CSE 5317 Spring 2010: Assignment 1

Assigned: 2 Feb 2010
Due: 11 Feb 2010, 2:00pm

For the written questions, please hand in a hard copy at the beginning of class on the due date, or email it to me before the deadline. Type or write neatly. For the programming component, email your Parse.java to nystrom@uta.edu before the deadline.

1 Scanning

(a) Write a regular expression that matches double numbers as in Java. Your expressions should cover all of the following examples:

   1. .2 3.14
   5e6 5e-6 5E+6
   7.e8 9.0E-10 .11e12

(b) Fortran provides string literals called Hollerith constants. These are written as an arbitrary positive integer \( n \), followed by an “H” followed by \( n \) characters. For example, “3Habc” is a Hollerith constant for the string “abc”. Write a regular expression that matches all Hollerith constants or explain why it cannot be done.

2 Parsing

Consider the following grammar that generates a subset of the expressions in the AWK programming language:

\[
\begin{align*}
\text{Expr} & ::= \text{Num} \mid \text{Lvalue} \mid \text{IncrOp Expr} \mid \text{Expr IncrOp} \mid \text{Expr BinOp Expr} \mid (\text{Expr}) \\
\text{Lvalue} & ::= \$ \text{Expr} \\
\text{IncrOp} & ::= \text{++} \mid \text{--} \\
\text{BinOp} & ::= + \mid - \mid \epsilon \\
\text{Num} & ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{align*}
\]

Note the empty binary operator (written \( \epsilon \)) for string concatenation.

The various operators have precedences as follows, from highest (tightest) to lowest: (1) lvalue (\( \$ \)), (2) post-increment and post-decrement, (3) pre-increment and pre-decrement, (4) addition and subtraction, (5) string concatenation (the empty BinOp). All binary operators are left associative.

(a) Rewrite the grammar so it respects the above precedence rules. Show your work. Hint: First remove the BinOp ::= \( \epsilon \) rule and introduce a rule for sequences of expressions; then handle the lower precedence operators by introducing nonterminals for each precedence level.

(b) Rewrite the grammar from (a) into a grammar that is LL(1). Show your work.