

## Semantica big-step

$$\begin{array}{c}
\text{(B-Num)} \frac{-}{\langle n, s \rangle \Downarrow n} \quad \text{(B-Loc)} \frac{-}{\langle l, s \rangle \Downarrow s(l)} \quad \text{(B-Skip)} \frac{-}{\langle \text{skip}, s \rangle \Downarrow s} \quad \text{(B-Add)} \frac{\langle E_1, s \rangle \Downarrow n_1 \quad \langle E_2, s \rangle \Downarrow n_2}{\langle E_1 + E_2 \rangle \Downarrow n_3} n_3 = \text{add}(n_1, n_2) \\
\\
\text{(B-Assign)} \frac{\langle E, s \rangle \Downarrow n}{\langle l := e, s \rangle \Downarrow s[l \mapsto n]} \quad \text{(B-Assign.s)} \frac{\langle E, s \rangle \Downarrow n}{\langle l := e, s \rangle \Downarrow \langle \text{skip}, s[l \mapsto n] \rangle} \quad \text{(B-Seq)} \frac{\langle C_1, s \rangle \Downarrow s_1 \quad \langle C_2, s_1 \rangle \Downarrow s'}{\langle C_1; C_2, s \rangle \Downarrow s'} \quad \text{(B-Seq.s)} \frac{\langle C_1, s \rangle \Downarrow \langle \text{skip}, s_1 \rangle \quad \langle C_2, s_1 \rangle \Downarrow \langle r, s' \rangle}{\langle C_1; C_2, s \rangle \Downarrow \langle r, s' \rangle} \\
\\
\text{(B-If.T)} \frac{\langle B, s \rangle \Downarrow \text{true} \quad \langle C_1, s \rangle \Downarrow s'}{\langle \text{if } B \text{ then } C_1 \text{ else } C_2, s \rangle \Downarrow s'} \quad \text{(B-If.T)} \frac{\langle B, s \rangle \Downarrow \text{true} \quad \langle C_1, s \rangle \Downarrow \langle r, s' \rangle}{\langle \text{if } B \text{ then } C_1 \text{ else } C_2, s \rangle \Downarrow \langle r, s' \rangle} \quad \text{(B-If.F)} \frac{\langle B, s \rangle \Downarrow \text{false} \quad \langle C_2, s \rangle \Downarrow s'}{\langle \text{if } B \text{ then } C_1 \text{ else } C_2, s \rangle \Downarrow s'} \quad \text{(B-If.F)} \frac{\langle B, s \rangle \Downarrow \text{false} \quad \langle C_2, s \rangle \Downarrow \langle r, s' \rangle}{\langle \text{if } B \text{ then } C_1 \text{ else } C_2, s \rangle \Downarrow \langle r, s \rangle} \\
\\
\text{(B-While.T)} \frac{\langle B, s \rangle \Downarrow \text{true} \quad \langle C, s \rangle \Downarrow s_1 \quad \langle \text{while } B \text{ do } C, s_1 \rangle \Downarrow s'}{\langle \text{while } B \text{ do } C, s \rangle \Downarrow s'} \quad \text{(B-While.F)} \frac{\langle B, s \rangle \Downarrow \text{false}}{\langle \text{while } B \text{ do } C, s \rangle \Downarrow s} \quad \text{(B-Do)} \frac{\langle B, s \rangle \Downarrow v_1 \quad \langle C, s \rangle \Downarrow v_2}{\langle \text{do } E \text{ return } C, s \rangle \Downarrow v_2} \\
\\
\text{(B-While.UN)} \frac{\langle \text{if } B \text{ then } (C; \text{while } B \text{ do } C) \text{ else skip}, s \rangle \Downarrow r, s'}{\langle \text{while } B \text{ do } C, s \rangle \Downarrow r, s'} \quad \text{(B-Await)} \frac{\langle B, s \rangle \Downarrow \text{true}, s_1 \quad \langle C, s_1 \rangle \rightarrow^* \langle \text{skip}, s' \rangle}{\langle \text{await } B \text{ protect } C \text{ end}, s \rangle \rightarrow \langle \text{skip}, s' \rangle} \\
\\
\text{(B-Let-CBV)} \frac{P \Downarrow m \quad E\{m/x\} \Downarrow n}{\text{let } x = P \text{ in } E \Downarrow n} \quad \text{(B-Let-CBN)} \frac{E\{P/x\} \Downarrow n}{\text{let } x = P \text{ in } E \Downarrow n} \quad \text{(B-Fn)} \frac{-}{\langle \text{fn } x: T \Rightarrow e, s \rangle \rightarrow \langle \text{fn } x: T \Rightarrow e, s \rangle} \\
\\
\text{(B-App-CBV)} \frac{E_1 \Downarrow \text{fn } x: T \Rightarrow E \quad E\{E_2/x\} \Downarrow n}{E_1 E_2 \Downarrow n} \quad \text{(B-App-CBN)} \frac{E_1 \Downarrow \text{fn } x: T \Rightarrow E \quad E_2 \Downarrow v \quad E\{v/x\} \Downarrow n}{E_1 E_2 \Downarrow n}
\end{array}$$

# Semantica small-step

## Grammatica delle espressioni

$$\text{op} :: = + \mid \geq$$

$$e \in \text{Exp} :: = n \mid b \mid e \text{ op } e \mid \text{if } e \text{ then } e \text{ else } e \mid \text{skip} \mid e; e \mid \text{while } e \text{ do } e \mid e_1 : = e_2 \mid !e \mid \text{ref } e \mid l \mid \text{fn } x : T \Rightarrow e \mid e_1 e_2 \mid \text{let } x : T = e \text{ in } e \mid e \oplus e \mid e \parallel e \mid \text{await } e \text{ protect } e \text{ end} \mid \# \text{lab } e$$

## Regole per la semantica

$$(\text{S-Left}) \frac{E_1 \rightarrow E'_1}{E_1 + E_2 \rightarrow E'_1 + E_2}$$

$$(\text{S-N.Right}) \frac{E_2 \rightarrow E'_2}{n_1 + E_2 \rightarrow n_1 + E'_2}$$

$$(\text{S-Add}) \frac{-}{n_1 + n_2 \rightarrow n_3} n_3 = \text{add}(n_1, n_2)$$

$$(\text{S-Left}) \frac{E_1 \rightarrow_{\text{ch}} E'_1}{E_1 + E_2 \rightarrow_{\text{ch}} E'_1 + E_2}$$

$$(\text{S-Right}) \frac{E_2 \rightarrow_{\text{ch}} E'_2}{E_1 + E_2 \rightarrow_{\text{ch}} E_1 + E'_2}$$

$$(\text{op}+) \frac{-}{\langle n_1 + n_2, s \rangle \rightarrow \langle n, s \rangle} n = \text{add}(n_1, n_2)$$

$$(\text{op-geq}) \frac{-}{\langle n_1 \geq n_2, s \rangle \rightarrow \langle b, s \rangle} b = \text{geq}(n_1, n_2)$$

$$(\text{op1}) \frac{\langle e_1, s \rangle \rightarrow \langle e'_1, s' \rangle}{\langle e_1 + e_2, s \rangle \rightarrow \langle e'_1 + e_2, s' \rangle}$$

$$(\text{op2}) \frac{\langle e_2, s \rangle \rightarrow \langle e'_2, s' \rangle}{\langle v + e_2, s \rangle \rightarrow \langle v + e'_2, s' \rangle}$$

$$(\text{op1b}) \frac{\langle e_2, s \rangle \rightarrow \langle e'_2, s' \rangle}{\langle e_1 + e_2, s \rangle \rightarrow \langle e_1 + e'_2, s' \rangle}$$

$$(\text{op2b}) \frac{\langle e_1, s \rangle \rightarrow \langle e'_1, s' \rangle}{\langle e_1 + v, s \rangle \rightarrow \langle e'_1 + v', s' \rangle}$$

$$(\text{deref1}) \frac{-}{\langle !l, s \rangle \rightarrow \langle v, s \rangle} \text{if } l \in \text{dom}(s) \wedge s(l) = v$$

$$(\text{deref2}) \frac{\langle e, s \rangle \rightarrow \langle e', s' \rangle}{\langle !e, s \rangle \rightarrow \langle !e', s' \rangle}$$

$$(\text{ref1}) \frac{-}{\langle \text{ref } v, s \rangle \rightarrow \langle l, s[l \mapsto v] \rangle} l \notin \text{dom}(s)$$

$$(\text{ref2}) \frac{\langle e, s \rangle \rightarrow \langle e', s' \rangle}{\langle \text{ref } e, s \rangle \rightarrow \langle \text{ref } e', s \rangle}$$

$$(\text{assign1}) \frac{-}{\langle l := v, s \rangle \rightarrow \langle \text{skip}, s[l \mapsto v] \rangle} \text{if } l \in \text{dom}(s)$$

$$(\text{assign2}) \frac{\langle e, s \rangle \rightarrow \langle e', s' \rangle}{\langle l := e, s \rangle \rightarrow \langle l := e', s \rangle}$$

$$(\text{assign3}) \frac{\langle e_1, s \rangle \rightarrow \langle e'_1, s' \rangle}{\langle e_1 := e_2, s \rangle \rightarrow \langle e'_1 := e_2, s' \rangle}$$

$$(\text{if-tt}) \frac{-}{\langle \text{if true then } e_1 \text{ else } e_2, s \rangle \rightarrow \langle e_1, s \rangle}$$

$$(\text{if-ff}) \frac{-}{\langle \text{if false then } e_1 \text{ else } e_2, s \rangle \rightarrow \langle e_2, s \rangle}$$

$$(\text{if}) \frac{\langle e, s \rangle \rightarrow \langle e', s' \rangle}{\langle \text{if } e \text{ then } e_1 \text{ else } e_2, s \rangle \rightarrow \langle \text{if } e' \text{ then } e_1 \text{ else } e_2, s \rangle}$$

$$(\text{while}) \frac{-}{\langle \text{while } e \text{ do } e_1, s \rangle \rightarrow \langle \text{if } e \text{ then } (e_1; \text{while } e \text{ do } e_1) \text{ else skip}, s \rangle}$$

$$(\text{assign1b}) \frac{-}{\langle l := n, s \rangle \rightarrow \langle n, s[l \mapsto n] \rangle} l \in \text{dom}(s)$$

$$(\text{seq.skip}) \frac{-}{\langle \text{skip}; e_2, s \rangle \rightarrow \langle e_2, s \rangle}$$

$$(\text{seq}) \frac{\langle e_1, s \rangle \rightarrow \langle e'_1, s' \rangle}{\langle e_1; e_2, s \rangle \rightarrow \langle e'_1; e_2, s' \rangle}$$

$$(\text{seq.skipb}) \frac{-}{\langle v; e_2, s \rangle \rightarrow \langle e_2, s \rangle}$$

$$(\text{record1}) \frac{\langle e_i, s \rangle \rightarrow \langle e'_i, s' \rangle}{\langle \{\text{lab}_1 = v_1, \dots, \text{lab}_i = e_i, \dots, \text{lab}_k = e_k\}, s \rangle \rightarrow \langle \{\text{lab}_1 = v_1, \dots, \text{lab}_i = e'_i, \dots, \text{lab}_k = e_k\}, s' \rangle}$$

$$(\text{record2}) \frac{-}{\langle \#\text{lab}_i \{\text{lab}_1 = v_1, \dots, \text{lab}_i = e_i, \dots, \text{lab}_k = e_k\}, s \rangle \rightarrow \langle v_i, s \rangle} \quad (\text{record3}) \frac{\langle e, s \rangle \rightarrow \langle e', s' \rangle}{\langle \#\text{lab } e, s \rangle \rightarrow \langle \#\text{lab } e', s' \rangle}$$

$$(\text{par-L}) \frac{\langle e_1, s \rangle \rightarrow \langle e'_1, s' \rangle}{\langle e_1 \parallel e_2, s \rangle \rightarrow \langle e'_1 \parallel e_2, s' \rangle} \quad (\text{par-R}) \frac{\langle e_2, s \rangle \rightarrow \langle e'_2, s' \rangle}{\langle e_1 \parallel e_2, s \rangle \rightarrow \langle e_1 \parallel e'_2, s' \rangle} \quad (\text{end-L}) \frac{-}{\langle \text{skip} \parallel e, s \rangle \rightarrow \langle e, s \rangle} \quad (\text{end-R}) \frac{-}{\langle e \parallel \text{skip}, s \rangle \rightarrow \langle e, s \rangle}$$

$$(\text{await}) \frac{\langle e_1, s \rangle \rightarrow^* \langle \text{true}, s' \rangle \quad \langle e_2, s' \rangle \rightarrow^* \langle \text{skip}, s'' \rangle}{\langle \text{await } e_1 \text{ protect } e_2 \text{ end}, s \rangle \rightarrow \langle \text{skip}, s'' \rangle} \quad (\text{ChoiceL}) \frac{\langle e_1, s \rangle \rightarrow \langle e'_1, s' \rangle}{\langle e_1 \oplus e_2, s \rangle \rightarrow \langle e'_1, s' \rangle} \quad (\text{ChoiceR}) \frac{\langle e_2, s \rangle \rightarrow \langle e'_2, s' \rangle}{\langle e_1 \oplus e_2, s \rangle \rightarrow \langle e'_2, s' \rangle}$$

## Call-By-Value

$$(\text{CBV-app1}) \frac{\langle e_1, s \rangle \rightarrow \langle e'_1, s' \rangle}{\langle e_1 e_2, s \rangle \rightarrow \langle e'_1 e_2, s' \rangle} \quad (\text{CBV-app2}) \frac{\langle e_2, s \rangle \rightarrow \langle e'_2, s' \rangle}{\langle v e_2, s \rangle \rightarrow \langle v e'_2, s' \rangle} \quad (\text{CBV-fn}) \frac{-}{\langle (\text{fn } x: T \Rightarrow e) v, s \rangle \rightarrow \langle e\{v/x\}, s \rangle}$$

$$(\text{CBV-let1}) \frac{\langle e_1, s \rangle \rightarrow \langle e'_1, s' \rangle}{\langle \text{let } x: T = e_1 \text{ in } e_2, s \rangle \rightarrow \langle \text{let } x: T = e'_1 \text{ in } e_2, s' \rangle} \quad (\text{CBV-let2}) \frac{-}{\langle \text{let } x: T = v \text{ in } e_2, s \rangle \rightarrow \langle e_2\{v/x\}, s \rangle}$$

$$(\text{CBV-fix}) \frac{e \equiv \text{fn } f: T_1 \rightarrow T_2 \Rightarrow e_2}{\text{fix}.e \rightarrow e(\text{fn } x: T_1 \Rightarrow (\text{fix}.e) \ x)}$$

## Call-By-Name

$$(\text{CBN-app}) \frac{\langle e_1, s \rangle \rightarrow \langle e'_1, s' \rangle}{\langle e_1 e_2, s \rangle \rightarrow \langle e'_1 e_2, s' \rangle} \quad (\text{CBN-fn}) \frac{-}{\langle (\text{fn } x: T \Rightarrow e) e_2, s \rangle \rightarrow \langle e\{e_2/x\}, s \rangle}$$

$$(\text{CBN-let}) \frac{-}{\langle \text{let } x: T = e_1 \text{ in } e_2, s \rangle \rightarrow \langle e_2\{e_1/x\}, s \rangle}$$

$$(\text{CBN-fix}) \frac{-}{\text{fix}.e \rightarrow e(\text{fix}.e)}$$

## Grammatica dei tipi e type system

$T ::= \text{int} \mid \text{bool} \mid \text{unit} \mid T_1 \rightarrow T_2 \mid T_1 + T_2 \mid T_1 * T_2 \mid \text{ref } T \mid \{\text{lab}_1: T_1, \dots, \text{lab}_k: T_k\} \mid \text{proc}$   
 $T_{\text{loc}} ::= \text{ref } T$

### Tipi primitivi e operatori

$$\begin{array}{llll}
 (\text{int}) \frac{-}{\Gamma \vdash n: \text{int}} \text{ for } n \in \mathbb{Z} & (\text{bool}) \frac{-}{\Gamma \vdash b: \text{bool}} \text{ for } b \in \{\text{true}, \text{false}\} & (\text{op } +) \frac{\Gamma \vdash e_1: \text{int} \quad \Gamma \vdash e_2: \text{int}}{\Gamma \vdash e_1 + e_2: \text{int}} & (\text{op } *) \frac{\Gamma \vdash e_1: \text{int} \quad \Gamma \vdash e_2: \text{int}}{\Gamma \vdash e_1 * e_2: \text{int}} \\
 (\text{op or}) \frac{\Gamma \vdash e_1: \text{bool} \quad \Gamma \vdash e_2: \text{bool}}{\Gamma \vdash e_1 \text{ or } e_2: \text{bool}} & (\text{op and}) \frac{\Gamma \vdash e_1: \text{bool} \quad \Gamma \vdash e_2: \text{bool}}{\Gamma \vdash e_1 \text{ and } e_2: \text{bool}} & (\text{op } \geq) \frac{\Gamma \vdash e_1: \text{int} \quad \Gamma \vdash e_2: \text{int}}{\Gamma \vdash e_1 \geq e_2: \text{bool}} & (\text{skip}) \frac{-}{\Gamma \vdash \text{skip}: \text{unit}} \\
 (\text{seq}) \frac{\Gamma \vdash e_1: \text{unit} \quad \Gamma \vdash e_2: T}{\Gamma \vdash e_1; e_2: T} & (\text{if}) \frac{\Gamma \vdash e_1: \text{bool} \quad \Gamma \vdash e_2: T \quad \Gamma \vdash e_3: T}{\Gamma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3: T} & (\text{while}) \frac{\Gamma \vdash e_1: \text{bool} \quad \Gamma \vdash e_2: T}{\Gamma \vdash \text{while } e_1 \text{ do } e_2: T} & (\text{let}) \frac{\Gamma \vdash e_1: T \quad \Gamma, x: T \vdash e_2: T'}{\Gamma \vdash \text{let } x: T = e_1 \text{ in } e_2: T'} \\
 (\text{T-fix}) \frac{\Gamma \vdash e: (T_1 \rightarrow T_2) \rightarrow (T_1 \rightarrow T_2)}{\Gamma \vdash \text{fix}.e: T_1 \rightarrow T_2}
 \end{array}$$

### Referenze

$$\begin{array}{llll}
 (\text{ref}) \frac{\Gamma \vdash e: T}{\Gamma \vdash \text{ref } e: \text{ref } T} & (\text{deref}) \frac{\Gamma \vdash e: \text{ref } T}{\Gamma \vdash !e: T} & (\text{assign}) \frac{\Gamma \vdash e_1: \text{ref } T \quad \Gamma \vdash e_2: T}{\Gamma \vdash (e_1 := e_2): \text{unit}} & (\text{loc}) \frac{-}{\Gamma \vdash l: \text{ref } T} \Gamma(l) = \text{ref } T
 \end{array}$$

### Funzioni

$$\begin{array}{ll}
 (\text{var}) \frac{-}{\Gamma \vdash x: T} \text{ if } \Gamma(x) = T & (\text{fn}) \frac{\Gamma, x: T \vdash e: T'}{\Gamma \vdash (\text{fn } x: T \Rightarrow e): T \rightarrow T'} \\
 (\text{app}) \frac{\Gamma \vdash e_1: T \rightarrow T' \quad \Gamma \vdash e_2: T}{\Gamma \vdash e_1 e_2: T'}
 \end{array}$$

### Record

$$\begin{array}{ll}
 (\text{record}) \frac{\Gamma \vdash e_1: T_1 \dots \Gamma \vdash e_k: T_k}{\Gamma \vdash \{\text{lab}_1 = e_1, \dots, \text{lab}_k = e_k\}: \{\text{lab}_1: T_1, \dots, \text{lab}_k: T_k\}} & (\text{recordproj}) \frac{\Gamma \vdash e: \{\text{lab}_1: T_1, \dots, \text{lab}_k: T_k\}}{\Gamma \vdash \# \text{lab}_i e: T_i}
 \end{array}$$

### Concorrenza

$$\begin{array}{lll}
 (\text{T-sq1}) \frac{\Gamma \vdash e_1: \text{unit} \quad \Gamma \vdash e_2: \text{unit}}{\Gamma \vdash e_1; e_2: \text{unit}} & (\text{T-sq2}) \frac{\Gamma \vdash e_1: \text{proc} \quad \Gamma \vdash e_2: \text{proc}}{\Gamma \vdash e_1; e_2: \text{proc}} & (\text{T-par}) \frac{\Gamma \vdash e_1: T_1 \quad \Gamma \vdash e_2: T_2}{\Gamma \vdash e_1 \parallel e_2: \text{proc}} \quad T_1, T_2 \in \{\text{unit}, \text{proc}\} \\
 (\text{T-await}) \frac{\Gamma \vdash e_1: \text{bool} \quad \Gamma \vdash e_2: \text{unit}}{\Gamma \vdash \text{await } e_1 \text{ protect } e_2 \text{ end}: \text{unit}} & (\text{T-choice}) \frac{\Gamma \vdash e_1: \text{unit} \quad \Gamma \vdash e_2: \text{unit}}{\Gamma \vdash e_1 \oplus e_2: \text{unit}} &
 \end{array}$$

## Sottotipaggio

$$\text{(sub)} \frac{\Gamma \vdash e : T \quad T < : T'}{\Gamma \vdash e : T'} \quad \text{(s-refl)} \frac{-}{T < : T} \quad \text{(s-trans)} \frac{T < : T' \quad T' < : T''}{T < : T''}$$

## Sottotipaggio dei record

$$\text{(rec-perm)} \frac{\pi \text{ una permutazione di } 1, 2, \dots, k}{\{p_1 : T_1, \dots, p_k : T_k\} < : \{p_{\pi(1)} : T_{\pi(1)}, \dots, p_{\pi(k)} : T_{\pi(k)}\}} \quad \text{(rec-width)} \frac{-}{\{p_1 : T_1, \dots, p_k : T_k, p_{k+1} : T_{k+1}, \dots, p_z : T_z\} < : \{p_1 : T_1, \dots, p_k : T_k\}}$$

$$\text{(rec-depth)} \frac{T_1 < : T'_1 \dots T_k < : T'_k}{\{p_1 : T_1, \dots, p_k : T_k\} < : \{p_1 : T'_1, \dots, p_k : T'_k\}}$$

## Sottotipaggio delle funzioni

$$\text{(fun-sub)} \frac{T_1 : > T'_1 \quad T_2 < : T'_2}{T_1 \rightarrow T_2 < : T'_1 \rightarrow T'_2}$$

## Sottotipaggio somma e prodotto

$$\text{(prod-sub)} \frac{T_1 < : T'_1 \quad T_2 < : T'_2}{T_1 * T_2 < : T'_1 * T'_2} \quad \text{(sum-sub)} \frac{T_1 < : T'_1 \quad T_2 < : T'_2}{T_1 + T_2 < : T'_1 + T'_2}$$

# Progress

**Teorema 1** (Progress). *Sia  $e$  chiusa,  $\Gamma \vdash e : T$  e  $\text{dom}(\Gamma) \subseteq \text{dom}(s)$  allora  $e$  è un valore oppure  $\exists e', s' \text{ t.c. } \langle e, s \rangle \rightarrow \langle e', s' \rangle$ .*

**Proof 1.** Sia  $\phi$  una relazione ternaria definita come segue:  $\phi(\Gamma, e, T) \stackrel{\text{def}}{=} \forall s. \text{dom}(\Gamma) \subseteq \text{dom}(s) \Rightarrow \text{value}(e) \vee (\exists e', s' \text{ t.c. } \langle e, s \rangle \rightarrow \langle e', s' \rangle)$   
Proviamo che per ogni  $\Gamma, e, T$ , se  $\Gamma \vdash e : T$  allora  $\phi(\Gamma, e, T)$  usando *rule induction* su  $\Gamma \vdash e : T$ .

Provare la progress sul seguente linguaggio:

$E ::= x \mid n \mid E_1 + E_2 \mid E_1; E_2; \mid \text{fn } x : T \Rightarrow E \mid EE \mid \text{while } E \text{ do } E \mid E_1 : = E_2 \mid$

**Proof 2** (by *rule induction* su come ho tipato  $E$  ( $\Gamma \vdash E : T$ )).

**Caso Base 1** (int).

$$(\text{int}) \frac{}{\Gamma \vdash n : \text{int}} \text{ for } n \in \mathbb{Z}$$

Supponiamo di aver derivato  $\Gamma \vdash E : T$  utilizzando la regola di tipaggio **(int)**. Posso concludere che  $E \equiv n$ , per  $n \in \mathbb{Z}$ , e  $T \equiv \text{int}$ . Ne consegue che  $\phi(\Gamma, E, T) = \text{true}$  perché  $\text{value}(n) = \text{true}$ .

**Caso Base 2** (bool).

$$(\text{bool}) \frac{}{\Gamma \vdash b : \text{bool}} \text{ for } b \in \{\text{true}, \text{false}\}$$

Supponiamo di aver derivato  $\Gamma \vdash E : T$  utilizzando la regola di tipaggio **(bool)**.

Posso concludere che  $E \equiv b$ , per  $b \in \{\text{true}, \text{false}\}$ , e  $T \equiv \text{bool}$ .

Ne consegue che  $\phi(\Gamma, E, T) = \text{true}$  perché  $\text{value}(b) = \text{true}$ .

**Caso Base 3** (skip).

$$(\text{skip}) \frac{}{\Gamma \vdash \text{skip} : \text{unit}}$$

Supponiamo di aver derivato  $\Gamma \vdash E : T$  utilizzando la regola di tipaggio **(skip)**. Posso concludere che  $E \equiv \text{skip}$ ,  $T \equiv \text{unit}$ . Ne consegue che  $\phi(\Gamma, \text{skip}, \text{unit}) = \text{true}$  perché  $\text{value}(\text{skip}) = \text{true}$ .

**Caso Base 4** (var).

$$(\text{var}) \frac{}{\Gamma \vdash x : T} \text{ if } \Gamma(x) = T$$

Supponiamo di aver derivato  $\Gamma \vdash E : T$  utilizzando la regola di tipaggio **(var)**. Posso concludere che  $E \equiv x$ , e  $T \equiv T_1 \rightarrow T_2$  oppure  $T \equiv \text{int}$ . Posso dire che  $E$  non è chiusa perché  $x \in \text{FV}(E)$ . Dato che la premessa della dimostrazione è falsa posso dedurre che  $\phi(\Gamma, E, T) = \text{true}$ .

**Caso induttivo 1** (fn).

$$(\text{fn}) \frac{\Gamma, x : T \vdash e : T'}{\Gamma \vdash (\text{fn } x : T \Rightarrow e) : T \rightarrow T'}$$

Supponiamo di aver derivato  $\Gamma \vdash E : T$  utilizzando la regola di tipaggio **(fn)**. Posso concludere che  $E \equiv \text{fn } x : T \Rightarrow e$  e  $T \equiv T \rightarrow T'$ . Ne consegue che  $\phi(\Gamma, E, T) = \text{true}$  perché  $\text{value}(\text{fn } x : T \Rightarrow e) = \text{true}$ .

**Caso induttivo 2** (op +).

$$(\text{op } +) \frac{\Gamma \vdash e_1 : \text{int} \quad \Gamma \vdash e_2 : \text{int}}{\Gamma \vdash e_1 + e_2 : \text{int}}$$

Supponiamo di aver derivato  $\Gamma \vdash E : T$  utilizzando la regola di tipaggio **(op +)**. Posso concludere che  $E \equiv e_1 + e_2$ ,  $T \equiv \text{int}$ ,  $\Gamma \vdash e_1 : \text{int}$  e  $\Gamma \vdash e_2 : \text{int}$ . Per **ipotesi induttiva** ho che  $\phi(\Gamma, e_1, \text{int}) = \text{true}$  e  $\phi(\Gamma, e_2, \text{int}) = \text{true}$ . Perciò contraddistinguo tre casi:

1.

$$(\text{op1}) \frac{\langle e_1, s \rangle \rightarrow \langle e'_1, s' \rangle}{\langle e_1 + e_2, s \rangle \rightarrow \langle e'_1 + e_2, s' \rangle}$$

Se  $\text{value}(e_1) = \text{false}$  allora  $\langle e_1, s \rangle$  fa un passo e quindi, applicando la regola della semantica small-step **(op1)**, abbiamo che  $\phi(\Gamma, e_1 + e_2, \text{int}) = \text{true}$ ;

2.

$$(\text{op}+) \frac{}{\langle n_1 + n_2, s \rangle \rightarrow \langle n, s \rangle} n = \text{add}(n_1, n_2)$$

Se  $\text{value}(e_1) = \text{true} \wedge \text{value}(e_2) = \text{true}$  allora, applicando la regola della semantica small-step **(op +)**, abbiamo che  $\phi(\Gamma, e_1 + e_2, \text{int}) = \text{true}$ ;

3.

$$(\text{op2}) \frac{\langle e_2, s \rangle \rightarrow \langle e'_2, s' \rangle}{\langle v + e_2, s \rangle \rightarrow \langle v + e'_2, s' \rangle}$$

Se  $\text{value}(e_1) = \text{true} \wedge \text{value}(e_2) = \text{false}$  allora, applicando la regola della semantica small-step **(op2)**, abbiamo che  $\phi(\Gamma, e_1 + e_2, \text{int}) = \text{true}$ .

**Caso induttivo 3** (seq).

$$(\text{seq}) \frac{\Gamma \vdash e_1 : \text{unit} \quad \Gamma \vdash e_2 : T}{\Gamma \vdash e_1; e_2 : T}$$

Supponiamo di aver derivato  $\Gamma \vdash E : T$  utilizzando la regola di tipaggio **(seq)**. Posso concludere che  $E \equiv e_1; e_2$ ,  $T \equiv \text{unit}$ ,  $\Gamma \vdash e_1 : \text{unit}$  e  $\Gamma \vdash e_2 : \text{unit}$ . Per **ipotesi induttiva** ho che  $\phi(\Gamma, e_1, \text{unit}) = \text{true}$  e  $\phi(\Gamma, e_2, \text{unit}) = \text{true}$ . Perciò contraddistinguo due casi:

1.

$$(\mathbf{seq}) \frac{\langle e_1, s \rangle \rightarrow \langle e'_1, s \rangle}{\langle e_1; e_2, s \rangle \rightarrow \langle e'_1; e_2, s' \rangle}$$

Se  $\text{value}(e_1) = \text{false}$  allora  $\langle e_1, s \rangle$  fa un passo e quindi, applicando la regola della semantica small-step **(seq)**, abbiamo che  $\phi(\Gamma, e_1; e_2, \text{unit}) = \text{true}$ ;

2.

$$(\mathbf{seq.skip}) \frac{-}{\langle \text{skip}; e_2, s \rangle \rightarrow \langle e_2, s \rangle}$$

Se  $\text{value}(e_1) = \text{true}$  allora, applicando la regola della semantica small-step **(seq.skip)**, abbiamo che  $\phi(\Gamma, e_1; e_2, \text{unit}) = \text{true}$ .

**Caso induttivo 4 (while).**

$$(\mathbf{while}) \frac{\Gamma \vdash e_1 : \text{bool} \quad \Gamma \vdash e_2 : T}{\Gamma \vdash \text{while } e_1 \text{ do } e_2 : T}$$

Supponiamo di aver derivato  $\Gamma \vdash E : T$  utilizzando la regola di tipaggio **(while)**. Posso concludere che  $E \equiv \text{while } e_1 \text{ do } e_2$ , per qualche  $e_1, e_2$  tali che  $\Gamma \vdash e_1 : \text{bool}$  e  $\Gamma \vdash e_2 : T$ , e  $T \equiv \text{unit}$ . Ne consegue che  $\phi(\Gamma, E, T) = \text{true}$  perché applico la regola di riscrittura, applicando la regola semantica **(while)** e quindi faccio un passo:  $\langle e, s \rangle \equiv \langle \text{while } e_1 \text{ do } e_2 \rangle \rightarrow \langle \text{if } e \text{ then } (e_1; \text{while } e \text{ do } e_2) \text{ else skip} \rangle \equiv \langle e', s' \rangle$

# Type Preservation

**Teorema 1** (Type Preservation). *Sia  $e$  chiusa e  $\Gamma \vdash e : T$  e  $\text{dom}(\Gamma) \subseteq \text{dom}(s)$  e  $\langle e, s \rangle \rightarrow \langle e', s' \rangle$  allora  $\Gamma \vdash e' : T$  e  $e'$  è chiusa e  $\text{dom}(\Gamma) \subseteq \text{dom}(s')$ .*

Provare la type preservation sul seguente linguaggio:

$$E ::= x \mid n \mid E_1 + E_2 \mid \text{fn } x : T \Rightarrow E \mid EE$$

**Proof 1** (by rule induction su come ho derivato  $E \rightarrow F$ ).

**Caso Base 1** (S-Add).

$$\text{(S-Add)} \frac{-}{\langle n_1 + n_2, s \rangle \rightarrow \langle n_3, s \rangle} n_3 = \text{add}(n_1, n_2)$$

Supponiamo di aver derivato  $E \rightarrow F$  utilizzando la regola semantica **(S-Add)**. Ne consegue che  $E \equiv n_1 + n_2$  e  $F \equiv n_3$ . Se  $\Gamma \vdash E : \text{int}$  sarò stato in grado di tipare  $E$  utilizzando la regola di tipaggio **(op +)**. Di conseguenza ho tipato  $n_1$  e  $n_2$ :  $\Gamma \vdash n_1 : \text{int}$ ,  $\Gamma \vdash n_2 : \text{int}$ . Ne consegue che  $\Gamma \vdash n_3 : \text{int}$  come richiesto.

**Caso Base 2** (CBV-fn).

$$\text{(CBV-fn)} \frac{-}{\langle (\text{fn } x : T \Rightarrow e)v, s \rangle \rightarrow \langle e\{v/x\}, s \rangle}$$

Supponiamo di aver derivato  $E \rightarrow F$  utilizzando la regola semantica **(CBV-fn)**. Ne consegue che  $E \equiv (\text{fn } x : T \Rightarrow e)v$  e  $F \equiv e\{v/x\}$ . Se  $\Gamma \vdash E : T$  sarò stato in grado di tipare  $E$  utilizzando la regola di tipaggio **(app)**. In tal caso avrò dedotto che  $\Gamma \vdash v : T'$  e  $\Gamma \vdash \text{fn } x : T' \Rightarrow e : T \rightarrow T'$ . Di conseguenza sarò stato in grado di tipare il corpo di  $e$ :  $\Gamma, x : T' \vdash e : T$ . Dunque per il *substitution lemma* avrò che  $\Gamma \vdash e\{v/x\} \equiv F : T$  come richiesto.

**Caso induttivo 1** (S-Left).

$$\text{(S-Left)} \frac{E_1 \rightarrow E'_1}{E_1 + E_2 \rightarrow E'_1 + E_2}$$

Supponiamo di aver derivato  $E \rightarrow F$  utilizzando la regola semantica **(S-Left)**. Ne consegue che  $E \equiv E_1 + E_2$  e  $F \equiv E'_1 + E_2$ . Se  $\Gamma \vdash E : T$  allora vuol dire che sarò stato in grado di tipare  $E$  usando la regola di tipaggio **(op +)**. In tal caso avrò dedotto che  $T \equiv \text{int}$ ,  $\Gamma \vdash E_1 : \text{int}$ ,  $\Gamma \vdash E_2 : \text{int}$ . Per ipotesi induttiva da  $E_1 \rightarrow E'_1$  concludo che  $\Gamma \vdash E'_1 : \text{int}$ . Ma allora applicando la regola di tipaggio **(op +)** derivo che  $\Gamma \vdash E'_1 + E_2 \equiv F : \text{int}$  come richiesto.

**Caso induttivo 2** (S-Right).

$$\text{(S-Right)} \frac{E_2 \rightarrow E'_2}{E_1 + E_2 \rightarrow E_1 + E'_2}$$

Supponiamo di aver derivato  $E \rightarrow F$  utilizzando la regola semantica **(S-Right)**. Ne consegue che  $E \equiv E_1 + E_2$  e  $F \equiv E_1 + E'_2$ . Se  $\Gamma \vdash E : T$  allora vuol dire che sarò stato in grado di tipare  $E$  usando la regola di tipaggio **(op +)**. In tal caso avrò dedotto che  $T \equiv \text{int}$ ,  $\Gamma \vdash E_1 : \text{int}$ ,  $\Gamma \vdash E_2 : \text{int}$ . Per ipotesi induttiva da  $E_2 \rightarrow E'_2$  concludo che  $\Gamma \vdash E'_2 : \text{int}$ . Ma allora applicando la regola di tipaggio **(op +)** derivo che  $\Gamma \vdash E_1 + E'_2 \equiv F : \text{int}$  come richiesto.

**Caso induttivo 3** (CBV-app1).

$$\text{(CBV-app1)} \frac{\langle e_1, s \rangle \rightarrow \langle e'_1, s' \rangle}{\langle e_1 e_2, s \rangle \rightarrow \langle e'_1 e_2, s' \rangle}$$

Supponiamo di aver derivato  $E \rightarrow F$  utilizzando la regola semantica **(CBV-app1)**. Ne consegue che  $E \equiv E_1 E_2$  e  $F \equiv E'_1 E_2$  e  $E_1 \rightarrow E'_1$ . Se  $\Gamma \vdash E : T$  allora vuol dire che sarò stato in grado di tipare  $E$  usando la regola di tipaggio **(app)**. Quindi per ipotesi induttiva da  $E_1 \rightarrow E'_1$  deduco che  $\Gamma \vdash E'_1 : T_A \rightarrow T$ . Siccome  $F \equiv E'_1 E_2$  e  $\Gamma \vdash E_2 : T_A$ , ne consegue che, applicando la regola di tipaggio **(app)**, derivo che  $\Gamma \vdash E'_1 E_2 \equiv F : T$  come richiesto.

**Caso induttivo 4** (CBV-app2).

$$\text{(CBV-app2)} \frac{\langle e_2, s \rangle \rightarrow \langle e'_2, s' \rangle}{\langle v e_2, s \rangle \rightarrow \langle v e'_2, s' \rangle}$$

Supponiamo di aver derivato  $E \rightarrow F$  utilizzando la regola semantica **(CBV-app2)**. Ne consegue che  $E \equiv v E_2$  e  $F \equiv v E'_2$  e  $E_2 \rightarrow E'_2$ . Se  $\Gamma \vdash E : T$  allora vuol dire che sarò stato in grado di tipare  $E$  usando la regola di tipaggio **(app)**. Quindi per ipotesi induttiva da  $E_2 \rightarrow E'_2$  deduco che  $\Gamma \vdash E'_2 : T_A$ . Siccome  $F \equiv v E'_2$  e  $\Gamma \vdash v : T_A \rightarrow T$ , ne consegue che, applicando la regola di tipaggio **(app)**, derivo che  $\Gamma \vdash v E'_2 \equiv F : T$  come richiesto.