vdash c$$
$$\mathit{tyenv} \vdash \mathit{com}$$

Read: *In context* Γ*,* c *is well-typed*.

The rules:

$$\Gamma \vdash \mathit{SKIP} \qquad \frac{\Gamma \vdash a : \Gamma\ x}{\Gamma \vdash x ::= a}$$

$$\frac{\Gamma \vdash c_1 \qquad \Gamma \vdash c_2}{\Gamma \vdash c_1;;\ c_2}$$

$$\frac{\Gamma \vdash b : \mathsf{bool} \quad \Gamma \vdash c_1 \qquad \Gamma \vdash c_2}{\Gamma \vdash \mathit{IF}\ b\ \mathit{THEN}\ c_1\ \mathit{ELSE}\ c_2}$$

$$\frac{\Gamma \vdash b : \mathsf{bool} \quad \Gamma \vdash c}{\Gamma \vdash \mathit{WHILE}\ b\ \mathit{DO}\ c}$$

Syntax-directedness

All three sets of typing rules are *syntax-directed*:

- There is exactly one rule for each syntactic construct ($SKIP$, $::=$, $\ldots$).
- Well-typedness of a term $C\ t_1 \ldots t_n$ depends only on the well-typedness of its subterms t_1, $\ldots$, t_n.

A syntax-directed set of rules

- is executable by backchaining without backtracking and
- backchaining terminates and requires at most as many steps as the size of the term.

Syntax-directedness

The big-step semantics is not syntax-directed:

- more than one rule per construct and
- the execution of $WHILE$ depends on the execution of $WHILE$.