HW1
50pts
Posted Friday, September 2, 2022
Due Friday, September 16, 2022

Note: Lecture 2 covers material for problems 1-3. Lectures 3 and 4 cover problems 4-5.

Problem 1 (10pts). Describe in English the languages denoted by the following regular expressions.

(a) (2pts) $\texttt{a(a|b)}^*\texttt{a}$
(b) (2pts) $\texttt{(b}^*\texttt{(}\epsilon\texttt{|a))}^*$
(c) (2pts) $\texttt{(a|b)}^*\texttt{a(a|b)(a|b)}$
(d) (2pts) $\texttt{a}^*\texttt{ba}^*\texttt{ba}^*\texttt{ba}^*$
(e) (2pts) $\texttt{(aa|bb)}^*\texttt{((ab|ba)(aa|bb)}^*\texttt{(ab|ba)(aa|bb)}^*\texttt{)}^*$

Note: Your description should be the most general high-level characterization. For example, $\texttt{(ba}^*\texttt{ba}^*\texttt{)}^*$ should be described as "All strings of `a`'s and `b`'s, beginning with `b` and having even number of `b`'s." not as, for example, "The string of `b` followed by any number of `a`'s followed by a `b` followed by any number of `a`'s, repeated any number of times.".

Problem 2 (10pts). The following grammar generates numbers in binary notation. C is the start symbol.

(1) $C \to C\ \texttt{0} \mid A\ \texttt{1} \mid \texttt{0}$
(2) $A \to B\ \texttt{0} \mid C\ \texttt{1} \mid \texttt{1}$
(3) $B \to A\ \texttt{0} \mid B\ \texttt{1}$

(a)(2pts) Construct a derivation that generates the binary notation of 21.
(b)(4pts) Prove that the generated numbers are multiples of 3.
(c)(4pts) Prove that all such numbers are generated by the grammar.

Note: To receive full credit your answer must set up a proof and clearly state the key steps/arguments. Be concise and use mathematical notation instead of prose as much as possible. Overly verbose answers may incur a penalty.

Problem 3 (10pts). The following is an ambiguous expression grammar with one unary operator * and n binary infix operators, $\theta_1, \theta_2, ..., \theta_n$, at n different levels of precedence:

$$expr \to expr\ \theta_1\ expr \mid expr\ \theta_2\ expr \mid ... \mid expr\ \theta_n\ expr \mid expr^* \mid (expr) \mid \texttt{id}$$

(a) (2pts) Show that the grammar is ambiguous.
(b) (8pts) Construct an equivalent unambiguous grammar such that binary operators $\theta_1, \theta_2, ..., \theta_n$ are all *left-associative*. Unary operator * has the highest precedence. θ_n has precedence over θ_{n-1}, θ_{n-1} has precedence over θ_{n-2} and so on, and θ_1 has the lowest precedence.

Problem 4 (10pts). Consider the following LL(1) grammar for a simplified subset of Lisp:

$$\begin{aligned} P &\rightarrow E \ \texttt{\$\$} \\ E &\rightarrow \texttt{atom} \\ E &\rightarrow \texttt{'} \ E \\ E &\rightarrow \texttt{(} \ E \ Es \ \texttt{)} \\ Es &\rightarrow E \ Es \\ Es &\rightarrow \epsilon \end{aligned}$$

`atom`, `'`, `(`, `)`, and `$$` are the terminals (tokens), and P, E and Es are the nonterminals.

(a) (3pts) What is FOLLOW(Es)? FOLLOW(E)? PREDICT($Es \rightarrow \epsilon$)?

(b) (3pts) Give a parse tree for the string `(cdr '(a b c)) $$`. Note: keyword `cdr` and identifiers `a`, `b`, and `c` are `atom`s.

(c) (4pts) Consider a recursive descent parser running on the same input. At the point where the quote token (`'`) is matched, which recursive descent routines will be active (i.e., what routines will have a frame on the run-time stack)?

Problem 5 (10pts). For each grammar below, determine if it is LL(1) or not and if it is ambiguous or not. You do not need to justify your answer, just write YES or NO. As an example of the format for answers we are looking for: (x) LL(1): YES, Ambiguous: YES.

(a) (2pts) $A \rightarrow 0 \ A \ 1 \mid 0 \ 1$

(b) (2pts) $A \rightarrow + \ A \ A \mid * \ A \ A \mid \texttt{a}$

(c) (2pts) $A \rightarrow A \ (\ A \ ) \ A \mid \epsilon$

(d) (2pts) $A \rightarrow B \ \texttt{a} \mid \texttt{b} \ B \ \texttt{c} \mid \texttt{d c} \mid \texttt{b d c} \qquad B \rightarrow \texttt{d}$

(e) (2pts) $S \rightarrow C\texttt{a}C\texttt{b} \mid D\texttt{b}D\texttt{a} \qquad C \rightarrow \epsilon \qquad D \rightarrow \epsilon$