CS4120/4121/5120/5121—Spring 2016
Homework 2
Syntactic Analysis
Due: Friday, February 12, 11:59PM

0 Updates

- 2/6: Problem 1(c): Require EOF ($) as an additional column in the parse table.

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 KARMA.

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 a simple markup language that uses tags. Possible terminal symbols are <, >, /, =, and `word`. Every tag begins with < and ends with >. A tag may be an open or a close tag. In an open tag, the first token after < is a `word` representing the tag's name, followed by an optional list of attributes which are pairs of `word`s related by =. In a close tag, the first token after < is a /, followed by the tag's name, but no attributes. Every open tag must be paired with a close tag. Any number of `word`s or tags may appear between an open and close tags. For example, here is a valid string in this markup language:

```
<word word=word word=word><word>word word word</word><word></word></word>
```

(a) Write a context-free grammar for this language.

(b) Find the nullable, FIRST, and FOLLOW sets for your grammar in part (a).

(c) Construct the LL(1) parse table and clearly identify any conflicts in the table. Make sure the columns of the table are in this order: <, >, /, =, `word`, and `$`.

(d) Fundamentally, what makes this language not LL(1)? Is there a k for which this language is LL(k)?

2. Ambiguous grammars

Consider the following grammar:

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

This grammar is ambiguous. We would like a grammar to derive parse trees in which exponentiation (^) has higher precedence than multiplication (*), and both have higher precedence than addition (+). Exponentiation should be right-associative, while multiplication and addition should be left-associative.

(a) Show that the grammar is ambiguous.

(b) Write an LL(1) grammar that accepts the same string as this grammar and respects the desired operator precedence. Associativity need not be enforced, however.

(c) Show the derivation of the expression `2^3*4+5` using your grammar in part (b).

(d) Write an LR(1) grammar that accepts the same language, but enforces both the precedence and associativity of the operators. You need not show the parsing tables.

3. LR grammars

Show that the following grammar is LR(1) but not LALR(1):

$$\begin{aligned} S &\rightarrow A\texttt{a} \mid \texttt{b}A\texttt{c} \mid B\texttt{c} \mid \texttt{b}B\texttt{a} \\ A &\rightarrow \texttt{d} \\ B &\rightarrow \texttt{d} \end{aligned}$$

3 Problem for CS5120

4. Predictive parsing

Describe a grammar that is not LL(1), but for which every nonterminal has a unique applicable production given the first token, i.e., given a nonterminal, the FIRST sets of its productions are disjoint.

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.