School of Computer Science, University of Nottingham

COMP2012/G52LAC Languages and Computation, Spring 2019

Dr. Henrik Nilsson

Coursework Problems, Set 2

13 March 2019

Deadline: 20 March 2019, 3 PM

1. Consider the following Context-Free Grammar (CFG) G:

$$\begin{array}{rcl} S & \rightarrow & XYX \mid YXY \\ X & \rightarrow & aYb \mid ab \\ Y & \rightarrow & cXd \mid cd \end{array}$$

   S, X, Y are nonterminal symbols, S is the start symbol, and a, b, c, d are terminal symbols.

   (a) Derive the following words in the grammar G. Answer by giving the entire derivation sequence from the start symbol S:

      i. $abcdab$

      ii. $cacdbdacdbcd$

   (b) Does the string $abccddab$ belong to the language $L(G)$ generated by the grammar G? Provide a brief justification.

2. The syntax of programming languages is usually given by context-free grammars. Consider a very simple language J. Its syntax is given by the following grammar $G_J = (V, T, P, S)$, i.e. $J = L(G_J)$:

   - $V = \{\mathit{Prog}, \mathit{Stmts}, \mathit{Stmt}, \mathit{Expr}, \mathit{PrimExpr}, \mathit{BinOp}, \mathit{Id}, \mathit{Num}\}$
   - $T = \{\{, \}, (,), \texttt{if}, \texttt{print}, =, ;, +, -, *, \texttt{div}, \texttt{x}, \texttt{y}, \texttt{z}, \texttt{0}, \texttt{1}\}$
   - P is given by:

$$\begin{array}{rcl}
\mathit{Prog} & \rightarrow & \{\ \mathit{Stmts}\ \} \\
\mathit{Stmts} & \rightarrow & \mathit{Stmt}\ \mathit{Stmts} \mid \epsilon \\
\mathit{Stmt} & \rightarrow & \mathit{Id} = \mathit{Expr}\ ; \\
 & \mid & \texttt{if}\ (\ \mathit{Expr}\ )\ \mathit{Stmt} \\
 & \mid & \texttt{print}\ \mathit{Expr}\ ; \\
 & \mid & ; \\
 & \mid & \mathit{Prog} \\
\mathit{Expr} & \rightarrow & \mathit{Expr}\ \mathit{BinOp}\ \mathit{Expr} \mid \mathit{PrimExpr} \\
\mathit{PrimExpr} & \rightarrow & \mathit{Id} \mid \mathit{Num} \mid (\ \mathit{Expr}\ ) \\
\mathit{BinOp} & \rightarrow & \texttt{+} \mid \texttt{-} \mid \texttt{*} \mid \texttt{div} \\
\mathit{Id} & \rightarrow & \texttt{x} \mid \texttt{y} \mid \texttt{z} \\
\mathit{Num} & \rightarrow & \texttt{0} \mid \texttt{1}
\end{array}$$

   - $S = \mathit{Prog}$

   (To keep the grammar simple, the set of identifiers and the set of numeric literals are both finite.)

   (a) Which of the following are syntactically correct J programs? Which are not? You only need to state the answer for each one.

      i. `{ print x; { print y; } }`
      ii. `{ print (x + y; }`
      iii. `{ if (x) ; ; }`
      iv. `{ if (x = 0) print x; }`
      v. `{ x = x * (y - z + 1); print x; }`
      vi. `{ print z + 1 }`
      vii. `{ print ((0)); }`
      viii. `{ print; }`
      ix. `{ { { } } }`
      x. ϵ

   (b) Draw the derivation tree for

      `{ if (z) { print 1 ; print x + y ; } }`

   (c) Is the grammar G_J ambiguous? If no, briefly explain why. If yes, demonstrate this.

3. The following Context-free grammar (CFG) is immediately left-recursive:

$$\begin{array}{rcl}
S & \rightarrow & aS \mid bX \\
X & \rightarrow & XXc \mid YXd \mid Y \\
Y & \rightarrow & Ye \mid Yf \mid g
\end{array}$$

   S, X, and Y are nonterminals, a, b, c, d, e, f, and g are terminals, and S is the start symbol.

   Transform this grammar into an equivalent right-recursive CFG. State the general transformation rule you are using and show the main transformation steps.

4. Consider the following Context-Free Grammar (CFG):

$$\begin{array}{rcl} S & \rightarrow & AAS \mid ABB \\ A & \rightarrow & aA \mid \epsilon \\ B & \rightarrow & BCDb \mid \epsilon \\ C & \rightarrow & cD \mid ef \\ D & \rightarrow & dC \mid fe \end{array}$$

S, A, B, C, and D are nonterminals, a, b, c, d, e, and f are terminals, and S is the start symbol.

(a) What is the set N_ϵ of *nullable* nonterminals? Provide a brief justification.

(b) Systematically compute the *first sets* for all nonterminals, i.e., first(S), first(A), first(B), first(C), and first(D), by setting up and solving the equations according to the definitions of first sets for nonterminals and strings of grammar symbols. Show your calculations.

(c) Set up the subset constraint system that defines the *follow sets* for all nonterminals; i.e., follow(S), follow(A), follow(B), follow(C), and follow(D). Simplify where possible using the law

$$X \subseteq Z \wedge Y \subseteq Z \iff X \cup Y \subseteq Z$$

and by removing trivially satisfied constraints such as $\emptyset \subseteq X$ and $X \subseteq X$.

(d) Solve the subset constraint system for the follow sets from the previous question by finding the *smallest* sets satisfying the constraints.