Programming Languages Quiz II **Last, First =** **Score =**

1. (15 points) Give a well-typed term whose evaluation all the way to a value (beginning in the empty store) will produce the following store when evaluation terminates.

$$\mu = (l_1 \mapsto \texttt{5}, \\ l_2 \mapsto l_1)$$

(Hint: you can use " `let x = ... in ...` ") . (5 points)

As *let* is a derived form (syntactic sugar), you can also represent the above term using lambda-calculus: (5 points)

What's the store typing Σ (such that $\emptyset|\Sigma \vdash \mu$)? (5 points)

2. (22 points) Given the above store μ and store typing Σ, now consider the following term `t`:

$$\texttt{ref (ref 0) := !}l_2$$

(a) What is the type of this term (i.e., $\emptyset|\Sigma \vdash \texttt{t : T}$)? (3 points)

(b) Give a typing derivation (i.e., draw the derivation tree). (7 points)

(c) Evaluate the above term step by step all the way to a value; in each step, fill in the store, store typing, and the computation rule used (note there is exactly one computation rule in each step). (12 points)

step	term	store	comp. rule
0	`ref (ref 0) := !`l_2	$(l_1 \mapsto 5, l_2 \mapsto l_1)$	N/A

3. (10 points) Is there a well-typed term whose evaluation (beginning in the empty store) will produce the following store when evaluation terminates?

$$\begin{aligned} \mu = (&l_1 \mapsto l_2, \\ &l_2 \mapsto l_3, \\ &l_3 \mapsto l_1) \end{aligned}$$

If so, give it. If not, explain briefly why no such term exists.

4. (26 points) Consider the following term:

$$\begin{array}{l} \texttt{case <l}_2\texttt{=ref 0> as T of <l}_1\texttt{=x}_1\texttt{>} \Rightarrow \texttt{x}_1 \\ \qquad\qquad\qquad\qquad\quad \texttt{<l}_2\texttt{=x}_2\texttt{>} \Rightarrow \texttt{!x}_2 \end{array}$$

(a) What T makes this term typecheck? Fill in the blanks: $\texttt{T = <l}_1\texttt{:}$ __________ $\texttt{, l}_2\texttt{:}$ __________ $\texttt{>}$. (6 points)

(b) What is the type of this term (given $\mu = \emptyset$ and $\Sigma = \emptyset$)? (3 points)

(c) Give the typing derivation (i.e., draw the derivation tree). (7 points)

(d) Evaluate this term all the way to a value. Distinguish between labels (l_i) and locations (l_i). (10 points)

step	term	store	comp. rule
0	$\texttt{case <l}_2\texttt{=ref 0> as T of <l}_1\texttt{=x}_1\texttt{>} \Rightarrow \texttt{x}_1$ $\texttt{<l}_2\texttt{=x}_2\texttt{>} \Rightarrow \texttt{!x}_2$	$\emptyset$	N/A

5. (15 points) Define the big-step semantics evaluation rule for `case` terms:

$$\texttt{case t of <l}_i\texttt{=x}_i\texttt{>}\Rightarrow\texttt{t}_i \;\; ^{i \in 1..n}$$

Do not forget the store (i.e., instead of $\texttt{t} \Downarrow \texttt{v}$, use $\texttt{t}|\mu \Downarrow \texttt{v}|\mu'$ to represent the evaluation sequence $\texttt{t}|\mu \to \texttt{t}_1|\mu_1 \to ... \to \texttt{v}|\mu'$).

6. (12 points) Forget about references and stores for now. We know that (preservation theorem says) $\texttt{t} \to \texttt{t}'$ and $\emptyset \vdash \texttt{t : T}$ implies $\emptyset \vdash \texttt{t}' \texttt{ : T}$. But is it also true that $\texttt{t} \to \texttt{t}'$ and $\emptyset \vdash \texttt{t}' \texttt{ : T}$ implies $\emptyset \vdash \texttt{t : T}$? If so, explain; otherwise, give an example. You can only use the syntax defined in the companion sheet (excluding reference creation, dereference, assignment, and store location).

Simply-typed lambda calculus with variants, references, $\texttt{Unit}$ and $\texttt{Nat}$

Syntax

$\texttt{t} ::=$		*terms*
	$\texttt{unit}$	*constant* $\texttt{unit}$
	$\texttt{x}$	*variable*
	$\lambda\texttt{x:T.t}$	*abstraction*
	$\texttt{t t}$	*application*
	$\texttt{ref t}$	*reference creation*
	$\texttt{!t}$	*dereference*
	$\texttt{t:=t}$	*assignment*
	l	*store location*
	$\texttt{0}$	*constant zero*
	$\texttt{succ t}$	*successor*
	$\texttt{<l=t> as T}$	*variant*
	$\texttt{case t of <l}_i\texttt{=x}_i\texttt{>}\Rightarrow\texttt{t}_i\ ^{i\in 1..n}$	*case*
$\texttt{v} ::=$		*values*
	$\texttt{unit}$	*constant* $\texttt{unit}$
	$\lambda\texttt{x:T.t}$	*abstraction value*
	l	*store location*
	$\texttt{nv}$	*numeric value*
	$\texttt{<l=v> as T}$	*variant value*
$\texttt{T} ::=$		*types*
	$\texttt{Unit}$	*unit type*
	$\texttt{T}\to\texttt{T}$	*type of functions*
	$\texttt{Ref T}$	*type of reference cells*
	$\texttt{Nat}$	*type of natural numbers*
	$\texttt{<l}_i\texttt{:T}_i\ ^{i\in 1..n}\texttt{>}$	*type of variants*
$\mu ::= \emptyset \mid \mu, l\mapsto \texttt{v}$		*stores*
$\Gamma ::= \emptyset \mid \Gamma, \texttt{x:T}$		*type environments*
$\Sigma ::= \emptyset \mid \Sigma, l\texttt{:T}$		*store typings*
$\texttt{nv} ::= \texttt{0} \mid \texttt{succ nv}$		*numeric values*

Evaluation $\boxed{\texttt{t}|\mu \longrightarrow \texttt{t}'|\mu'}$

$$\frac{\texttt{t}_1|\mu \longrightarrow \texttt{t}_1'|\mu'}{\texttt{t}_1\ \texttt{t}_2|\mu \longrightarrow \texttt{t}_1'\ \texttt{t}_2|\mu'} \qquad \text{(E-App1)}$$

$$\frac{\texttt{t}_2|\mu \longrightarrow \texttt{t}_2'|\mu'}{\texttt{v}_1\ \texttt{t}_2|\mu \longrightarrow \texttt{v}_1\ \texttt{t}_2'|\mu'} \qquad \text{(E-App2)}$$

$$(\lambda\texttt{x:T}_{11}\texttt{.t}_{12})\ \texttt{v}_2|\mu \longrightarrow [\texttt{x}\mapsto\texttt{v}_2]\texttt{t}_{12}|\mu \qquad \text{(E-AppAbs)}$$

$$\frac{l \notin dom(\mu)}{\texttt{ref v}_1|\mu \longrightarrow l|(\mu, l\mapsto\texttt{v}_1)} \qquad \text{(E-RefV)}$$

$$\frac{\texttt{t}_1|\mu \longrightarrow \texttt{t}_1'|\mu'}{\texttt{ref t}_1|\mu \longrightarrow \texttt{ref t}_1'|\mu'} \qquad \text{(E-Ref)}$$

$$\frac{\mu(l) = \texttt{v}}{\texttt{!}l|\mu \longrightarrow \texttt{v}|\mu} \qquad \text{(E-DerefLoc)}$$

$$\frac{\texttt{t}_1|\mu \longrightarrow \texttt{t}_1'|\mu'}{\texttt{!t}_1|\mu \longrightarrow \texttt{!t}_1'|\mu'} \qquad \text{(E-Deref)}$$

$$l\texttt{:=v}_2|\mu \longrightarrow \texttt{unit}|[l \mapsto \texttt{v}_2]\mu \qquad \text{(E-Assign)}$$

$$\frac{\texttt{t}_1|\mu \longrightarrow \texttt{t}_1'|\mu'}{\texttt{t}_1\texttt{:=t}_2|\mu \longrightarrow \texttt{t}_1'\texttt{:=t}_2|\mu'} \qquad \text{(E-Assign1)}$$

$$\frac{\texttt{t}_2|\mu \longrightarrow \texttt{t}_2'|\mu'}{\texttt{v}_1\texttt{:=t}_2|\mu \longrightarrow \texttt{v}_1\texttt{:=t}_2'|\mu'} \qquad \text{(E-Assign2)}$$

$$\frac{\texttt{t}_1|\mu \longrightarrow \texttt{t}_1'|\mu'}{\texttt{succ t}_1|\mu \longrightarrow \texttt{succ t}_1'|\mu'} \qquad \text{(E-Succ)}$$

$$\texttt{case (<l}_j\texttt{=v}_j\texttt{> as T) of <l}_i\texttt{=x}_i\texttt{>}\Rightarrow\texttt{t}_i \ ^{i\in 1..n} \longrightarrow [\texttt{x}_j \mapsto \texttt{v}_j]\texttt{t}_j \qquad \text{(E-CaseVariant)}$$

$$\frac{\texttt{t}_0 \longrightarrow \texttt{t}_0'}{\texttt{case t}_0\texttt{ of <l}_i\texttt{=x}_i\texttt{>}\Rightarrow\texttt{t}_i \ ^{i\in 1..n} \longrightarrow \texttt{case t}_0'\texttt{ of <l}_i\texttt{=x}_i\texttt{>}\Rightarrow\texttt{t}_i \ ^{i\in 1..n}} \qquad \text{(E-Case)}$$

$$\frac{\texttt{t}_i \longrightarrow \texttt{t}_i'}{\texttt{<l}_i\texttt{=t}_i\texttt{> as T} \longrightarrow \texttt{<l}_i\texttt{=t}_i'\texttt{> as T}} \qquad \text{(E-Variant)}$$

Typing $\boxed{\Gamma|\Sigma \vdash \texttt{t : T}}$

$$\Gamma|\Sigma \vdash \texttt{unit : Unit} \qquad \text{(T-Unit)}$$

$$\frac{\texttt{x:T} \in \Gamma}{\Gamma|\Sigma \vdash \texttt{x : T}} \qquad \text{(T-Var)}$$

$$\frac{\Gamma, \texttt{x:T}_1|\Sigma \vdash \texttt{t}_2 \texttt{ : T}_2}{\Gamma|\Sigma \vdash \lambda\texttt{x:T}_1\texttt{.t}_2 \texttt{ : T}_1\texttt{→T}_2} \qquad \text{(T-Abs)}$$

$$\frac{\Gamma|\Sigma \vdash \texttt{t}_1 \texttt{ : T}_{11}\texttt{→T}_{12} \qquad \Gamma|\Sigma \vdash \texttt{t}_2 \texttt{ : T}_{11}}{\Gamma|\Sigma \vdash \texttt{t}_1\texttt{ t}_2 \texttt{ : T}_{12}} \qquad \text{(T-App)}$$

$$\frac{\Sigma(l) = \texttt{T}_1}{\Gamma|\Sigma \vdash l \texttt{ : Ref T}_1} \qquad \text{(T-Loc)}$$

$$\frac{\Gamma|\Sigma \vdash \texttt{t}_1 \texttt{ : T}_1}{\Gamma|\Sigma \vdash \texttt{ref t}_1 \texttt{ : Ref T}_1} \qquad \text{(T-Ref)}$$

$$\frac{\Gamma|\Sigma \vdash \texttt{t}_1 \texttt{ : Ref T}_{11}}{\Gamma|\Sigma \vdash \texttt{!t}_1 \texttt{ : T}_{11}} \qquad \text{(T-Deref)}$$

$$\frac{\Gamma|\Sigma \vdash \texttt{t}_1 \texttt{ : Ref T}_{11} \qquad \Gamma|\Sigma \vdash \texttt{t}_2 \texttt{ : T}_{11}}{\Gamma|\Sigma \vdash \texttt{t}_1\texttt{:=t}_2 \texttt{ : Unit}} \qquad \text{(T-Assign)}$$

$$\frac{\Gamma|\Sigma \vdash \texttt{t}_1 \texttt{ : Nat}}{\Gamma|\Sigma \vdash \texttt{succ t}_1 \texttt{ : Nat}} \qquad \text{(T-Succ)}$$

$$\frac{\Gamma \vdash \texttt{t}_j \texttt{ : T}_j}{\Gamma \vdash \texttt{<l}_j\texttt{=t}_j\texttt{> as <l}_i\texttt{:T}_i \ ^{i\in 1..n}\texttt{> : <l}_i\texttt{:T}_i \ ^{i\in 1..n}\texttt{>}} \qquad \text{(T-Variant)}$$

$$\frac{\begin{array}{c}\Gamma \vdash \texttt{t}_0 \texttt{ : <l}_i\texttt{:T}_i \ ^{i\in 1..n}\texttt{>} \\ \text{for each } i \quad \Gamma, \texttt{x}_i\texttt{:T}_i \vdash \texttt{t}_i \texttt{ : T}\end{array}}{\Gamma \vdash \texttt{case t}_0\texttt{ of <l}_i\texttt{=x}_i\texttt{>}\Rightarrow\texttt{t}_i \ ^{i\in 1..n} \texttt{ : T}} \qquad \text{(T-Case)}$$