

Complete the interpreter (code provided) to support procedures parametrized by expressions under call-by-name (lazy) semantics. The syntax, typing rules and semantic equations for this extension are found in Schmidt. You will need to extend *meaning\_of\_declaration* to properly handle procedure declarations with lazily evaluated  $\tau\text{exp}$  parameters that will be evaluated lazily.

The equations to implement are on page 80<sup>1</sup>.

$$\begin{aligned} \llbracket \pi \vdash \text{proc } I_1(I_2 : \tau\text{exp}) = C : \{I_1 : \tau\text{exp} \rightarrow \text{comm}\} \text{dec} \rrbracket e s &= (\{I_1 : p\}, s) \\ \text{where } p f s' &= \llbracket \pi \uplus \{I_2 : \tau\text{exp}\} \vdash C : \text{comm} \rrbracket (e \uplus \{I_2 : f\}) s' \end{aligned}$$

This rule becomes two rules when we specialize  $\tau$  to be *int* or *bool* as follows:

$$\begin{aligned} \llbracket \pi \vdash \text{proc } I_1(I_2 : \text{intexp}) = C : \{I_1 : \text{intexp} \rightarrow \text{comm}\} \text{dec} \rrbracket e s &= (\{I_1 : p\}, s) \\ \text{where } p f s' &= \llbracket \pi \uplus \{I_2 : \text{intexp}\} \vdash C : \text{comm} \rrbracket (e \uplus \{I_2 : f\}) s' \\ \llbracket \pi \vdash \text{proc } I_1(I_2 : \text{boolexp}) = C : \{I_1 : \text{boolexp} \rightarrow \text{comm}\} \text{dec} \rrbracket e s &= (\{I_1 : p\}, s) \\ \text{where } p f s' &= \llbracket \pi \uplus \{I_2 : \text{boolexp}\} \vdash C : \text{comm} \rrbracket (e \uplus \{I_2 : f\}) s' \end{aligned}$$

For the first rule, you update the `proc_int` environment and for the second you must update the `proc_bool` environment.

Note that for type assignments,  $\pi_1 \uplus \pi_2$  is implemented by the following ML code: `(bar_union_ta pi1 pi2)` and for environments  $e_1 \uplus e_2$  is implemented by the call `(union_env e1 e2)`.

For completeness I include the semantics of the call here as well (though it is implemented in the code provided).

$$\begin{aligned} \llbracket \pi \vdash \text{call } I_1(E) : \text{comm} \rrbracket e s &= p f s \\ \text{where } \langle I_1, p \rangle &\in e \text{ and } f s' = \llbracket \pi \vdash E : \tau\text{exp} \rrbracket e s \end{aligned}$$

---

<sup>1</sup>Note that the line above the semantic equations on page 80 should say

$$p \in \llbracket \tau\text{exp} \rrbracket \rightarrow \llbracket \text{comm} \rrbracket$$

The inclusion of the  $Env_\pi$  argument to  $p$  is incorrect. You can see this in the definition of the type `env` in the code.