CS4120/4121/5120/5121—Spring 2020
Homework 2
Syntactic Analysis
Due: Saturday, February 8, 11:59PM

0 Updates

- Problem 4 has been changed to ask for Earley parsing using the same grammar. You are also welcome to solve the previous problem 4 involving SLR parsing.

1 Instructions

1.1 Partners

You may work alone or with *one* partner on this assignment. But remember that the course staff is happy to help with problems you run into. Use Piazza for questions, attend office hours, or set up meetings with any course staff member for help.

1.2 Homework structure

There are two parts of the homework. The first part is required of all students. The second part is required of students taking CS5120, but those enrolled in CS4120 are welcome to try it for good HARMA.

1.3 Tips

You may find the Dot and Graphviz packages helpful for drawing graphs. You can get these packages for multiple OSes from the Graphviz download page.

2 Problems

1. Context-free grammars

Consider the following BNF grammar over the alphabet {`let`, `n`, `p`, `q`, `+`, `-`, `;`, `=`}, where the start symbol is S:

$$
\begin{aligned}
S &\to D \mid SD \\
D &\to \texttt{let n = } E\texttt{;} \\
E &\to T\texttt{+}E \mid T\texttt{-}E \mid T \\
T &\to V \mid \varepsilon \\
V &\to \texttt{p}V \mid \texttt{q}V \mid \texttt{p} \mid \texttt{q}
\end{aligned}
$$

(a) Briefly identify two reasons why this grammar is not LL(1).

(b) Is it possible to refactor this grammar into an equivalent one that is LL(1)? If so, give the new grammar. If not, explain why such a grammar cannot be written.

(c) Is the string `let n = -pqp;` in the language of the grammar? If so, show the rightmost derivation of the string. If not, explain why it cannot be derived.

(d) Show the start state for the LR(1) automaton for this grammar. (Hint: remember to take the closure of items.)

2. Designing grammars

Consider the basic symbols needed to represent simple arithmetic operations. The terminals are {num, (,), +, -, *, /, %, ^} and the operations are as follows (in decreasing order of precedence):

Operator	Associativity
^	right
*, /, %	left
+, -	left

A restricted first attempt might be

$$\begin{aligned} E &\rightarrow E\ Op\ E \mid (E) \mid \texttt{<num>} \\ Op &\rightarrow + \mid * \mid \texttt{\^{}} \end{aligned}$$

This grammar has a few problems: it is ambiguous, it does not enforce operator precedence, and it does not enforce associativity.

(a) Give an example to show that the grammar is ambiguous.

(b) Write an LR(1) grammar for the full problem domain, including all the operators in the table with the appropriate precedence relationships and associativity. You don't need to show parse tables.

(c) Show the bottom-up derivation of 4^5%2 + 1 * (2 + 2) using your grammar.

3. LR grammars

Is the following grammar LALR(1)? What about LR(1)? Explain.

$$\begin{aligned} S &\rightarrow \texttt{c}X\texttt{q} \mid \texttt{c}Y\texttt{p} \mid X\texttt{p} \mid Y\texttt{q} \\ X &\rightarrow \texttt{a} \\ Y &\rightarrow \texttt{a} \end{aligned}$$

3 Problem for CS5120

4. Earley parsing

Consider the following grammar:

$$S \rightarrow SA \mid A$$
$$A \rightarrow a$$

Show the steps of parsing the string "aaa"' using an Earley parser, including all the sets I_j.

4 Submission

Submit your solution as a PDF file on CMS. This file should contain your name, your NetID, all known issues you have with your solution, and the names of anyone you have discussed the homework with.