CS 4240 Compilers and Interpreters - Spring 2015
Assignment 3 (100 pts)
Due in class: Tuesday, March 3, 2015

Guidelines

1. Georgia Tech Honor Code will be enforced.
2. Answers should be concise, complete and precise.
3. Return a hard copy of the homework, the homework is due at the beginning of the class; late submissions should be slid under Dr. Naik's door, KACB 2320 or brought to class (no later than 1:30pm on Thu, March 5.)

Questions

Question 1 (35 points)

Consider the following grammar (<list> is the start symbol):
<lexp> → <atom> | <list>
<atom> → **<number> | <identifier>**
<list> → '(' <lexp-seq> ')'
<lexp-seq> → <lexp> ',' <lexp-seq> | <lexp>

1. Left factor this grammar.
2. Construct First and Follow sets for the non-terminals of the resulting grammar.
3. Show that the resulting grammar is LL(1).
4. Construct the LL(1) parsing table for the resulting grammar.
5. Show the actions of the corresponding LL(1) parser, given the following input string: (a , (b , (2)) , (c))
6. Find the average number of derivations done for the above input per token consumed.

Question 2 (30 points)

Consider an LL(1) grammar with k rules. Answer the following questions:

1. First construct a grammar for $k = 5$ which forces the LL(1) parsing algorithm to produce the highest number of derivations (expansions) to be done on the stack before a token is consumed. We will call this the worst case complexity of LL(1) parsing.
2. For this grammar, construct an input that will force the highest number of derivations.
3. If the number of tokens were to be n, what is the complexity (in terms of total number of derivations to consume n tokens) of this grammar? Generalize your observation in (a) in terms of k and give a formula using n and k for worst case complexity of any LL(1) grammar.

Question 3 (35 points)

Consider the following grammar (<E> is the start symbol).
<E> → '(' <L> ')' | 'a'
<L> → <L> ',' <E> | <E>

1. Construct the DFA of LR(0) items for this grammar.
2. Construct the SLR(1) parsing table.
3. Show the parsing stack and the actions of an SLR(1) parser for the following input string: ((a) , a , (a , a))
4. Is this grammar an LR(0) grammar? If not, describe the LR(0) conflict. If so, construct the LR(0) parsing table, and describe how a parse might differ from an SLR(1) parse.
5. Is it LL(1)? Prove it or show LL(1) conflict.