Mid-term Exam

Type Systems

December 20, 2006

Last Name : _____

First Name : _____

Section : _____

Exercise	Points	Achieved Points
1	10	
2	10	
3	10	
Total	30	

Exercice 1 : Normal Forms (10 points)

Let $\mathcal{V}$ be the set of variables and Λ the set of λ-terms.

Let $\mathcal{N} \subset \Lambda$ be the set of normal forms ($\mathcal{N} = \{t \in \Lambda \mid \nexists t' \in \Lambda : t \to t'\}$).
We define inductively the subset $\mathcal{N}'$ of Λ:

$$(\textsc{Var})\ \frac{x \in \mathcal{V}}{x \in \mathcal{N}'} \quad (\textsc{Abs})\ \frac{x \in \mathcal{V} \quad t \in \mathcal{N}'}{\lambda x.t \in \mathcal{N}'} \quad (\textsc{App}_n)\ \frac{x \in \mathcal{V} \quad t_1 \in \mathcal{N}' \quad \cdots \quad t_n \in \mathcal{N}'}{x\ t_1 \cdots t_n \in \mathcal{N}'}\ n \in \mathbb{N}, n > 0$$

Show that $\mathcal{N}' = \mathcal{N}$.

Hint: Show that if $t \in \mathcal{N}$ then $t \in \mathcal{N}'$ by induction on $t \in \Lambda$ and that if $t \in \mathcal{N}'$ then $t \in \mathcal{N}$ by induction on the derivation $t \in \mathcal{N}'$.

Exercice 2 : Typed Arithmetic Expressions (10 points)

We first recall the syntax for arithmetic expressions (TAPL, p.91):

t	::=		**terms :**	v	::=		**values :**
	$\mid$	$true$	*constant true*		$\mid$	$true$	*true value*
	$\mid$	$false$	*constant false*		$\mid$	$false$	*false value*
	$\mid$	$if\ t\ then\ t\ else\ t$	*condition*		$\mid$	nv	*numeric value*
	$\mid$	0	*constant zero*				
	$\mid$	$succ\ t$	*successor*	nv	::=		**numeric values :**
	$\mid$	$pred\ t$	*predecessor*		$\mid$	0	*zero value*
	$\mid$	$iszero\ t$	*zero test*		$\mid$	$succ\ nv$	*successor value*

and the evaluation rules for numbers (TAPL, p.41):

$$\text{(E-Succ)}\ \frac{t_1 \longrightarrow t_1'}{succ\ t_1 \longrightarrow succ\ t_1'}$$

$$\text{(E-PredZero)}\ pred\ 0 \longrightarrow 0 \qquad \text{(E-IsZeroZero)}\ iszero\ 0 \longrightarrow true$$

$$\text{(E-PredSucc)}\ pred\ (succ\ nv_1) \longrightarrow nv_1 \qquad \text{(E-IsZeroSucc)}\ iszero\ (succ\ nv_1) \longrightarrow false$$

$$\text{(E-Pred)}\ \frac{t_1 \longrightarrow t_1'}{pred\ t_1 \longrightarrow pred\ t_1'} \qquad \text{(E-IsZero)}\ \frac{t_1 \longrightarrow t_1'}{iszero\ t_1 \longrightarrow iszero\ t_1'}$$

Suppose we remove the E-PredZero rule.

Does progress still hold ? What about preservation ?

Change the definition of values in the modified language such that both progress and preservation hold. However, you are not allowed to reintroduce E-PredZero or to add another reduction rule for terms of the form $pred(x)$.

Exercice 3 : Option Types (10 points)

We extend the syntax for the simply typed λ-calculus (TAPL, p.103) with option types in a similar way as in Scala, e.g.

```
(\o: option Nat. o match {
  case some x => x
  case none   => 0
}) some (succ 0)
```

The meaning of the above is that when o is an instance of *some*, the first branch is selected and x takes the value carried inside o. When o is an instance of *none*, the second branch is evaluated. As a rule of thumb, the evaluation rules should match Scala's behavior, like call-by-value evaluation order and the usual meaning for *match*. The language should also allow you to create values of both kinds.

Formalize this extension. Your solution should include a grammar extension (for terms, values and types), evaluation rules and typing rules for the new terms. Typing rules should peserve type safety and make it impossible to find two different types for the same term (uniqueness of types). (There is no need to prove these properties).