Statements

1/31

Last time: syntax + semantics of expressions
Today: syntax + semantics of statements

$s ::= x := e \mid a[e] := e \mid \text{if } e \{ s \} \text{ else } \{ s \} \mid \text{while } e \{ s \} \mid s; s \mid \text{skip}$

$s_1; s_2$ sequences statements. Can be nested, right associative
$s_1; s_2; s_3 \equiv s_1; (s_2; s_3)$

Remember: As simple as possible.
e.g. no for loops

```
for (x=0; x<n, x++)   =>   x := 0;
{ s }                      while (x<n) {
                             s;
                             x := x+1
                           }
```

If without else? Use skip

```
r := 1;
while (n>=1)
{                          At end: n≥1 → r=n!
  r := r*n;
  n := n-1
}
```

Remember: No assignment expressions

```
y = x++;   =>   y := x;
                x := x+1;
```

Small-Step Operational Semantics

What a program means

How a program executes

Like a computer executing one step at a time

Configuration $\langle s, \sigma \rangle$ — current state (can change)

↑ current program

$\langle s_1, \sigma_1 \rangle \to \langle s_2, \sigma_2 \rangle$ "steps to"

In one step, s_1 in state σ_1 changes the state to σ_2, program to s_2.

When is it done? $\langle skip, \sigma \rangle$

Program s <u>terminates in</u> state σ' if there's a series of steps (starting in state σ) $\langle s, \sigma \rangle \to \langle s_1, \sigma_1 \rangle \to \ldots \to \langle s', \sigma' \rangle$.

Sometimes just say "s terminates" or "s converges".

Not always true: while (1>0) {skip}

<u>Rules</u> tell us how to step programs

(≥1 rule for each kind of statement)

$$\langle x := e, \sigma \rangle \to \langle skip, \sigma[x \mapsto \sigma(e)] \rangle$$

$$\langle a[e] := e_2, \sigma \rangle \to \langle skip, \sigma[a[\sigma(e_1)] \mapsto \sigma(e_2)] \rangle$$

↑ Update an array element.

$$\frac{\sigma(e) = T}{\langle \text{if } e \text{ then } \{s_1\} \text{ else } \{s_2\}, \sigma \rangle \to \langle s_1, \sigma \rangle}$$

← premise ← inference rule ← conclusion

if [premise] then [conclusion]

$$\frac{\sigma(e) = F}{\langle \text{if } e \text{ then } \{s_1\} \text{ else } \{s_2\}, \sigma \rangle \to \langle s_2, \sigma \rangle}$$

"axiom" ← (inference rule w/ no premises)

$$\langle \text{while } e \{s\}, \sigma \rangle \to \langle \text{if } e \{s; \text{while } e \{s\}\} \text{ else skip}, \sigma \rangle$$

$$\frac{\langle s_1, \sigma \rangle \to \langle s_1', \sigma' \rangle}{\langle s_1; s_2, \sigma \rangle \to \langle s_1'; s_2, \sigma' \rangle}$$

$$\langle \text{skip}; s, \sigma \rangle \to \langle s, \sigma \rangle$$

Examples

$\langle x := x+\bar{1}, \{x=2\} \rangle \to \langle \text{skip}, \{x=2\}[x \mapsto 3] \rangle$

$= \{x=3\}$

$\langle x := x+\bar{1}; y := x, \{x=2\} \rangle \to^2 \langle y := x, \{x=3\} \rangle$

$\to \langle \text{skip}; y := x, \{x=3\} \rangle$

$\to \langle y := x, \{x=3\} \rangle$

$\to \langle \text{skip}, \{x=3\} \rangle$

↑ Can use to step over boring steps

$\to^n$ - n steps $\to^*$ - some $n \geq 0$ # of steps

let W = while $x \geq \bar{0}$ $\{x := x-\bar{1}\}$

$\langle W, \{x=1\} \rangle$

$\to \langle \text{if } x \geq 0 \{x := x-\bar{1}; W\} \text{ else } \{\text{skip}\}, \{x=1\} \rangle$

$\to \langle x := x-\bar{1}; W, \{x=1\} \rangle$

$\to^2 \langle W, \{x=1\}[x \mapsto 0] \rangle$

$\to^3 \langle W, \{x=1\}[x \mapsto 0][x \mapsto -1] \rangle$

$\to^2 \langle \text{skip}, \{x=-1\} \rangle$