CAP Companion Sheet

Mini-while abstract syntax

Expressions:

$e \in \mathcal{E}$	::=	n	*integer constant*
	|	$tt \mid ff$	*boolean constant*
	|	x	*variable*
	|	$e + e$	*addition*
	|	$e \times e$	*multiplication*
	|	e `or` e	*or*
	|	e `and` e	*and*
	|	$e == e \mid e < e \mid \ldots$	*tests*
	|	$\ldots$	

Statements:

$S \in Stm$	::=	$x := e$	assign
	|	*skip*	do nothing
	|	$S_1; S_2$	sequence
	|	`if` b `then` S_1 `else` S_2	test
	|	`while` b `do` S `done`	loop

Typing and static semantic for mini-while

We add declarations for the language:

P	::=	$D; S$	program
D	::=	$var\ x : \tau \mid D; D$	type declaration

From declarations we infer $\Gamma : Var \to Basetype$ with the two following rules:

$$\frac{}{var\ x : \tau \to_d [x \mapsto \tau]}$$

$$\frac{D_1 \to_d \Gamma_1 \qquad D_2 \to_d \Gamma_2 \qquad Dom(\Gamma_1) \cap Dom(\Gamma_2) = \emptyset}{D_1; D_2 \to_d \Gamma_1 \cup \Gamma_2}$$

Then a typing judgment for expressions is $\Gamma \vdash e : \tau \in Basetype$.

Statements and programs have no type.

$$\frac{\Gamma \vdash e_1 : \texttt{int} \qquad \Gamma \vdash e_2 : \texttt{int}}{\Gamma \vdash e_1 + e_2 : \texttt{int}} \qquad \frac{}{\Gamma \vdash x : \Gamma(x)} \qquad \frac{\Gamma \vdash S_1 \qquad \Gamma \vdash S_2}{\Gamma \vdash S_1; S_2}$$

$$\frac{\Gamma \vdash x : \tau \qquad \Gamma \vdash e : \tau}{\Gamma \vdash x := e} \qquad \frac{\Gamma \vdash b : \texttt{bool} \qquad \Gamma \vdash S_1 \qquad \Gamma \vdash S_2}{\Gamma \vdash \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2}$$

$$\frac{\Gamma \vdash b : \texttt{bool} \qquad \Gamma \vdash S}{\Gamma \vdash \texttt{while}\ b\ \texttt{do}\ S\ \texttt{done}} \qquad \frac{D \to_d \Gamma \qquad \Gamma \vdash S}{\emptyset \vdash D; S}$$

(. . . and other similar rules)

$State\ \sigma : Var \to Value$ ($Value = \mathbb{Z} \cup \mathbb{B}$).
Typing configurations:

$$\Gamma \vdash (S, \sigma) \iff (\Gamma \vdash S \wedge \forall x\ \tau, \emptyset \vdash \sigma(x) : \tau \iff \Gamma(x) = \tau)$$

Operational semantics for mini-while

Evaluation

$$Val : \mathcal{E} \to State \to Value$$
$$Val(n, \sigma) = value(n)$$
$$Val(x, \sigma) = \sigma(x)$$
$$Val(e_1 + e_2, \sigma) = Val(e_1, \sigma) + Val(e_2, \sigma)$$
$$\ldots$$

Small step

$(Stm, State) \Rightarrow (Stm, State)$ or $(Stm, State) \Rightarrow State$

$$(x := e, \sigma) \Rightarrow \sigma[x \mapsto Val(e, \sigma)] \qquad (\texttt{skip}, \sigma) \Rightarrow \sigma \qquad \frac{(S_1, \sigma) \Rightarrow \sigma'}{\big((S_1; S_2), \sigma\big) \Rightarrow (S_2, \sigma')}$$

$$\frac{(S_1, \sigma) \Rightarrow (S_1', \sigma')}{\big((S_1; S_2), \sigma\big) \Rightarrow (S_1'; S_2, \sigma')} \qquad \frac{Val(b, \sigma) = tt}{(\texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2, \sigma) \Rightarrow (S_1, \sigma)}$$

$$\frac{Val(b, \sigma) = ff}{(\texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2, \sigma) \Rightarrow (S_2, \sigma)}$$

$$(\texttt{while}\ b\ \texttt{do}\ S\ \texttt{done}, \sigma) \Rightarrow (\texttt{if}\ b\ \texttt{then}\ (S; \texttt{while}\ b\ \texttt{do}\ S\ \texttt{done})\ \texttt{else skip}, \sigma)$$