CS 4110 - 9/12/2025
Lab #8 - IMP

Suppose we extend IMP with concurrent composition?

$c ::= \text{skip}$
$\quad | \; x := a$
$\quad | \; c_1 ; c_2$
$\quad | \; \text{if } b \text{ then } c_1 \text{ else } c_2$
$\quad | \; \text{while } b \text{ do } c$
$\quad | \; c_1 \,\|\| \, c_2$

How can we give semantics to $c_1 \,\|\| \, c_2$?

As an example, consider the following program

$x := 0;$
$y := 0;$
$\left(\begin{array}{l} x := y+1; \\ y := y+1 \end{array} \;\|\|\; y := y+1 \right)$

Attempt #1: Big-Step semantics

$$\frac{\langle \sigma, c_1 \rangle \Downarrow \sigma_1 \qquad \langle \sigma_1, c_2 \rangle \Downarrow \sigma'}{\langle \sigma, c_1 \,\|\| \, c_2 \rangle \Downarrow \sigma'}$$

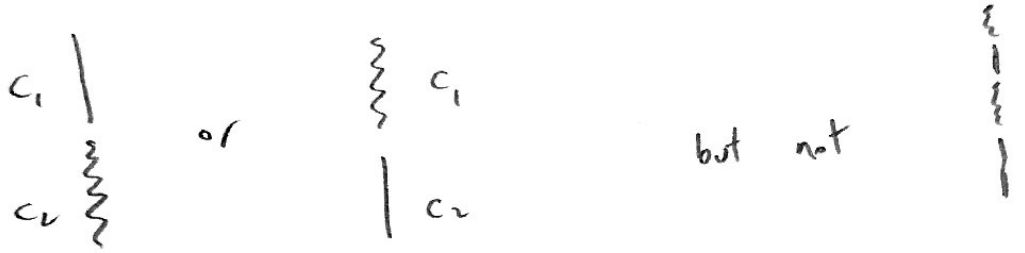
But this is identical to $c_1 ; c_2$!

Attempt #2: Big-step with symmetric rules

We can add a symmetric rule

$$\frac{\langle \sigma, c_2 \rangle \Downarrow \sigma_2 \quad \langle \sigma_2, c_1 \rangle \Downarrow \sigma'}{\langle \sigma, c_1 \| c_2 \rangle \Downarrow \sigma'}$$

But this doesn't capture interleavings of commands in c_1 and c_2.



Attempt #3: Small-step semantics

$$\frac{\langle \sigma, c_1 \rangle \to \langle \sigma', c_1' \rangle}{\langle \sigma, c_1 \| c_2 \rangle \to \langle \sigma', c_1' \| c_2 \rangle} \qquad \frac{\langle \sigma, c_2 \rangle \to \langle \sigma', c_2' \rangle}{\langle \sigma, c_1 \| c_2 \rangle \to \langle \sigma', c_1 \| c_2' \rangle}$$

Are we done? No! Need to eliminate $\|$ when subcomputations are done.

$$\frac{}{\langle \sigma, \text{skip} \| \text{skip} \rangle \to \langle \sigma, \text{skip} \rangle} \quad \text{or} \quad \frac{}{\langle \sigma, c_1 \| \text{skip} \rangle \to \langle \sigma, c_1 \rangle} \qquad \frac{}{\langle \sigma, \text{skip} \| c_2 \rangle \to \langle \sigma, c_2 \rangle}$$

If we don't have these rules, $(x := 3 \| y := 4); z := x + y$

will get stuck at $(\text{skip} \| \text{skip}); z := x + y$.

What about our example program? What are possible final values of x and y?

x	y
1	2
2	2
1	1 ← if assignments don't happen atomically!

II Proofs by induction on derivations.

<u>Theorem</u>: If $\sigma(i)$ is even, and $\langle \sigma, \overbrace{\text{while } b \text{ do } i := i+2}^{c} \rangle \Downarrow \sigma'$, then $\sigma'(i)$ is also even.

<u>Proof</u>: By induction on derivation of $\langle \sigma, c \rangle \Downarrow \sigma'$.

There is a case for every rule in the big-step semantics, but all rules whose conclusion doesn't match c vacuously hold.

<u>Case While-F:</u>

The result is immediate: $\sigma'(i)$ is even as $\sigma = \sigma'$ and $\sigma(i)$ is even

<u>Case While-T:</u>

We have two sub-derivations, $\langle \sigma, i+2 \rangle \Downarrow \sigma_1$ and $\langle \sigma_1, c \rangle \Downarrow \sigma'$

By inversion* on $\langle \sigma, i := i+2 \rangle \Downarrow \sigma_1$, we have that $\sigma_1(i) = n+2$ where $\sigma(i) = n$.

As n is even by assumption, $n+2$ is also even.

By IH applied to the sub-derivation $\langle \sigma_1, c \rangle \Downarrow \sigma'$, we have that $\sigma'(i)$ is even, which finishes the case, and the proof. □

Lemma [Inversion]

If $\langle \sigma, x := a \rangle \Downarrow \sigma'$ and $\langle \sigma, a \rangle \Downarrow n$ then $\sigma'(x) = n$.