CS 520 Problem Set 8

Yarom Gabay

Nov 15, 2004

Problem 1, Exercise 18.13.1 [Pierce]

The solution adds a $Ref\ Nat$ field to the object representation, and relies on the solution we had (one of the previous assignments) to identify same references.
We add a field to the object representation,
$IdCounterRep = \{..., id : \ Ref\ Nat\}$

We add two methods $setId$ and $getId$. These method Set and Get the id field.
The implementation of $SameObject$ will be as followed,(implementation relies on $equal$ function)

$$\begin{array}{ll} SameObject = & \lambda\ o1 : idCounterClass. \\ & \lambda\ o2 : idCounterClass. \\ & o1.setId\ 0; \\ & o2.setId\ 1; \\ & if\ (equal\ (o1.getId\ unit)\ (o2.getId\ unit)) \\ & then\ true \\ & else\ false \end{array}$$

Problem 2

Part 1

In $SetCounterClass2$ we try to achieve recursion by just calling $SetCounterClass2$ again. But we know that we can only achieve recursion in our language by either using fix or references. More specifically, the inner call to $SetCounterClass2$ is not bound to the implementation of $SetCounterClass2$.

Part 2

The expected behavior here is an open recursion behavior but that's not what will happen. The behavior we want is than when the inc method will be called, the superclass's inc will be called and since it is using the set and get methods, the subclass's set and get methods will actually be invoked.

The problem is that we don't pass the subclass's $self$ to the superclass, and by not doing so the superclass has it's own $self$ and it's own set and get will be invoked.

Part 3

The implementation of inc in $instrCounterClass2$ is now identical to the inc implementation in the superclass. The difference is that they both use $self$ which is bound to different records (of methods). The superclass's $self$ is bound to the method record of $SetCounterClass$ while the subclass's $self$ is bound to $instrClassClass2$.
While invoking $instrCounterClass1$.inc causes $SetCounterClass$'s set and get to be called, invoking $instrCounterClass2$.inc causes $instrCounterClass2$'s set and get to be called.

Problem 3

Part 1

$$\Gamma \vdash 0 : Nat \mid \{\} \ \textbf{(CT-Zero)}$$

$$\frac{\Gamma \vdash t : T \mid C}{\Gamma \vdash succ\ t : Nat \mid C \cup \{T = Nat\}} \ \textbf{(CT-Succ)}$$

$$\frac{\Gamma \vdash t : T \mid C}{\Gamma \vdash pred\ t : Nat \mid C \cup \{T = Nat\}} \ \textbf{(CT-Pred)}$$

$$\frac{\Gamma \vdash t : T \mid C}{\Gamma \vdash iszero\ t : Bool \mid C \cup \{T = Nat\}} \ \textbf{(CT-IsZero)}$$

$$\Gamma \vdash true : Bool \mid \{\} \ \textbf{(CT-True)}$$

$$\Gamma \vdash false : Bool \mid \{\} \ \textbf{(CT-False)}$$

$$\frac{\Gamma \vdash t_1 : T_1 | C_1 \quad \Gamma \vdash t_2 : T_2 | C_2 \quad \Gamma \vdash t_3 : T_3 | C_3}{\Gamma \vdash if\ t_1\ then\ t_2\ else\ t_3 : T_2 \mid C_1 \cup C_2 \cup C_3 \cup \{T_1 = Bool, T_2 = T_3\}} \ \textbf{(CT-If)}$$

Part 2

(Algorithm in HD19 p. 8 (Continued))

```
IF t = 0
THEN return (Nat, {})
ELSE
IF t = succ t1
THEN LET constraints(G,t1) = (T1,C1)
```

```
     IN return (Nat, C1 union {T1 = Nat})
ELSE
IF t = pred t1
THEN LET constraints(G,t1) = (T1,C1)
     IN return (Nat, C1 union {T1 = Nat})
ELSE
IF t = iszero t1
THEN LET constraints(G,t1) = (T1,C1)
     IN return (Bool, C1 union {T1 = Nat})
ELSE
IF t = true
THEN return (Bool, {})
ELSE
IF t = false
THEN return (Bool, {})
ELSE
IF t = if t1 then t2 else t3
THEN LET constraints(G,t1) = (T1,C1) and
         constraints(G,t2) = (T2,C2) and
         constraints(G,t3) = (T3,C3)
     IN return (T2, C1 union C2 union C3 union {T1 = Bool, T2 = T3})
ELSE 'error'
```

Problem 4, Exercise 22.3.11 [Pierce]

$$\frac{\Gamma \vdash t : T \mid C \quad X\ fresh}{\Gamma \vdash fix\ t : X \mid C \cup \{T = X \to X\}}\ \textbf{(CT-Fix)}$$

Add the following case to the algorithm in HD19 p. 8,

```
IF t = fix t1
THEN LET constraints(G,t1) = (T1,C1) and
         X = fresh type variable
     IN return (X, C1 union {T1 = X -> X})
ELSE
```

Problem 5

```
(lambda f:A. lambda x:B. f (f x)) (lambda y:C. succ y)
```

Constraints Set: $\{C = Nat, A = B \to X_1, A = X_1 \to X_2, A \to B \to X_2 = (C \to Nat) \to X_3\}$
Principal type: $Nat \to Nat$

Problem 6

```
(lambda f:A. lambda x:B. f (f x)) (lambda y:C. lambda y)
```

Constraints Set: $\{A \to B \to X_2 = (C \to D \to C) \to X_3, A = B \to X_1, A = X_1 \to X_2\}$
Principal type: unification fails.