FR Type / Borrow Checking (I)

Typing Judgement: $\underbrace{\Gamma}_{\text{context}} \vdash \underbrace{e}_{\text{expr}} : \underbrace{\tau}_{\text{types}}$.

Flow Sensitive Typing Judgements:

$$\underbrace{\Gamma_1}_{\text{before context}} \vdash \langle \underbrace{e}_{\text{expr}} : \underbrace{\tau}_{\text{type}} \rangle^{\ell}_{\sigma} \dashv \underbrace{\Gamma_2}_{\text{after context}}$$

lifetime

store typing

meaningless for implementing type checker

store typing:

let x = box 5 →

$$\boxed{\text{let } x = \ell_i^{\bullet}}$$

NOT a part of surface syntax

Types :

$$T ::= \overset{\text{unit}}{\varepsilon} \mid \text{int} \mid \&[\text{mut}]\ \underbrace{\vec{w}}_{\text{for this week} \rightarrow \cancel{\text{sequence}}\ \text{of}\ \underbrace{\text{l-values}}_{\text{variables + derefs}}} \mid \square T$$

variables + derefs

x , *x , **x , ...

ex.

$$\left.\begin{array}{l} \text{let mut } x : \text{int} = 2; \\ \text{let mut } y : \underbrace{\&\ \text{int}}_{\uparrow\ \text{replace } \&x \text{ in FR}} = \&x; \end{array}\right\}\ \text{rust}$$

⋮

ex:

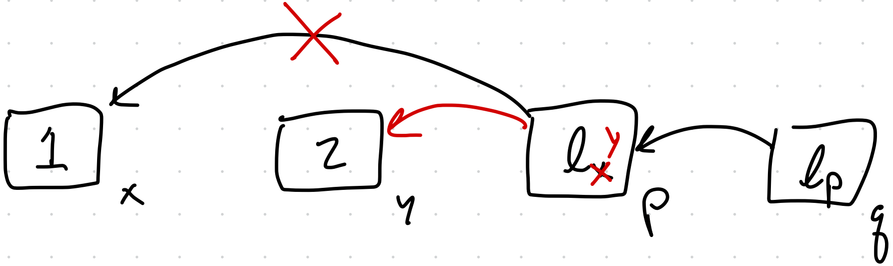
```
let x = 2;
let y = 3;
let z = if b { &mut x } else { &mut y };
```

$$z \overset{}{\underset{\wedge}{}}\ \&\text{mut } x, y$$

Strong vs. Weak Updates

ex.

```
let mut x = 1;
let mut y = 2;
let mut p = &mut x;
let mut q = &mut p;
*q = &mut y;
x + *p
```



p: &x, y by weak update

p: &y by strong update

Context: map from vars to slotted types

ex:

$$\{\ x \mapsto \underbrace{\boxed{\langle \underbrace{T}_{\text{type}} \rangle^{\overset{\text{lifetime}}{\ell}}}}_{\text{slot}},\ y \mapsto \langle T' \rangle^{m}\ \dots\ \}$$

Typing L-values (Definition 3.11)

side-condition

$$\frac{\boxed{\Gamma(x) = \langle T \rangle^{m}}}{\Gamma \vdash x : \langle T \rangle^{m}}\ (\text{var})$$

$$\frac{\Gamma \vdash w : \langle \Box T \rangle^{m}}{\Gamma \vdash *w : \langle T \rangle^{m}}\ (\text{box})$$

$$\frac{\Gamma \vdash w : \langle \&[\text{mut}]\ u \rangle^{n} \qquad \Gamma \vdash u : \langle T \rangle^{m}}{\Gamma \vdash *w : \langle T \rangle^{m}}\ (\text{Borrow})$$

not syntax directed

Typing Expression

constants:

$$\frac{}{\Gamma \vdash \langle \varepsilon : \varepsilon \rangle^{\ell} \dashv \Gamma}\,(\text{unit}) \qquad \frac{c \text{ is a num.}}{\Gamma \vdash \langle c : \text{int} \rangle^{\ell} \dashv \Gamma}\,(\text{int})$$

moves and copies:

$$\frac{\Gamma \vdash w : \langle T \rangle^{m} \qquad \text{copy}(T) \qquad \neg\,\text{readProhibited}(\Gamma, w)}{\Gamma \vdash \langle \hat{w} : T \rangle^{\ell} \dashv \Gamma}\,(\text{copy})$$

T is copyable, i.e., $T = \text{int}$ or $T = \& w$

$$\frac{\Gamma' \vdash w : \langle T \rangle^{m} \qquad \neg\,\text{writeProhibited}(\Gamma, w) \qquad \Gamma_2 = \text{move}(\Gamma, w)}{\Gamma_1 \vdash \langle w : T \rangle^{\ell} \dashv \Gamma_2}\,(\text{move})$$

Partial Type :

$$\tilde{T} = T \mid \Box \tilde{T} \mid \underbrace{\lfloor T \rfloor}_{\text{undefined}}$$

ex $\{ x \mapsto \Box \underbrace{\lfloor \mathrm{int} \rfloor}_{\text{moved out int}} \}$

Read / Write Prohibited

readProhibited (Γ, w)

writeProhibited (Γ, w)

Path (Def. 3.12) a sequence of dereferences, eg.

$\dagger \quad \dagger \quad \dagger \quad \dagger \quad \dagger$

Path Conflict (Def 3.14) $(w_1 \bowtie w_2)$ w_1 and w_2 conflict if

$w_1 = \dagger\ \dagger \cdots \dagger\ x$

$w_2 = \dagger\ \dagger \cdots \dagger\ \dagger\ \dagger\ x$

eg. $(x \bowtie x)$, $(x \bowtie \dagger x)$ $(x \bowtie \dagger \dagger x)$

Type Containment

$$\text{contains}(\cancel{\Gamma}, \tilde{T}, T) = \begin{cases} \text{contain}(\cancel{\Gamma}, \tilde{T}', T) & \tilde{T} = \square \tilde{T}' \\ \text{true} & \tilde{T} = T \\ \text{false} & \text{o.w.} \end{cases}$$

ex:

$$\text{contains}(\cancel{\Gamma}, \square\square\square\,\&x, \&x) = \text{true}$$

$$\text{contains}(\cancel{\Gamma}, \square\,\&\text{array}, \text{array}) = \text{true}$$

$$\text{contains}(\cancel{\Gamma}, \square\square\square\,[\square\,\text{int}], \square\,\text{int}) = \text{false}$$

$$\boxed{\text{readProhibited}(\Gamma, w)} = \exists x.\ \Gamma(x) = \langle T \rangle^{\ell}$$

$\underline{\text{and}}$ $\text{contains}(\cancel{\Gamma}, T, \&\text{mut } u)$ $\underline{\text{and}}$ $u \bowtie w$

~~[illegible]~~ $x \bowtie *x$

$$\boxed{\text{writeProhibited}(\Gamma, w)} = \text{readProhibited}(\Gamma, w) \ \underline{\text{or}}$$

$$\exists x.\ \Gamma(x) = \langle T \rangle^{\ell} \text{ and } \text{contains}(T, \&u) \text{ and } u \bowtie w$$

$\underline{\text{e.g.}}$

$$\Gamma = \{ x \mapsto \langle \overset{\text{contains}}{\boxed{\& * y}} \rangle^{\ell},\ y \mapsto \langle \Box\, \text{int} \rangle,\ z \mapsto \langle \Box \Box \overset{\text{contains}}{\boxed{\& y}} \rangle \}$$

$*y \bowtie ***y$

$y \bowtie ***y$

$\text{readProhibit}(\Gamma, ***y) = \text{false}$

$\text{writeProhibit}(\Gamma, ***y) = \text{true}$