Intro to Formal Semantics, Big step semantics

Mathematically describe a program's behavior
- → rigorous/precise
- → abstract/symbolic representation
- → amenable to proof methods

Types of semantics
- Operational semantics - describe execution of a program
- Denotational semantics - model as another type of math object

Types of operational semantics
- Big step $e \Downarrow v$ "e evaluates to v" (one big step)
- Small step $e \to e' \to e'' \ldots \to e$ show all steps

To assert/prove things about behavior of programs:
1. Make assertions about objects (syntax, typing, execution, etc.)
2. Connect these assertions
3. Build up into larger narratives

Judgment - type of thing we might want to assert

n is odd $\quad e \to e' \quad e \Downarrow v \quad \vdash e : \tau$ ← e has type τ

We assert/define judgments w/ inference rules

$$\text{(NAME)}\ \frac{\text{premise}_1 \ \cdots \ \text{premise}_n}{\text{conclusion}}$$ If premises hold, conclusion holds

Axiom - rule w/ no premises

Ex. even / odd

$$\text{OEVEN}\ \dfrac{}{0 \text{ is even}} \qquad \text{INCO}\ \dfrac{n \text{ is odd}}{n+1 \text{ is even}} \qquad \text{INCE}\ \dfrac{n \text{ is even}}{n+1 \text{ is odd}}$$

$$\text{EVL}\ \dfrac{x \text{ is even}}{x * y \text{ is even}} \qquad \text{EVR}\ \dfrac{y \text{ is even}}{x * y \text{ is even}} \qquad \text{ODD ODD}\ \dfrac{x \text{ is odd} \quad y \text{ is odd}}{x * y \text{ is odd}}$$

Proof trees ("derivations") → End w/ axioms

Build up a proof of C using rules

$$\dfrac{\dfrac{\overline{P_4}}{P_1} \qquad \overline{P_2} \qquad \dfrac{\dfrac{\overline{P_7}}{P_5} \quad \overline{P_6}}{P_3}}{C}$$

→ Each rule should be a rule from our system

Building a proof tree:

Can use forward chaining - start w/ axioms - what can we derive?
may not be able to reach goal, may take a long time

backward chaining - start w/ conclusion

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{0 \text{ is even}}}{1 \text{ is odd}}}{2 \text{ is even}} \qquad \dfrac{\dfrac{\dfrac{\overline{0 \text{ is even}}}{1 \text{ is odd}}}{2 \text{ is even}}}{3 \text{ is odd}}}{\dfrac{6 \text{ is even}}{7 \text{ is odd}}}$$

IMP - A simple imperative language

Arithmetic exps $E ::= \bar{n} \mid x \mid E + E$

$\bar{n}$: Any int ($\bar{0}, \overline{42}, \dots$) x: Variables

Boolean exps $B ::= \text{true} \mid \text{false} \mid E = E$

Statements $S ::= \text{skip}$ ← no-op ("pass" in Python)

$\mid x := E$

$\mid S; S$

$\mid \text{if } B \text{ then } S_1 \text{ else } S_2$

$\mid \text{while } B \text{ do } S$

Evaluation $\langle E,\sigma\rangle \Downarrow_e n$ Under environment σ, E eval's to n

$\langle B,\sigma\rangle \Downarrow_b b$ " B " b (true/false)

$\langle S,\sigma\rangle \Downarrow_s \sigma'$ " S gives new env σ'.

"side condition"

$$\text{INT-LIT}\ \frac{}{\langle n,\sigma\rangle \Downarrow_e n}\ n\in\mathbb{Z} \qquad \text{VAR}\ \frac{\sigma(x)=v}{\langle x,\sigma\rangle \Downarrow v} \qquad \text{EQ-T}\ \frac{\langle e_1,\sigma\rangle\Downarrow n \quad \langle e_2,\sigma\rangle\Downarrow n}{\langle e_1=e_2,\sigma\rangle \Downarrow \text{true}}$$

$$\text{PLUS}\ \frac{\langle e_1,\sigma\rangle\Downarrow_e n_1 \qquad \langle e_2,\sigma\rangle\Downarrow_e v_2}{\langle e_1+e_2,\sigma\rangle\Downarrow_e n_1+n_2} \qquad \text{B-LIT}\ \frac{}{\langle b,\sigma\rangle\Downarrow_b b}\ b\in\{\text{true}, \text{false}$$

$$\text{SKIP}\ \frac{}{\langle \text{skip},\sigma\rangle \Downarrow_s \sigma} \qquad \text{ASSIGN}\ \frac{\langle e,\sigma\rangle\Downarrow_e v}{\langle x:=e,\sigma\rangle\Downarrow_s \sigma[x\mapsto v]} \qquad \text{SEQ}\ \frac{\langle S_1,\sigma\rangle\Downarrow \sigma' \quad \langle S_2,\sigma'\rangle\Downarrow\sigma''}{\langle S_1;S_2,\sigma\rangle\Downarrow\sigma''}$$

σ ext. w/ $x\mapsto v$

$$\text{IF-T}\ \frac{\langle B,\sigma\rangle\Downarrow_b \text{true} \quad \langle S_1,\sigma\rangle\Downarrow_s\sigma'}{\langle \text{if } B \text{ then } S_1 \text{ else } S_2,\sigma\rangle\Downarrow_s\sigma'} \qquad \text{IF-F}\ \frac{\langle B,\sigma\rangle\Downarrow_b \text{false} \quad \langle S_2,\sigma\rangle\Downarrow_s\sigma'}{\langle \text{if } B \text{ then } S_1 \text{ else } S_2,\sigma\rangle\Downarrow_s\sigma'}$$

$$\text{WHILE-F}\ \frac{\langle B,\sigma\rangle\Downarrow_b \text{false}}{\langle \text{while } B \text{ do } S,\sigma\rangle\Downarrow_s\sigma} \qquad \text{WHILE-T}\ \frac{\langle B,\sigma\rangle\Downarrow \text{true} \quad \langle S; \text{while } B \text{ do } S,\sigma\rangle\Downarrow_s\sigma'}{\langle \text{while } B \text{ do } S,\sigma\rangle\Downarrow_s\sigma'}$$

$$\text{EQ-F}\ \frac{\langle e_1,\sigma\rangle\Downarrow n_1 \quad \langle e_2,\sigma\rangle\Downarrow n_2 \quad n_1\neq n_2}{\langle e_1=e_2,\sigma\rangle\Downarrow \text{false}}$$

Prove: $\langle (3+a)+(4+b), \{a\mapsto 5, b\mapsto 6\}\rangle \Downarrow 18$

$$\text{PLUS}\ \dfrac{\text{PLUS}\ \dfrac{\text{I-LIT}\ \dfrac{}{\langle 3,\sigma\rangle\Downarrow 3} \quad \text{VAR}\ \dfrac{\sigma(a)=5}{\langle a,\sigma\rangle\Downarrow 5}}{\langle 3+a,\sigma\rangle\Downarrow 8} \qquad \text{PLUS}\ \dfrac{\text{ILIT}\ \dfrac{}{\langle 4,\sigma\rangle\Downarrow 4} \quad \text{VAR}\ \dfrac{\sigma(b)=6}{\langle b,\sigma\rangle\Downarrow 6}}{\langle 4+b,\sigma\rangle\Downarrow 10}}{\langle (3+a)+(4+b),\sigma\rangle\Downarrow 18}$$

Prove: $\overset{W=}{\text{While } B \text{ do } S}$ is equivalent to if B then (S; while B do S) else skip

WTS: $\langle W, \sigma\rangle \Downarrow \sigma' \iff \langle \text{if } B \text{ then } (S;W) \text{ else skip}, \sigma\rangle \Downarrow \sigma'$

$\iff$: if and only if

$\Rightarrow$ Assume $\langle W, \sigma\rangle \Downarrow \sigma'$ so ① $\dfrac{\overline{\langle B,\sigma\rangle \Downarrow \text{false}}^{\,\vdots *}}{\langle W,\sigma\rangle \Downarrow \sigma}$ or ② $\dfrac{\overline{\langle B,\sigma\rangle \Downarrow \text{true}}^{\,\vdots **} \quad \overline{\langle S;W,\sigma\rangle \Downarrow \sigma'}^{\,\vdots ***}}{\langle W,\sigma\rangle \Downarrow \sigma'}$

①

$$\text{IF-F}\ \dfrac{\overline{\langle B,\sigma\rangle \Downarrow \text{false}}^{\,\vdots *} \quad \text{SKIP}\ \overline{\langle \text{skip},\sigma\rangle \Downarrow \sigma}}{\langle \text{if } B \text{ then } S;W \text{ else skip}, \sigma\rangle \Downarrow \sigma}$$

②

$$\text{IF-T}\ \dfrac{\overline{\langle B,\sigma\rangle \Downarrow \text{true}}^{\,\vdots **} \quad \overline{\langle S;W,\sigma\rangle \Downarrow \sigma'}^{\,\vdots ***}}{\langle \text{if } B \text{ then } S;W \text{ else skip}, \sigma\rangle \Downarrow \sigma'}$$

$\Leftarrow$ Assume $\langle \text{if } B \text{ then } (S;W) \text{ else skip}, \sigma\rangle \Downarrow \sigma'$

so ① $\dfrac{\overline{\langle B,\sigma\rangle \Downarrow \text{true}}^{\,\vdots *} \quad \overline{\langle S;W,\sigma\rangle \Downarrow \sigma'}^{\,\vdots **}}{\langle \text{if } B \text{ then } (S;W) \text{ else skip}, \sigma\rangle \Downarrow \sigma'}$ or ② $\dfrac{\overline{\langle B,\sigma\rangle \Downarrow \text{false}}^{\,\vdots *} \quad \overline{\langle \text{skip},\sigma\rangle \Downarrow \sigma}}{\langle \text{if } B \text{ then } (S;W) \text{ else skip}, \sigma\rangle \Downarrow \sigma}$

① $$\text{WHILE-T}\ \dfrac{\overline{\langle B,\sigma\rangle \Downarrow \text{true}}^{\,\vdots *} \quad \overline{\langle S;W,\sigma\rangle \Downarrow \sigma}^{\,\vdots **}}{\langle W,\sigma\rangle \Downarrow \sigma'}$$

② $$\text{WHILE-F}\ \dfrac{\overline{\langle B,\sigma\rangle \Downarrow \text{false}}^{\,\vdots ***}}{\langle W,\sigma\rangle \Downarrow \sigma}$$