Exercises for LLLR course
Midlands Graduate School 2016

Hayo Thielecke

April 10, 2016

See `http://www.cs.bham.ac.uk/~hxt/2015/mgs-ll-lr/`

These exercises do not have to be attempted in order; you may mix and match them as you like. Some are meant to be straightforward, whereas others may require some thought.

1. Write a rigorous inductive definition of the set of parse trees for a given grammar.
2. Describe in general what the stuck states of the LL machine are, in which the machine can make no further steps, but cannot accept the input. Hint: there are 3 cases, plus another one if the grammar is silly.
3. Show that the LL machine for a grammar is not always deterministic.
4. Does the LL machine for any grammar always terminate?
5. Are nontermination of the LL machine and FIRST/FIRST conflict the same problem? Discuss.
6. Consider the following grammar:

   $$\begin{array}{lcl} S & \to & a\,S\,b\,S \\ S & \to & c\,S\,d\,S \\ S & \to & \varepsilon \end{array}$$

   Show how the LL machine can accept the input

   $$a\,b\,c\,d$$

   by giving an accepting machine run.

7. Same as previous exercise, but for the LR machine.

8. Prove soundness of the LL and LR machines: if the machine accepts its input, then there is a derivation of the string in the grammar

9. Suppose we have a rule in our grammar like this:

$$E \to E\,\alpha$$

   Explain why this grammar rule causes a problem for the LL(1) machine. Hint: suppose we also have $E \to b$, and think about b.

10. Recall that for any k, LL(k) is the set of languages that can be parsed by an LL machine with k symbols of lookahead. In particular, what is LL(0)? One can give a very concise description without speaking about parsing.

11. Prove that the FIRST and FOLLOW calculation terminates.

12. Which of these are a problem for LL(1), and why?

    (a) The same symbols occur on the right-hand side of two rules for the same non-terminal symbol.

    (b) The same symbol occurs as the first symbol on the right-hand side of two rules for the same non-terminal symbol.

    (c) A non-terminal symbol occurs on the right-hand side of one of its rules.

    (d) A non-terminal symbol occurs multiple times on the right-hand side of one of its rules.

    (e) The same symbol occurs on the right-hand side of two rules.

    (f) There are no symbols on the right-hand side of some rules.

    (g) A non-terminal symbol occurs as the first symbol on the right-hand side of one of its rules.

    (h) A non-terminal symbol A occurs as the first symbol of a rule for a non-terminal B, such that B occurs as the first symbol of a rule for A

13. Consider the following grammar:

$$\begin{array}{rcl} S & \to & A \\ A & \to & b \\ D & \to & b \end{array}$$

Discuss whether this grammar presents a problem for LR(0) parsing, or not. Your answer should refer to the LR and/or LR(0) machines.

14. Implement the LL(1) machine for this grammar in a language of your choice, preferably C. Bonus for writing the shortest possible implementation in C.

15. Construct the LR(0) automaton for the Dyck language grammar:

$$\begin{array}{rcl} D & \to & [\, D\,]\, D \\ D & \to & (\, D\,)\, D \\ D & \to & \end{array}$$