Operational Semantics

Key Idea: Program Execution ≈ Constructing a derivation.
≈ Finding a final state

$$\dfrac{\dfrac{}{x \leftarrow 1, [\,] \Downarrow \sigma^{?}[x \mapsto 1]} \quad \dfrac{}{y \leftarrow x * 2, [x \mapsto 1] \Downarrow \sigma^{?}[x \mapsto 1, y \mapsto 2]}}{x \leftarrow 1; y \leftarrow x * 2, [\,] \Downarrow \sigma^{?}[x \mapsto 1, y \mapsto 2]}$$

→ Final State Uncovered from the construction of the derivation.

Apply rules bottom-up

$$\dfrac{\dfrac{}{x \leftarrow x+2, [\,] \Downarrow \sigma^{?}[x \mapsto 2]} \quad \dfrac{}{\text{while } x \leq 1 \text{ do } x \leftarrow x+2, [x \mapsto 2] \Downarrow \sigma^{?}[x \mapsto 2]}}{\text{while } x \leq 1 \text{ do } x \leftarrow x+2, [\,] \Downarrow \sigma^{?}[x \mapsto 2]}$$

While Language statements

$$S \to \text{skip} \mid S ; S \mid x \leftarrow A \mid \text{if } B \text{ then } S \text{ else } S$$
$$\mid \text{while } B \text{ do } S$$

Operational Semantics Relation / Set of triples

$$\Downarrow \; \subseteq \underbrace{S}_{\text{program}} \times \underbrace{\text{State}}_{\text{initial state}} \times \underbrace{\text{State}}_{\text{final state}}$$

$$S, \sigma \Downarrow \sigma' \iff (S, \sigma, \sigma') \in \Downarrow$$

Convenient infix notation for set inclusion.

Inductively Defined Set.

$$N \to 0 \mid N+1$$

*) $0 \in N$ *) if $n \in N$, then $n+1 \in N$.

Inference Rule

$$\dfrac{P_1 \; \cdots \; P_n}{Q}$$

$P_1 \cdots P_n$ ← premises; Q ← conclusion } Judgements.

$$\dfrac{}{0 \in N} \qquad \dfrac{n \in N}{\underset{\text{metavariable}}{n} + 1 \in N}$$

metavariable are universally quantified.

$$\dfrac{\dfrac{\dfrac{}{0 \in N}}{0+1 \in N}}{0+1+1 \in N}$$ } Derivation Tree.

$$\dfrac{}{0 \in N} \qquad \underbrace{\dfrac{0 \in N}{\underset{n}{0}+1 \in N} \qquad \dfrac{0+1 \in N}{\underset{n}{0+1}+1 \in N}}_{\text{Instances of the rule: } \frac{n \in N}{n+1 \in N}}$$

Inductively defined set is the least set that satisfies a set of inference rules.
least → all elements come from rules.

Inference Rules for Operational Semantics

$$\dfrac{}{\text{skip}, \underset{\text{metavariable}}{\sigma} \Downarrow \sigma} \qquad \dfrac{}{x \leftarrow e, \sigma \Downarrow \underbrace{\sigma[x \mapsto [\![e]\!]_A(\sigma)]}_{\text{State update. e.g.}}} \qquad \dfrac{S_1, \sigma \Downarrow \sigma' \quad S_2, \sigma' \Downarrow \sigma''}{S_1 ; S_2, \sigma \Downarrow \sigma''}$$

$$\sigma[x \mapsto 7](y) = \begin{cases} 7 & \text{if } y = x \\ \sigma(y) & \text{o/w} \end{cases}$$

$$\dfrac{S_1, \sigma \Downarrow \sigma'}{\text{if } e \text{ then } S_1 \text{ else } S_2, \sigma \Downarrow \sigma'} \; [\![e]\!]_B(\sigma) = T \qquad \dfrac{S_2, \sigma \Downarrow \sigma'}{\text{if } e \text{ then } S_1 \text{ else } S_2, \sigma \Downarrow \sigma'} \; [\![e]\!]_B(\sigma) = \bot$$

e, σ: metavariables

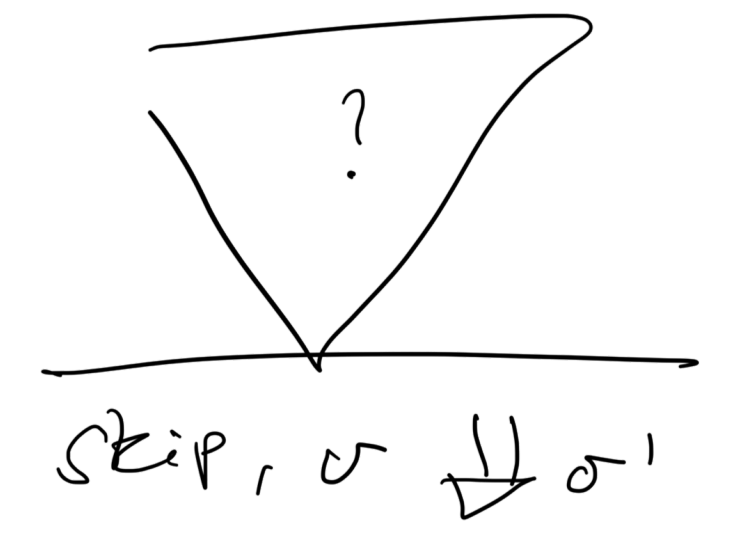
$[\![e]\!]_B(\sigma) = T$ → Side Conditions restrict metavariables

$$\dfrac{S, \sigma \Downarrow \sigma' \quad \text{while } e \text{ do } S, \sigma' \Downarrow \sigma''}{\text{while } e \text{ do } S, \sigma \Downarrow \sigma''} \; [\![e]\!]_B(\sigma) = T \qquad \dfrac{}{\text{while } e \text{ do } S, \sigma \Downarrow \sigma} \; [\![e]\!]_B(\sigma) = \bot$$

$(= S; \text{while } e \text{ do } S)$

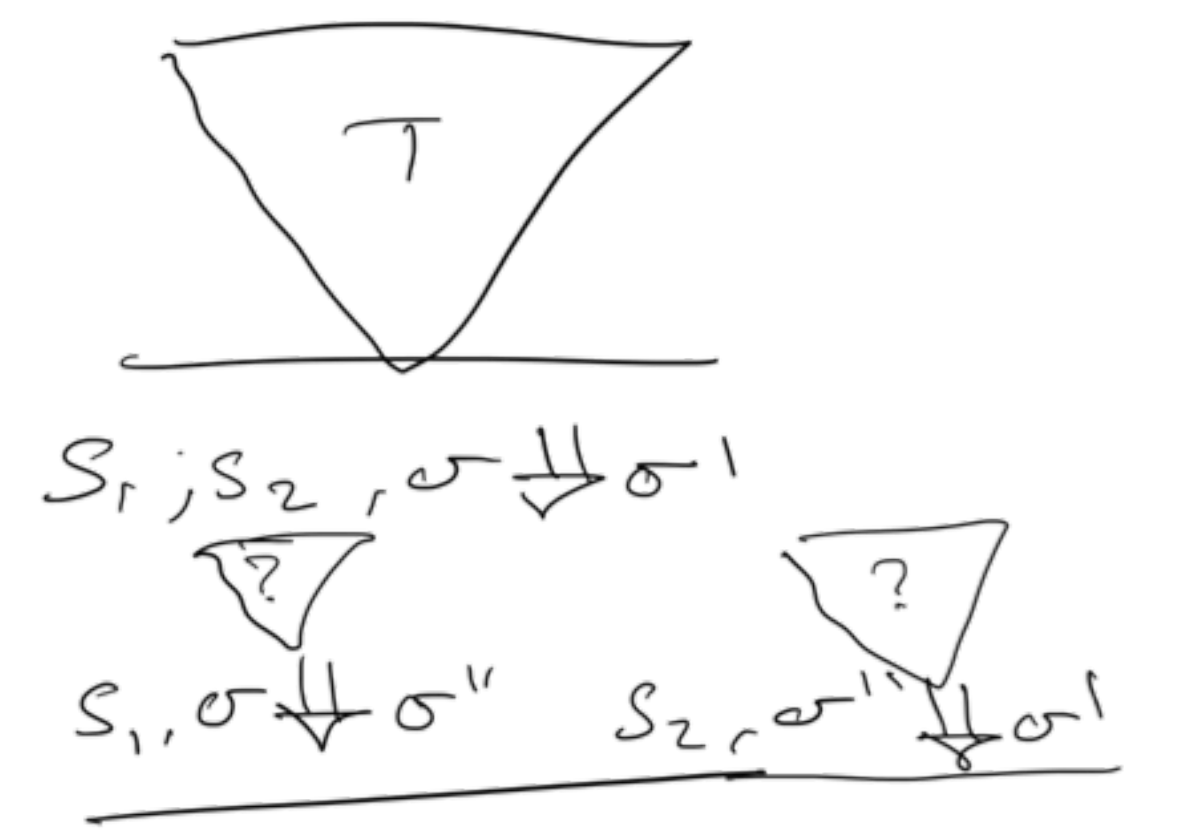
Inversion Principle

$$\text{skip}, \sigma \Downarrow \sigma'$$
$$\Rightarrow \sigma = \sigma'$$



$$\text{skip}, \sigma \Downarrow \sigma' \equiv \dfrac{}{\text{skip}, \sigma \Downarrow \sigma}$$

$$S_1 ; S_2, \sigma \Downarrow \sigma'$$
$$\Rightarrow S_1, \sigma \Downarrow \sigma'' \text{ and } S_2, \sigma'' \Downarrow \sigma'$$



$$S_1 ; S_2, \sigma \Downarrow \sigma' \equiv \dfrac{S_1, \sigma \Downarrow \sigma'' \quad S_2, \sigma'' \Downarrow \sigma'}{S_1 ; S_2, \sigma \Downarrow \sigma'}$$

More on this next time