

# Compiler Construction (List 4)

Hans de Nivelle

23 October 2013

1. Consider the language  $(\Sigma, R, S)$ , defined by  $\Sigma = \{ '(,')' \}$ ,  $R = \{ S \rightarrow SS, S \rightarrow (S), S \rightarrow \epsilon \}$ .

- Give derivations for

$'(())()'$ ,  $'((())'$ ,  $'(((())'$ .

- Show that the language is ambiguous. (There are words that have more than one derivation.)
  - Repair the grammar, so that it is not ambiguous anymore, but still accepts the same set of words.
  - Prove that the repaired grammar is equivalent (defines the same language) as the original grammar.
2. (a) In the programming language Lisp, everything is a list. The empty list has form  $()$  or **nil**. Non-empty lists have form  $(L_1)$ ,  $(L_1 L_2)$ ,  $(L_1 L_2 L_3)$ , etc. Give a complete grammar for Lisp. The elements of a list can be atoms, numbers, or by themselves lists. (You may ignore the existence of dotted pairs and arrays.)
  - (b) Give a grammar for the language of Prolog-style lists. Lists have form

$[], [L], [L_1, L_2], [L_1, L_2, L_3]$ , etc.

The elements of the lists can by themselves be lists again.

- (c) Consider the language consisting of functional expressions of form  $c$ , and  $f(t_1, \dots, t_n)$ , with  $n > 0$ , and  $t_1, \dots, t_n$  functional expressions by themselves. Give a grammar for this language.
3. (a) Give a attribute functions for the grammars of Task 3. Take into account that  $(L_1 \dots L_n)$  denotes **cons** $(L_1, \mathbf{cons}(L_2, \dots \mathbf{nil}))$ . Give a derivation for the list

$(\mathbf{car} (\mathbf{quote} (1\ 2\ 3)))$ .

4. The construction on the slides **parsing.pdf**, pages 30-31, (creating a grammar from a set of operators with priorities) is not correct. Can you find the problem? Can you repair it? What do you think is better? Repairing the problem, or adding conditions to the priorities of the operators?