CS383 Programming Languages

Quiz 7

1. What is the difference between a list and an array?

Array is mutable

2. What do you think are the major advantages of the imperative programming style over the functional style?

It’s easier to programmer.

3. Which rule is incorrect?

a. $$\frac{(M, e) \rightarrow (M', e')}{(M, \text{ref } e) \rightarrow (M', \text{ref } e')} \ (E-Ref)$$

b. $$\frac{l \notin \text{dom}(M)}{(M, \text{ref } v) \rightarrow ((M, l \mapsto v), l)} \ (E\text{-}RefV)$$

c. $$\frac{(M, e) \rightarrow (M', e')}{(M, !e) \rightarrow (M', !e')} \ (E-DeRef)$$

d. $$\frac{}{(M, !l) \rightarrow (M, l)} \ (E\text{-}DeRefLoc)$$

$$\frac{}{(M, !l) \rightarrow (M, M(l))} \ (E - DeRefLoc)$$

References (Operational Semantics, Cont'd)

$$\frac{(M, e) \to (M', e')}{(M, \text{ref } e) \to (M', \text{ref } e')} \text{ (E - Ref)} \qquad \frac{l \notin \text{dom}(M)}{(M, \text{ref } v) \to ((M, l \mapsto v), l)} \text{ (E - RefV)}$$

$$\frac{(M, e) \to (M', e')}{(M, !e) \to (M', !e')} \text{ (E - DeRef)} \qquad \frac{}{(M, !l) \to (M, M(l))} \text{ (E - DeRefLoc)}$$

$$\frac{(M, e_1) \to (M', e_1')}{(M, e_1 := e_2) \to (M', e_1' := e_2)} \text{ (E - Assign1)} \qquad \frac{(M, e_2) \to (M', e_2')}{(M, v_1 := e_2) \to (M', v_1 := e_2')} \text{ (E - Assign2)}$$

$$\frac{}{(M, l := v) \to (M[l \mapsto v], ())} \text{ (E - Assign)}$$

4.Which rule is incorrect?

a. $$\frac{(M, e_1) \to (M', e_1')}{(M, e_1 := e_2) \to (M', e_1' := e_2)} \text{ (E - Assign1)}$$

b. $$\frac{(M, e_2) \to (M', e_2')}{(M, v_1 := e_2) \to (M', v_1 := e_2')} \text{ (E - Assign2)}$$

c. $$\frac{}{(M, l := v) \to (M[l \mapsto v], v)} \text{ (E-Assign)}$$

d. None of the above

$$\frac{}{(M, l := v) \to (M[l \mapsto v], ())} \text{ (E - Assign)}$$

5. Which one is incorrect?

a. $$\frac{S;G \vdash e_1 : t \quad S;G \vdash e_2 : t}{S;G \vdash e_1 := e_2 : \text{unit}} \ (\text{T-Assign})$$

$$\frac{\Sigma;\Gamma \vdash e_1 : t\ \text{ref} \quad \Sigma;\Gamma \vdash e_2 : t}{\Sigma;\Gamma \vdash e_1 := e_2 : \text{unit}} \ (\text{T-Assign})$$

b. $$\frac{\Sigma;\Gamma \vdash e : t\ \text{ref}}{\Sigma;\Gamma \vdash !e : t} \ (\text{T-Deref})$$

c. $$\frac{\Sigma;\Gamma \vdash e : t}{\Sigma;\Gamma \vdash \text{ref}\ e : t\ \text{ref}} \ (\text{T-Ref})$$

d. $$\frac{\Sigma(l) = t}{\Sigma;\Gamma \vdash l : t\ \text{ref}} \ (\text{T-Loc})$$

References (Typing)

- We define the typing relation for memory store as Σ (or Si):

  Σ ::= . | Σ, l : t (t is the type of value stored at l)

- Our new typing judgment:

$$\Sigma; \Gamma \vdash e : t$$

- Types: t ::= .. | unit | t ref

$$\frac{}{\Sigma;\Gamma \mid\!- x : \Gamma(x)}\ (T\text{-}Var) \qquad \frac{\Sigma;\Gamma, x : t_1 \mid\!- e : t_2}{\Sigma;\Gamma \mid\!- \lambda x : t_1 . e\ :\ t_1 \to t_2}\ (T\text{-}Abs)$$

$$\frac{\Sigma;\Gamma \mid\!- e_1 : t_1 \to t_2 \quad \Sigma;\Gamma \mid\!- e_2 : t_1}{\Sigma;\Gamma \mid\!- e_1\ e_2 :\ t_2}\ (T\text{-}App) \qquad \frac{}{\Sigma;\Gamma \mid\!- () : unit}\ (T\text{-}Unit)$$

$$\frac{\Sigma(l) = t}{\Sigma;\Gamma \mid\!- l : t\ ref}\ (T\text{-}Loc) \qquad \frac{\Sigma;\Gamma \mid\!- e : t}{\Sigma;\Gamma \mid\!- ref\ e : t\ ref}\ (T\text{-}Ref)$$

$$\frac{\Sigma;\Gamma \mid\!- e : t\ ref}{\Sigma;\Gamma \mid\!- !e : t}\ (T\text{-}Deref) \qquad \frac{\Sigma;\Gamma \mid\!- e_1 : t\ ref \quad \Sigma;\Gamma \mid\!- e_2 : t}{\Sigma;\Gamma \mid\!- e_1 := e_2 : unit}\ (T\text{-}Assign)$$

6. In an expression e1; e2, if e1 doesn’t evaluate to (), what will happen?

Raise error(get stuck)

7. Which one is incorrect?

a. $$\frac{(M, e_1) \rightarrow (M', e_1')}{(M, e_1; e_2) \rightarrow (M', e_1'; e_2)} \ (E\text{-}Seq1)$$

b. $$\frac{}{(M, (); e) \rightarrow (M, e)} \ (E\text{-}Seq2)$$

c. $$\frac{\Sigma; \Gamma \mid\!- e_1 : unit \quad \Sigma; \Gamma \mid\!- e_2 : t}{\Sigma; \Gamma \mid\!- e_1; e_2 : t} \ (T\text{-}Var)$$

d. All of them are correct

8. Write down the evaluation steps of the following expression:

```
let y = ref 1 in
  let f = \x. x + !y in
    f (!y)
```

```
l 1, let y = l in
  let f = \x. x + !y in
    f (!y)
```

```
l 1, let f = \x. x + !l in
    f (!l)
```

```
l 1, \x.x+!l  (!l)
```

```
l 1, \x.x+!l  1
```

```
l 1, 1+!l
```

```
l 1, 1+1
```

```
l 1, 2
```

9. Which one is incorrect ?

a. $$\frac{}{\text{try v with e} \rightarrow \text{v}}\ (\text{E - TryV})$$

b. $$\frac{}{\text{try error with e} \rightarrow \text{error}}\ (\text{E-TryError})$$

$$\frac{}{\text{try error with e} \rightarrow \text{e}}\ (\text{E - TryError})$$

c. $$\frac{\text{e}_1 \rightarrow \text{e}_1'}{\text{try e}_1 \text{ with e}_2 \rightarrow \text{try e}_1' \text{ with e}_2}\ (\text{E - Try})$$

d. $$\frac{\Gamma \mid\!- e_1 : \text{t} \quad \Gamma \mid\!- \text{e}_2 : \text{t}}{\Gamma \mid\!- \text{try e}_1 \text{ with e}_2 : \text{t}}\ (\text{T - Try})$$

Handling Exception

- **Syntax:**

  e ::= …

  | try e_1 with e_2 (trap errors)

- **Evaluation:**

$$\frac{}{\text{try v with e} \rightarrow \text{v}}\ (\text{E - TryV}) \qquad \frac{}{\text{try error with e} \rightarrow \text{e}}\ (\text{E - TryError})$$

$$\frac{e_1 \rightarrow e_1'}{\text{try } e_1 \text{ with } e_2 \rightarrow \text{try } e_1' \text{ with } e_2}\ (\text{E - Try})$$

- **Typing:**

$$\frac{\Gamma \mid\!- e_1 : t \quad \Gamma \mid\!- e_2 : t}{\Gamma \mid\!- \text{try } e_1 \text{ with } e_2 : t}\ (\text{T - Try})$$

10. What is the problem with the following rule?

$$\frac{}{\Gamma \vdash \text{error} : t} \quad \text{(T-Error)}$$

t can be any type:

(\x:bool . x) error error: bool

(\x:bool . x) (error true) error: bool → bool

This breaks the uniqueness lemma

subtyping, or polymorphic types