FR: Progress & Preservation

Stuck: $2 + \text{"}2\text{"}$

Progress: Typeable don't get stuck

λ-calc based: $\Gamma \vdash e : T$ then (i) e is a value (NF)

(ii) $\exists e'$ s.t. $e \to_\beta e'$

Preservation: Types are preserved by computation

$\Gamma \vdash e : T$ and $e \to_\beta e'$ then $\Gamma \vdash e' : T$

In FR we want:

- borrow safety (at most 1 mut borrow)
- store safety (no dangling ref)

FR Progress

Lemma 4.10

(i) S_1, valid store

(ii) t_1, term (expr), T type

(iii) Γ_1, well-formed typing env.

(iv) l, lifetime

(v) $S_1 \sim \Gamma_1$

If $\Gamma_1 \vdash \langle t_1 : T \rangle^{l} \dashv \Gamma_2$ then (i) t_1 is a value

(ii) $\exists\, t_2, S_2$ s.t. $\langle S_1 \triangleright t_1 \rightarrow S_2 \triangleright t_2 \rangle^{l}$

small-step

big-step

$$\exists t_2, S_2 . \langle S_1 \triangleright t \Downarrow S_2 \triangleright v \rangle^{\ell}$$

No aliasing

Def. 4.7 A store S is **valid** if it has no duplicate owned locations.

$$\nexists x, y \quad S(l_x) = l_a = S(l_y)$$

No dangling pointers

Def. 4.8 A typing env. Γ is **well-formed** w.r.t. ℓ

(i) $\Gamma(x) = \langle T \rangle^m$, contained$_\Gamma(T) = \&[\text{mut}]\, w$ then $\Gamma \vdash w : \langle T' \rangle^n$ then $n \geq m$ (x gets dropped before w)

(ii) $\Gamma(x) = \langle T \rangle^n$ then $n \geq \ell$ (ℓ is "inside" all existing lifetimes)

Def 4.4 store: S, p-value: $v^\perp$, p-type: $\tilde{T}$ — $S \vdash v^\perp \sim \tilde{T}$

$$\frac{}{S \vdash \varepsilon \sim \varepsilon} \qquad \frac{}{S \vdash c \sim \text{int}} \qquad \frac{}{S \vdash \perp \sim [T]}$$

$$\frac{loc(S, w) = \ell_a}{S \vdash \ell_a^0 \sim \&[mut]\, w} \qquad \frac{S(\ell_a) = \langle v^\perp \rangle^m \quad S \vdash v^\perp \sim \tilde{T}}{S \vdash \ell_a^\bullet \sim \square\, \tilde{T}}$$

Def 4.7 (safe abstraction) $\Gamma \sim S$ iff

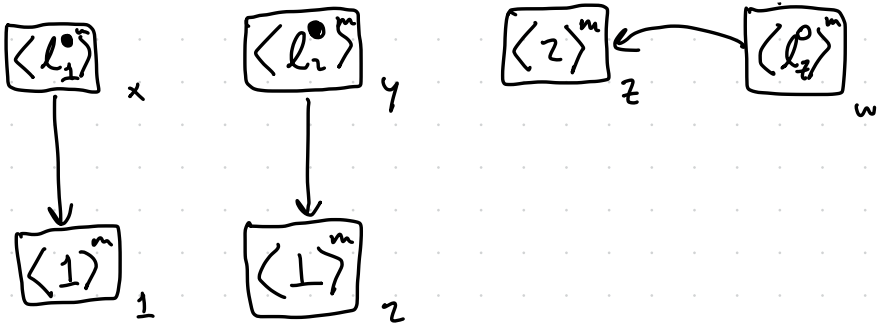
typing env. store

(i) $\Gamma(x) = \langle \tilde{T} \rangle^m \quad S(\ell_x) = \langle v^\perp \rangle^m \quad S \vdash v^\perp \sim \tilde{T}$

(ii) if $x \in dom(\Gamma)$ then $\ell_x \in dom(S)$

or. Γ and S have the same "named" information.

e.g. S



$\Gamma =$ $x \mapsto \langle \square\ int \rangle^m$

$y \mapsto \langle \square\ [\&z] \rangle^m$

$z \mapsto \langle int \rangle^m$

$w \mapsto \langle \&z \rangle^m$

S $\Gamma = \emptyset$

Preservation

FR Preservation (Lemma 4.11)

(i) S_1, valid state

(ii) t (term), T (type), l (lifetime)

(iii) Γ_1, w.f. typing env.

If $\Gamma_1 \vdash \langle t : T \rangle^{\ell} \dashv \Gamma_2$ and $\langle S_1 \triangleright t \Downarrow S_2 \triangleright v \rangle^{\ell}$

then S_2 is valid, Γ_2 is w.f. and $\Gamma_2 \sim S_2$

and $\boxed{S \vdash v \sim T}$

Note: if $S_1 \triangleright t_1 \longrightarrow S' \triangleright t'$ then it's not necessary that

$\Gamma_2 \sim S'$

ex. $\emptyset \vdash \{ \text{let mut } x = 0 \}^{m} : \varepsilon \dashv \emptyset$

$\emptyset \triangleright \{ \text{let mut } x = 0 \}^{m} \longrightarrow \{ x \mapsto 0 \} \triangleright \{ \varepsilon \}^{m}$

Type Safety

Thm 4.12

(i) S_1 : valid store

(ii) t (term), T (type), ℓ (lifetime)

(iii) Γ_1 : wf ctxt w.r.t. ℓ

if $\Gamma_1 \vdash \langle t : T \rangle^{\ell} \dashv \Gamma_2$ then $S_1 \triangleright t \to S_2 \triangleright v$

Def. 4.13

Γ is borrow safe if

(i) w.f. w.r.t. ℓ

(ii) $\nexists\, x, y$: $\overset{(i)}{\Gamma(x) = \langle T \rangle^{m}}$, $\overset{(ii)}{\Gamma(y) = \langle T' \rangle^{n}}$

(iii) $\text{contained}_\Gamma(T) = \&\,\text{mut}\ w$, (iv) $\text{contained}_\Gamma(T) = \&[\text{mut}]\ w'$

(v) $w \bowtie w'$

$\Gamma = \{\ x \mapsto \square\square\ \&\ *\,*\,\alpha_y\ ,\ z \mapsto \square\ \&\ \text{mut}\ y\ \}$ is

not borrow safe.

Corollary 4.14 (Borrow Safety)

(i) $S_1 \triangleright t_1$, $S_2 \triangleright t_2$

S_1: valid store; S_2: valid store

(ii) Γ_1, w.f. typing env. w.r.t ℓ where $\Gamma_1 \sim S_1$ and borrow safe

if $\Gamma_1 \vdash \langle t_1 : T_1 \rangle^{\ell}_{\sigma} \dashv \Gamma_2$ and $\langle S_1 \triangleright t_1 \to S_2 \triangleright t_2 \rangle^{\ell}$

then Γ_2 is w.f. and borrow safe.