

## Problem Set 1: CFGs and Top-Down Parsing

---

Due: February 6<sup>th</sup> at 5:00 p.m.

This week's assignment consists of written exercises on regular expressions, formal grammars, and top-down parsing. You are to write up solutions and hand them in on paper.

- 1) Consider the regular language described by  $(\epsilon + c)(cb)^*ca$ .
  - a) Develop an  $\epsilon$ -NFA that accepts this language. It's okay to collapse unimportant states.
  - b) Convert the  $\epsilon$ -NFA into a DFA using subset construction. Show your work. You do not need to minimize the DFA.
- 2) In PP-1, you used a lex pattern similar or equal to  $[0-9]^+$  to recognize decimal integers.
  - a) Specify a lex pattern that recognizes only decimal integers that are divisible by 2.
  - b) Specify a lex pattern that recognizes only decimal integers that are divisible by 3.  
Yes, it is possible. ☺
- 3) Consider the following grammar for regular expressions:

$$\begin{aligned} N &= \{R\} \\ T &= \{ |, *, (, ), <id> \} \\ P &= \{ R \rightarrow R R, R \rightarrow R | R, R \rightarrow R^*, R \rightarrow (R), R \rightarrow <id> \} \end{aligned}$$

The start symbol is R.

- a) Show that this grammar is ambiguous. Give an example.
  - b) How would the ambiguity cause problems when interpreting the parse trees?
  - c) Construct an LL(1) grammar for regular expressions that gives  $*$  higher precedence than  $|$ .
- 4) Consider the following context-free grammar:

$$\begin{aligned} N &= \{S, G, R, L, C\} \\ T &= \{n, s, c, y\} \\ P &= \{ S \rightarrow G, G \rightarrow R, G \rightarrow G R, R \rightarrow n L s, L \rightarrow C, L \rightarrow L c C, C \rightarrow \epsilon, C \rightarrow C y \} \end{aligned}$$

The start symbol is S.

- a) Is this grammar ambiguous?
- b) Explain why the grammar is not LL(1).

- c) Transform the grammar into a LL(1) grammar that accepts the same language.
- d) Develop the parse table for a top-down predictive parser for your LL(1) grammar.
- e) Trace through the parse for the input **nycys**. (turn over, there's more!)
- f) Sketch the functions required for a recursive-descent parser for your LL(1) grammar.
- g) Is the language described by the grammar regular?