We've seen 2 kinds of semantics:

$\sigma(e) = v$ — only care about end result "big step"

$\langle s, \sigma\rangle \to \langle s', \sigma'\rangle$ — care about every step we took to get there "small step"

We can define a big-step semantics for statements too.
We'll write it a little differently:

$M(s, \sigma) = \{\sigma'\} \iff \langle s, \sigma\rangle \to^* \langle \text{skip}, \sigma'\rangle$

↑ it's a set; right now programs are deterministic (only one possible ending state) but we'll change that later

We'll use Σ for sets of states

$M(\text{skip}, \sigma) = \{\sigma\}$ $\quad M(x := e, \sigma) = \{\sigma[x \mapsto \sigma(e)]\}$

$M(a[e_1] := e_2, \sigma) = \{\sigma[a[\sigma(e_1)] \mapsto \sigma(e_2)]\}$

$$M(\text{if } e \ \{s_1\} \text{ else } \{s_2\}, \sigma) = \begin{cases} M(s_1, \sigma) & \sigma(e) = T \\ M(s_2, \sigma) & \sigma(e) = F \end{cases}$$

$M(s_1; s_2, \sigma) = \bigcup_{\sigma' \in M(s_1, \sigma)} M(s_2, \sigma')$

$M(\text{while } e \ \{s\}, \sigma) = ?$

Attempt 1: $M(\text{if } e \ \{s_1\} \text{ else } \{s_2\}, \sigma)$

Not a valid recursive definition: s gets bigger

Let $\Sigma_0 = \{\sigma\}$.

Let $\Sigma_{k+1} = \bigcup_{\sigma \in \Sigma_k} M(s, \sigma)$

Σ_k is the set of states we might have after running s k times.

e.g. $M(\text{while } x \geq 0 \{x := x-1\}, \{x=3\})$

$\Sigma_0 = \{\{x=3\}\}$

$\Sigma_1 = \bigcup_{\sigma \in \{\{x=3\}\}} M(x:=x-1, \sigma) = M(x:=x-1, \{x=3\}) = \{\{x=2\}\}$

$\Sigma_2 = \{\{x=1\}\}$

$\Sigma_3 = \{\{x=0\}\}$

$\Sigma_4 = \{\{x=-1\}\}$

...

Let $M(\text{while } e \{s\})$ be Σ_K where K is the lowest #
such that if $\sigma \in \Sigma_K$, then $\sigma(e) = F$
(first state where it's false)
In the above example, $M(\text{while } x \geq 0 \{x := x-1\}, \{x=3\}) = \Sigma_4$

If there isn't such a K, we have an infinite loop (s "diverges")
In this case, let $M(s, \sigma) = \{\perp_d\}$ ← diverge
"bottom"

Ex. $M(x := \bar{5}; y := x + \bar{1}, \{\})$

$= \bigcup_{\sigma \in M(x:=5, \{\})} M(y := x + \bar{1}, \sigma)$

$M(x:=5, \{\}) = \{x=5\}$

$= M(y := x + \bar{1}, \{x=5\})$

$= \{x=5, y=6\}$

<u>Errors</u>

$\sigma(\overline{42}/\bar{0}) = ?$

We'll say $\sigma(\overline{42}/\bar{0}) = \perp_e$ ← error

So $\sigma(e) \in \text{Values} \cup \{\perp_e\}$

Other examples: $\{a = [1;2]\}(a[\bar{3}]) = \perp_e$ ← OOB array access

$\{x = -1\}(\text{sqrt}(x)) = \perp_e$ ← sqrt of a neg. #

$\sigma(1 + 2) = \perp_e$ ← runtime type err

Hereditary failure

$\sigma(3 + 42/0) = ?$

$\sigma(e_1 \text{ op } e_2) = \bot_e$ if $\sigma(e_1) = \bot_e$ or $\sigma(e_2) = \bot_e$

$\sigma(e_1 ? e_2 : e_3) = \bot_e$ if 1. $\sigma(e_1) = \bot_e$ or

2. $\sigma(e_1) = T$ and $\sigma(e_2) = \bot_e$ or

3. $\sigma(e_1) = F$ and $\sigma(e_3) = \bot_e$

Note: We <u>don't</u> fail if the not-taken branch fails

This lets us do, e.g. $\sigma(x = \bar{0} ? \bar{1} : y/x)$

<u>Errors in statements</u>

Write $\langle s, \sigma \rangle \to \langle skip, \bot_e \rangle$ when a step causes us to error

$$\frac{\sigma(e) = \bot_e}{\langle x := e, \sigma \rangle \to \langle skip, \bot_e \rangle} \qquad \frac{\sigma(e_1) = \bot_e \ (e_2)}{\langle a[e_1] := e_2, \sigma \rangle \to \langle skip, \bot_e \rangle}$$

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

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

Big-step: $M(s, \sigma) = \{\bot_e\}$ if $\langle s, \sigma \rangle \to^* \langle skip, \bot_e \rangle$

We'll use $\bot$ for $\bot_d$ or $\bot_e$

Note: $\bot$ appears in some of the places a state does (and $\bot_e$ in some of the places a value does) but it's ~~not~~ not a state or value. In particular, don't write:

$\cancel{\bot[x \mapsto 1]}$ $\quad \cancel{\bot(x)}$ $\quad \cancel{\bot(e)}$

$\cancel{\sigma[x \mapsto \bot]}$ $\quad \cancel{M(s, \bot)}$