Solution 7 - IMP

∗ If there is any problem, please contact TA.

Name:______________ Student ID:______________ Email: __________________

Problem 1. (30 points) Wouldn't it be simpler just to require the programmer to annotate error with its intended type in each context where it is used ? Why ?

Solution. Annotating error with its intended type would break the type preservation property. For example, the well-typed term

$$(\lambda x : Nat.x)\ ((\lambda y : Bool.5)\ (error\ as\ Bool));$$

(where error as T is the type-annotated syntax for exceptions) would evaluate in one step to an ill-typed term:

$$(\lambda x : Nat.x)\ \ (error\ as\ Bool);$$

As the evaluation rules propagate an error from the point where it occurs up to the top-level of a program, we may view it as having different types. The flexibility in the T-ERROR rule permits us to do this. □

Problem 2. (35 points) In lecture *Going Imperative*, the language is extended with while loop. In this problem, you are required to define the syntax and the semantics (including evaluation rules and typing rules) of while loop with `break` and `continue`

Solution.

(a) Syntax: (x and x_i are names)

$$e ::= \ ...\ |\ while\ e_1\ do\ e_2\ |\ break\ |\ continue$$

(Because we define the type of break and continue as unit, here we don't need to extend values and types)

(b) Semantics:

- Evaluation Rules (We introduce $< e_1, e_2 >$)

$$\frac{}{(M, while\ e_1\ do\ e_2) \to (M, if\ e_1\ then\ < e2, while\ e_1\ do\ e_2 >\ else\ ())} \quad \text{(E-while)}$$

$$\frac{(M, e_1) \to (M', e_1')}{(M, < e_1, e_2 >) \to (M', < e_1', e_2 >)} \quad \text{(E-whilePair)}$$

$$\frac{}{(M, < break, e >) \to (M, ())} \quad \text{(E-whilePairBreak)}$$

$$\frac{}{(M, < continue, e >) \to (M, e)} \quad \text{(E-whilePairContinue)}$$

$$\frac{}{(M, < break; e_1, e_2 >) \to (M, ())} \quad \text{(E-breakSeq)}$$

$$\frac{}{(M, < continue; e_1, e_2 >) \to (M, e_2)} \quad \text{(E-continueSeq)}$$

$$\frac{}{(M, < (), e_2 >) \to (M, e_2)} \quad \text{(E-whilePairUnit)}$$

- Typing Rules (We don't need to define typing rules for $< e_1, e_2 >$)

$$\frac{\Sigma; \Gamma \vdash e_1 : bool \quad \Sigma; \Gamma \vdash e_2 : unit}{\Sigma; \Gamma \vdash while\ e_1\ do\ e_2 : unit} \quad \text{(T-while)}$$

$$\frac{}{\Sigma; \Gamma \vdash break : unit} \quad \text{(T-break)}$$

$$\frac{}{\Sigma; \Gamma \vdash continue : unit} \quad \text{(T-continue)}$$

It's really hard to implement break and continue without introducing new statements like $< e_1, e_2 >$

□

Problem 3. (35 points)
Proof **Preservation Theorem**: If $\Sigma; \Gamma \vdash e : t, \Sigma; \Gamma \vdash M$, and $(M, e) \to (M', e')$, then for some $\Sigma' \supseteq \Sigma, \Sigma'; \Gamma \vdash e' : t, \Sigma'; \Gamma \vdash M'$. ($\Sigma' \supseteq \Sigma$ means Σ' agrees with Σ on all the old locations.)
Hint: You don't need to write "need to prove..." in this problem since in all cases it's quite similar. Also, you can use directly the following two lemma whose proofs are quite easy:

Lemma 1. ***Substitution:*** *If* $\Sigma; \Gamma, x : t_1 \vdash e : t_2$ *and* $\Sigma; \Gamma \vdash v : t_1$*, then* $\Sigma; \Gamma \vdash e[v/x] : t_2$ *(similar to the proof of previous substitution lemma)*

Lemma 2. *If* $\Sigma; \Gamma \vdash e : t$ *and* $\Sigma' \supseteq \Sigma$*, then* $\Sigma'; \Gamma \vdash e : t$ *(easy induction)*

Proof.

$$\frac{}{\Sigma;\Gamma \vdash x : \Gamma(x)} \quad (T-Var)$$
$$\frac{\Sigma;\Gamma x : t_1 \vdash e : t_2}{\Sigma;\Gamma \vdash \lambda x : t_1.e : t_1 \rightarrow t_2} \quad (T-Abs)$$
$$\frac{\Sigma;\Gamma \vdash e_1 : t_1 \rightarrow t_2 \quad \Sigma;\Gamma \vdash e_2 : t_1}{\Sigma;\Gamma \vdash e_1\ e_2 : t_2} \quad (T-App)$$
$$\frac{}{\Sigma;\Gamma \vdash () : unit} \quad (T-Unit)$$
$$\frac{\Sigma(l) = t}{\Sigma;\Gamma \vdash l : t\ ref} \quad (T-Loc)$$
$$\frac{\Sigma;\Gamma \vdash e : t}{\Sigma;\Gamma \vdash ref\ e : t\ ref} \quad (T-Ref)$$
$$\frac{\Sigma;\Gamma \vdash e : t\ ref}{\Sigma;\Gamma \vdash !e : t} \quad (T-DeRef)$$
$$\frac{\Sigma;\Gamma \vdash e_1 : t\ ref \quad \Sigma;\Gamma \vdash e_2 : t}{\Sigma;\Gamma \vdash e_1 := e_2 : unit} \quad (T-Assign)$$

(Here I don't list Boolean an Condition rules since they are not in the slides. Actually we should also proof these rules and the proof of these rules are similar.)
By induction on the derivation of $\Sigma;\Gamma \vdash e : t$

1. case $\dfrac{}{\Sigma;\Gamma \vdash x : \Gamma(x)}$

   Can't happen (There are no evaluations rules).

2. case $\dfrac{\Sigma;\Gamma x : t_1 \vdash e : t_2}{\Sigma;\Gamma \vdash \lambda x : t_1.e : t_1 \rightarrow t_2}$

   Can't happen.

3. case $\dfrac{\Sigma;\Gamma \vdash e_1 : t_1 \rightarrow t_2 \quad \Sigma;\Gamma \vdash e_2 : t_1}{\Sigma;\Gamma \vdash e_1\ e_2 : t_2}$

   - Subcase E-App1: $\dfrac{(M, e_1) \rightarrow (M', e_1')}{(M, (e_1\ e_2)) \rightarrow (M', e_1'\ e_2)}$

     (1) $\Sigma;\Gamma \vdash e_1 : t_1 \rightarrow t_2, (M, e_1) \rightarrow (M', e_1')$ (by assumption)
     (2) $\exists \Sigma' \supseteq \Sigma, \Sigma';\Gamma \vdash e_1' : t_1 \rightarrow t_2, \Sigma';\Gamma \vdash M'$ (by I.H.)
     (3) $\Sigma;\Gamma \vdash e_2 : t_1$ (by assumption)
     (4) $\Sigma';\Gamma \vdash e_2 : t_1$ (by (2), (3) and **Lemma 2**)
     (5) $\Sigma';\Gamma \vdash e_1'\ e_2 : t_2$ (by (2), (4) and T-App)

- Subcase E-App2: $\dfrac{(M, e_2) \to (M', e_2')}{(M, (v_1\ e_2)) \to (M', v_1'\ e_2')}$
  (1) $\Sigma; \Gamma \vdash e_2 : t_2, (M, e_2) \to (M', e_2')$ (by assumption)
  (2) $\exists \Sigma' \supseteq \Sigma, \Sigma'; \Gamma \vdash e_2' : t_2, \Sigma'; \Gamma \vdash M'$ (by I.H.)
  (3) $\Sigma; \Gamma \vdash v_1 : t_1 \to t_2$ (by assumption)
  (4) $\Sigma'; \Gamma \vdash v_1 : t_1 \to t_2$ (by (2), (3) and **Lemma 2**)
  (5) $\Sigma'; \Gamma \vdash v_1'\ e_2 : t_2$ (by (2), (4) and T-App)
- Subcase E-AppAbs: $\dfrac{}{(M, \lambda x : t_1.e_3\ v_2) \to (M, e_3[v_2/x])}$
  (1) $\Sigma; \Gamma \vdash \lambda x : t_1.e_3 : t_1 \to t_2$ (by assumption)
  (2) $\Sigma; \Gamma.x : t_1 \vdash e_3 : t_2$ (by inversion of T-Abs)
  (3) $\Sigma; \Gamma \vdash v_2 : t_1$ (by assumption)
  (4) $\Sigma; \Gamma \vdash e_3[v_2/x] : t_2$ (by (2), (3) and **Lemma 1**)
  (5) $\Sigma' = \Sigma$ satisfies.

4. case $\dfrac{}{\Sigma; \Gamma \vdash () : unit}$

   Can't happen.

5. case $\dfrac{\Sigma(l) = t}{\Sigma; \Gamma \vdash l : t\ ref}$

   Can't happen.

6. case $\dfrac{\Sigma; \Gamma \vdash e : t}{\Sigma; \Gamma \vdash ref\ e : t\ ref}$

   - Subcase E-RefV: $\dfrac{l \notin dom(M)}{(M, ref\ v) \to ((M, l \mapsto v), l)}$
     (1) Let $\Sigma' = \Sigma, l : t$
     (2) $\Sigma'(l) = t$ (by (1))
     (3) $\Sigma'; \Gamma \vdash l : t\ ref$ (by (2))
     (4) $\Sigma; \Gamma \vdash M$ (by I.H.)
     (5) $M'(l) = v$
     (6) $\Sigma; \Gamma \vdash v : t$ (by assumption and $\Sigma' \supseteq \Sigma$)
     (7) $\Sigma'; \Gamma \vdash M'$ (by (2), (4), (5), (6) and definition of $\Sigma; \Gamma \vdash M$)
   - Subcase E-Ref: $\dfrac{(M, e) \to (M', e')}{(M, ref\ e) \to (M', ref\ e')}$
     (1) $\Sigma; \Gamma \vdash e : t, (M, e) \to (M', e')$ (by assumption)
     (2) $\exists \Sigma', \Sigma'; \Gamma \vdash e' : t, \Sigma'; \Gamma \vdash M'$ (by (1) and I.H.)
     (3) $\Sigma'; \Gamma \vdash ref\ e' : t\ ref$ (by (2) and T-Ref)

7. case $\dfrac{\Sigma; \Gamma \vdash e : t\ ref}{\Sigma; \Gamma \vdash !e : t}$

- Subcase E-DeRefLoc: $\dfrac{}{(M, !l) \to (M, M(l))}$

  (1) Let $M(l) = v$ and $\Sigma' = \Sigma$

  Now we only need to prove $\Sigma; \Gamma \vdash v : t$

  (2) $\Sigma; \Gamma \vdash M$ (by I.H.)

  (3) $\Sigma; \Gamma \vdash !l : t$ (by assumption)

  (4) $\Sigma(l) = t$ (by (3))

  (5) $\Sigma; \Gamma \vdash v : t$ (by (1), (2), (4) and the definition of $\Sigma; \Gamma \vdash M$)

- Subcase E-DeRef: $\dfrac{(M, e) \to (M', e')}{(M, !e) \to (M', !e')}$

  (1) $\Sigma; \Gamma \vdash e : t\ ref$ and $(M, e) \to (M', e')$ (by assumption)

  (2) $\exists \Sigma' \supseteq \Sigma, \Sigma'; \Gamma \vdash e' : t\ ref$ and $\Sigma'; \Gamma \vdash M'$ (by (1) and I.H.)

  (3) $\Sigma'; \Gamma \vdash !e' : t$ (by (2) and T-DeRef)

8. case $\dfrac{\Sigma; \Gamma \vdash e_1 : t\ ref \quad \Sigma; \Gamma \vdash e_2 : t}{\Sigma; \Gamma \vdash e_1 := e_2 : unit}$

- Subcase E-Assign1: $\dfrac{(M, e_1) \to (M', e_1')}{(M, e_1 := e_2) \to (M', e_1' := e_2)}$

  (1) $\Sigma; \Gamma \vdash e_1 : t\ ref$, $\Sigma; \Gamma \vdash e_2 : t$ and $(M, e_1) \to (M', e_1')$ (by assumption)

  (2) $\exists \Sigma' \supseteq \Sigma, \Sigma'; \Gamma \vdash e_1 : t\ ref$ and $\Sigma'; \Gamma \vdash M'$ (by (1) and I.H.)

  (3) $\Sigma'; \Gamma \vdash e_1' := e_2 : unit$ (by (1), (2) and T-Assign)

- Subcase E-Assign2: $\dfrac{(M, e_2) \to (M', e_2')}{(M, v_1 := e_2) \to (M', v_1 := e_2')}$

  (1) $\Sigma; \Gamma \vdash v_1 : t\ ref$, $\Sigma; \Gamma \vdash e_2 : t$ and $(M, e_2) \to (M', e_2')$ (by assumption)

  (2) $\exists \Sigma' \supseteq \Sigma, \Sigma'; \Gamma \vdash e_2 : t$ and $\Sigma'; \Gamma \vdash M'$ (by (1) and I.H.)

  (3) $\Sigma'; \Gamma \vdash v_1 := e_2' : unit$ (by (1), (2) and T-Assign)

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

  (1) Let $\Sigma' = \Sigma, l : t$

  (2) $\Sigma'; \Gamma \vdash () : unit$ (by T-unit)

  (3) $\Sigma; \Gamma \vdash M$ (by I.H.)

  (5) $\Sigma'(l) = t$ and $M'(l) = v$

  (6) $\Sigma', \Gamma \vdash v : t$ (by assumption and **Lemma 2**)

  (7) $\Sigma'; \Gamma \vdash M'$ (by definition of $\Sigma; \Gamma \vdash M$)

$\square$