STATIC SEMANTICS

Goal: make judgments/check things about a program before running it.

What should the program do if uninit. vars are read?
Three common strategies:

1. Crash-and-burn:
   What languages do this?

2. Junk data:
   What languages do this?

3. Default value:
   What languages do this?

Try all this after A0

Ex. If $\Phi; \phi \vdash s$ ok $\rightsquigarrow \phi$, what can we say about s?
(Informally, then formally)

* All:

* C&B:

* JD, DF:

Try all this after A0

SEQUENTIAL COMPOSITION IN TINYIMP

Claim: $(\sigma_1, (s_1; s_2); s_3) \Downarrow \sigma_2 \iff (\sigma_1, s_1; (s_2; s_3)) \Downarrow \sigma_2$

Proof: The $\Leftarrow$ case will be sim. to $\Rightarrow$, so we just prove that

($\Rightarrow$) Assume $(\sigma_1, (s_1; s_2); s_3) \Downarrow \sigma_2$. Then there must be states σ, σ' such that the proof of this assumption starts off like

$$\dfrac{\dfrac{\dfrac{\vdots}{(\sigma_1, s_1) \Downarrow \sigma'} \quad \dfrac{\vdots}{(\sigma', s_2) \Downarrow \sigma}}{(\sigma_1, s_1; s_2) \Downarrow \sigma} \quad \dfrac{\vdots}{(\sigma, s_3) \Downarrow \sigma_2}}{(\sigma_1, (s_1; s_2); s_3) \Downarrow \sigma_2}$$

We can now prove the RHS evaluates similarly:

$$\dfrac{\dfrac{}{(\sigma_1, s_1) \Downarrow \sigma'} \quad \dfrac{\dfrac{}{(\sigma', s_2) \Downarrow \sigma} \quad \dfrac{}{(\sigma, s_3) \Downarrow \sigma_2}}{(\sigma', s_2; s_3) \Downarrow \sigma_2}}{(\sigma_1, s_1; (s_2; s_3)) \Downarrow \sigma_2}$$

DO ... UNTIL LOOPS IN TINYIMP

do s until e = repeat s until e evaluates to true, doing it at least once

Ex. 1: Big-step semantics for this construct: try this after AO

Ex. 2: do s until e ≡ s; while ¬e do s od

AXIOMATIC SEMANTICS, HOARE LOGIC

Goal: meaning through correctness. Commonly via Hoare logic.

Takes the form of triples: $\{\varphi\}\ s\ \{\psi\}$

- φ: precondition: what is true before
- s: statement (line of code, program, ...)
- ψ: postcondition: what we want to be true after

Specifies a proof calculus for proving desirable properties about your program (e.g. that it returns the right value).

Ex. Prove the rule for do...until is sound using our translation:

$$\frac{\{\varphi\}\ s\ \{\varphi\}}{\{\varphi\}\ \text{do } s \text{ until } e\ \{\varphi \wedge e\}}$$

Proof. Assume $\{\varphi\}\ s\ \{\varphi\}$. We will prove 2 lemmas first:

#1:
$$\frac{\{\varphi\}\ s\ \{\varphi\}}{\{\varphi\}\ \text{while } \neg e \text{ do } s \text{ od}\ \{\varphi \wedge \neg(\neg e)\}}\ L_1$$

#2:
$$\frac{\{\varphi\}\ s\ \{\varphi\}}{\{\varphi\}\ \text{while } \neg e \text{ do } s \text{ od}\ \{\varphi \wedge e\}}\ L_2$$

Assuming them for now, our main goal can be proven:

$$\frac{\dfrac{\{\varphi\}\ s\ \{\varphi\}}{\{\varphi\}\ s\ \{\varphi\}}\ \text{(Assumed)} \qquad \dfrac{\{\varphi\}\ s\ \{\varphi\}}{\{\varphi\}\ \text{while } \neg e \text{ do } s \text{ od}\ \{\varphi \wedge e\}}\ \text{(Assumed)},\ L_2}{\{\varphi\}\ s;\ \text{while } \neg e \text{ do } s \text{ od}\ \{\varphi \wedge e\}}\ \text{SEQ}$$

So let us now prove those lemmas:

Proof of L_1:

$$\dfrac{\dfrac{\dfrac{}{\varphi \wedge \neg e \Rightarrow \varphi}\,(\text{logic}) \qquad \dfrac{}{\{\psi\}\, s\, \{\psi\}}\,(\text{Assumed}) \qquad \dfrac{}{\psi \Rightarrow \psi}\,(\text{logic})}{\{\varphi \wedge \neg e\}\, s\, \{\varphi\}}\,\text{conseq.}}{\{\varphi\}\ \text{while}\ \neg e\ \text{do}\ s\ \text{od}\ \{\varphi \wedge \neg(\neg e)\}}\,\text{LOOP}$$

Proof of L_2:

$$\dfrac{\dfrac{\dfrac{}{\varphi \Rightarrow \varphi}\,(\text{logic}) \qquad \dfrac{\dfrac{}{\{\psi\}\, s\, \{\psi\}}\,(\text{Assumed})}{\{\varphi\}\ \text{while}\ \neg e\ \text{do}\ s\ \text{od}\ \{\varphi \wedge \neg(\neg e)\}}\,L_1 \qquad \dfrac{}{\varphi \wedge \neg(\neg e) \Rightarrow \varphi \wedge e}\,(\text{logic})}{\{\varphi\}\ \text{while}\ \neg e\ \text{do}\ s\ \text{od}\ \{\varphi \wedge e\}}\,\text{conseq.}$$