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

1. (8 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.

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

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

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

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

2. (20 points) Given the above store μ and store typing Σ, now consider the following term `t`:

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

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

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

(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).

step	term	store	comp. rule
0	`ref (ref false) := !`l_2	$(l_1 \mapsto \texttt{true}, l_2 \mapsto l_1)$	N/A

3. (8 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. (12 points) Consider the following term:

```
case <l₂=ref 0> as T of <l₁=x₁> ⇒ x₁
                        <l₂=x₂> ⇒ iszero (!x₂)
```

(a) What `T` makes this term typecheck? Fill in the blanks: `T = <l`$_1$`:` ______ `, l`$_2$`:` ______ `>`.

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

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

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

step	term	store	comp. rule
0	`case <l₂=ref 0> as T of <l₁=x₁> ⇒ x₁` `<l₂=x₂> ⇒ iszero (!x₂)`	$\emptyset$	N/A

5. (10 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 $\texttt{t}|\mu \rightarrow \texttt{t}_1|\mu_1 \rightarrow \ldots \rightarrow \texttt{v}|\mu'$).

6. (10 points) Forget about references and stores for now. We know that (preservation theorem says) $\texttt{t} \rightarrow \texttt{t}'$ and $\emptyset \vdash \texttt{t : T}$ implies $\emptyset \vdash \texttt{t' : T}$. But is it also true that $\texttt{t} \rightarrow \texttt{t}'$ and $\emptyset \vdash \texttt{t' : T}$ 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); for example, you can **not** use "if ... then ... else" which is **not** part of the syntax in the companion sheet.

Simply-typed lambda calculus with variants, references, Unit, Bool and Nat

Syntax

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

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

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

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

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

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

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

$$\frac{\mu(l) = \mathtt{v}}{\mathtt{!}l|\mu \longrightarrow \mathtt{v}|\mu} \tag{E-DerefLoc}$$

$$\frac{\mathtt{t}_1|\mu \longrightarrow \mathtt{t}_1'|\mu'}{\mathtt{!t}_1|\mu \longrightarrow \mathtt{!t}_1'|\mu'} \tag{E-Deref}$$

$$l\mathtt{:=v}_2|\mu \longrightarrow \mathtt{unit}|[l \mapsto \mathtt{v}_2]\mu \tag{E-Assign}$$

$$\frac{\mathtt{t}_1|\mu \longrightarrow \mathtt{t}_1'|\mu'}{\mathtt{t}_1\mathtt{:=t}_2|\mu \longrightarrow \mathtt{t}_1'\mathtt{:=t}_2|\mu'} \tag{E-Assign1}$$

$$\frac{\mathtt{t}_2|\mu \longrightarrow \mathtt{t}_2'|\mu'}{\mathtt{v}_1\mathtt{:=t}_2|\mu \longrightarrow \mathtt{v}_1\mathtt{:=t}_2'|\mu'} \tag{E-Assign2}$$

$$\frac{\mathtt{t}_1|\mu \longrightarrow \mathtt{t}_1'|\mu'}{\mathtt{succ\ t}_1|\mu \longrightarrow \mathtt{succ\ t}_1'|\mu'} \tag{E-Succ}$$

$$\mathtt{iszero\ 0}|\mu \longrightarrow \mathtt{true}|\mu \tag{E-IszeroZero}$$

$$\mathtt{iszero\ (succ\ nv}_1\mathtt{)}|\mu \longrightarrow \mathtt{false}|\mu \tag{E-IszeroSucc}$$

$$\frac{\mathtt{t}_1|\mu \longrightarrow \mathtt{t}_1'|\mu'}{\mathtt{iszero\ t}_1|\mu \longrightarrow \mathtt{iszero\ t}_1'|\mu'} \tag{E-IsZero}$$

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

$$\frac{\mathtt{t}_0 \longrightarrow \mathtt{t}_0'}{\mathtt{case\ t}_0\mathtt{\ of\ <l}_i\mathtt{=x}_i\mathtt{>}\Rightarrow\mathtt{t}_i\ {}^{i\in 1..n} \longrightarrow \mathtt{case\ t}_0'\mathtt{\ of\ <l}_i\mathtt{=x}_i\mathtt{>}\Rightarrow\mathtt{t}_i\ {}^{i\in 1..n}} \tag{E-Case}$$

$$\frac{\mathtt{t}_i \longrightarrow \mathtt{t}_i'}{\mathtt{<l}_i\mathtt{=t}_i\mathtt{>\ as\ T} \longrightarrow \mathtt{<l}_i\mathtt{=t}_i'\mathtt{>\ as\ T}} \tag{E-Variant}$$

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

$$\Gamma|\Sigma \vdash \mathtt{unit} : \mathtt{Unit} \tag{T-Unit}$$

$$\frac{\mathtt{x:T} \in \Gamma}{\Gamma|\Sigma \vdash \mathtt{x} : \mathtt{T}} \tag{T-Var}$$

$$\frac{\Gamma, \mathtt{x:T}_1|\Sigma \vdash \mathtt{t}_2 : \mathtt{T}_2}{\Gamma|\Sigma \vdash \lambda\mathtt{x:T}_1\mathtt{.t}_2 : \mathtt{T}_1\to\mathtt{T}_2} \tag{T-Abs}$$

$$\frac{\Gamma|\Sigma \vdash \mathtt{t}_1 : \mathtt{T}_{11}\to\mathtt{T}_{12} \qquad \Gamma|\Sigma \vdash \mathtt{t}_2 : \mathtt{T}_{11}}{\Gamma|\Sigma \vdash \mathtt{t}_1\ \mathtt{t}_2 : \mathtt{T}_{12}} \tag{T-App}$$

$$\frac{\Sigma(l) = \mathtt{T}_1}{\Gamma|\Sigma \vdash l : \mathtt{Ref\ T}_1} \tag{T-Loc}$$

$$\frac{\Gamma|\Sigma \vdash \mathtt{t}_1 : \mathtt{T}_1}{\Gamma|\Sigma \vdash \mathtt{ref\ t}_1 : \mathtt{Ref\ T}_1} \tag{T-Ref}$$

$$\frac{\Gamma|\Sigma \vdash \mathtt{t}_1 : \mathtt{Ref\ T}_{11}}{\Gamma|\Sigma \vdash \mathtt{!t}_1 : \mathtt{T}_{11}} \tag{T-Deref}$$

$$\frac{\Gamma|\Sigma \vdash \mathtt{t}_1 : \mathtt{Ref\ T}_{11} \qquad \Gamma|\Sigma \vdash \mathtt{t}_2 : \mathtt{T}_{11}}{\Gamma|\Sigma \vdash \mathtt{t}_1\mathtt{:=t}_2 : \mathtt{Unit}} \tag{T-Assign}$$

$$\Gamma|\Sigma \vdash \texttt{true} : \texttt{Bool} \qquad \text{(T-True)}$$

$$\Gamma|\Sigma \vdash \texttt{false} : \texttt{Bool} \qquad \text{(T-False)}$$

$$\Gamma|\Sigma \vdash \texttt{0} : \texttt{Nat} \qquad \text{(T-Zero)}$$

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

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

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