Assignment 9: Java, Java, Java

COS441: Programming Languages
Fall 2004

Assigned: Wednesday, 24 November 2004
Due: in class on Wednesday, 08 December 2004

Problem 1  [15 points] From the following rules

\[
\begin{align*}
\tau <: \tau & \quad \frac{\tau_1 <: \tau_3 \quad \tau_3 <: \tau_2}{\tau_1 <: \tau_2} \\
m > n & \quad \frac{\tau_1 \ast \ldots \ast \tau_m <: \tau_1 \ast \ldots \tau_n}{\forall 1 \leq i \leq n \quad \tau_i' <: \tau_i} \\
\{a : \tau_1, \ldots, z : \tau_m\} & <: \{a : \tau_1, \ldots, z : \tau_n\} \\
\tau_1 <: \tau_1' \quad \tau_2 <: \tau_2' & \quad \frac{\tau_1 \rightarrow \tau_2 <: \tau_1' \rightarrow \tau_2'}{	au' = \tau} \\
\tau'^\text{ref} <: \tau'^\text{ref} & \quad \tau' <: \tau
\end{align*}
\]

and assuming

\[
D <: B \quad E <: B \quad B <: A \quad C <: A
\]
determine which of these statements:

1. \(\tau <: \tau'\)
2. \(\tau' <: \tau\)
3. neither of the above

for the following cases

(a) \(\tau = E\)
\(\tau' = C\)

(b) \(\tau = A \ast B\)
\(\tau' = D \ast E\)

(c) \(\tau = B \rightarrow E\)
\(\tau' = D \rightarrow B\)

(d) \(\tau = \{a : A \ast A\}\)
\(\tau' = \{a : E \ast B, b : D\}\)

(e) \(\tau = A \rightarrow A \rightarrow C\)
\(\tau' = E \rightarrow D \rightarrow A\)

(f) \(\tau = A\text{ref} \rightarrow B \ast C\)
\(\tau' = B\text{ref} \rightarrow A \ast A\)
(g) \( \tau = A \rightarrow ((B \rightarrow A) \ast (B \rightarrow A)) \rightarrow B \ast E \)
\( \tau' = E \rightarrow (A \rightarrow B) \rightarrow D \)

(h) \( \tau = (E \ast D \ast A \rightarrow \{a : C\}) \rightarrow C \)
\( \tau' = (B \ast A \rightarrow \{a : A, b : C\} \rightarrow A \)

Problem 2  [10 points] Mitchell Exercise 10.2 (pg 296) The shape/ directory contains a Java version of the sample code provided in Mitchell. For problem (a) modify the Java code in the directory and submit the modified files online. Submit modified versions of all the files in the shape/ directory online.

Problem 3  [10 points] Mitchell Exercise 10.3 (pg 297) The directory exp/ contains a complete implementation of the visitor code written in Java. This is for clarification purposes only.

Problem 4  [10 points] Mitchell Exercise 13.3 (pg 420)

Problem 5  [15 points] Mitchell Exercise 13.5 (pg 421)

Problem 6  [10 points] Mitchell Exercise 13.6 (pg 423)

Problem 7  [15 points] Mitchell Exercise 13.9 (pg 425)

Problem 8  [15 points] The subdirectory fj/ contains a partial implementation of the static and dynamic semantics for Feather Weight Java. Both the dynamic and static semantics are almost complete except for the implementation of the following important auxiliary functions.

val getMth : tbl -> (cname * mname) -> mth

Defined in class-table.sml. Given a class table (tbl) a class name (c) and a method name (m) it should returns the appropriate method or raise the exception NoSuchMethod. It should search for a method named m in c if this it cannot find the m in c it should search for the method in the super-class of c. Note that the root of our class hierarchy Object is its own parent. You detect this case and avoid infinite looping. The function parentClass defined in class-table.sml maybe useful.

val subtype : (typ * typ) -> bool

Defined in static-sem.sml. If \( \tau_1 \ll \tau_2 \) then subtype(\( \tau_1, \tau_2 \)) should return true. Notice that the function is defined in a scope where it has access to information about the current class table. In particular the function val class : cname -> cls will return the class defintion for a given class name. As in the previous case, be careful to handle the special case of the Object class which is its own parent class.

Submit modified versions of fj/static-sem.sml and fj/class-table.sml.