HW 6: Substitution, LR Parsing
CS 440: Programming Languages and Translators
Due Sat Apr 14, 11:59 pm

Programming Problem [25 points]

1. Your job is to write a collection of substitution\(^1\) functions in Haskell for terms as represented below.

   ```haskell
   type Variable = String
   type Identifier = String
   data Term = VAR Variable | ID Identifier | FCN Identifier [Term]
   deriving (Show, Eq, Read)
   type Substitution = [(Variable, Term)]
   ```

   \(\text{-- The syntax of Variables and Identifiers is as in HW 5}\)

   (The Term type was called Ptree in HW 5).

   In the examples below, let \(t_1 = \text{FCN "f" [VAR "X", ID "y", FCN "g" [ID "z"]]}\) and let \(\text{subst}_1 = [(\"X\", \text{FCN "g" [ID "y"]})]\). (So \(t_1\) represents the term \(f(X, y, g(z))\) and \(\text{subst}_1\) represents the substitution \([X \mapsto g(y)]\). Note only variables appear in substitutions, not identifiers, so \((\"x\", \ldots)\) is not a legal substitution.

   You are to write three functions:

   - **sub_lookup subst var** looks for a binding for \(var\) in the substitution.
     - It returns Nothing if there isn't one and Just \(t\) if the binding \((var, t)\) is in \(\text{subst}\)\(^2\).
     - Example: \(\text{sub_lookup subst1 "X"} = \text{Just (FCN "g" [ID "y")}\)
     - Example: \(\text{sub_lookup subst1 "Y"} = \text{Nothing}\).

   - **sub_update subst (var, term)** updates the subst with \([var \mapsto term]\).
     - If \(var\) has no binding in \(\text{subst}\), then we simply add \((var, term)\) to the \(\text{subst}\) list.
     - If \(var\) is bound in \(\text{subst}\) then the result of update is like \(\text{subst}\) except that its binding of \(var\) is replaced by \((var, term)\).
     - Example: \(\text{sub_update subst1 ("Y", ID "q") = subst2 where subst2 = [("X", FCN "g" [ID "y")], ("Y", ID "q")}\)
     - (The order of bindings in a substitution doesn't matter: Your answer can have \(X\) then \(Y\) or \(Y\) then \(X\).)

---

\(^1\) See Lecture 18 for a definition of textual substitution.

\(^2\) The built-in datatype ```Maybe a = Nothing | Just a``` is just like the FAIL | OK ... we've been using.
• On the other hand, \texttt{sub_update subst1 ("X", ID "q") = [("X", ID "q")]}.  
• \texttt{sub_apply subst term} applies the substitution to the term.  
  • Example: Let \texttt{subst2 = [("X", FCN "g" [ID "y"]), ("Y", ID "q")]} (as above) and recall  
    \texttt{t1 = (FCN "f" [VAR "X", ID "y", FCN "g" [ID "z"]])}, then \texttt{sub_apply subst2 t1}  
    = \texttt{FCN "f" [FCN "g" [ID "y"], ID "y", FCN "g" [ID "z"]].}

\section*{Written Problems [25 points]}

1. Study the following grammar  

Rules:  
0. \texttt{S' \rightarrow S \$}  
1. \texttt{S \rightarrow x x a S}  
2. \texttt{S \rightarrow B c S}  
3. \texttt{S \rightarrow \varepsilon}  
4. \texttt{B \rightarrow x}  

(a) [4 points] Write out the First and Follow sets for the nonterminals.  
(b) [8 points] Write out the SLR characteristic finite state machine.  
(c) [5 points] Write out the Action / Go To table.  
(d) [8 points] Write out a trace of the parse for the input \texttt{x x a x c \$}.