Exercise Compiler Construction (4)

Hans de Nivelle

Due: 4 November 2009

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.

2. (a) In the programming language Lisp, everything is a list. The empty list has form ( ) or nil. Non-empty lists have form (L1), (L1 L2), (L1 L2 L3), 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], [L1, L2], [L1, L2, L3], etc.

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

   (c) Consider the language consisting of functional expressions of form c, and $f(t_1, \ldots, t_n)$, with $n > 0$, and $t_1, \ldots, t_n$ functional expressions by themselves. Give a grammar for this language.

   (d) Give a grammar for propositional formulas consisting of propositional variables, and the following operators:

<table>
<thead>
<tr>
<th>operator</th>
<th>meaning</th>
<th>binding strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>negation</td>
<td>600, prefix</td>
</tr>
<tr>
<td>&amp;</td>
<td>and</td>
<td>500, leftassoc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or</td>
</tr>
<tr>
<td>$\rightarrow$</td>
<td>implication</td>
<td>300, rightassoc</td>
</tr>
<tr>
<td>$\leftrightarrow$</td>
<td>equivalence</td>
<td>200, nonassoc.</td>
</tr>
</tbody>
</table>

   In addition to the operators, the language must allow parentheses.
3. (a) Give the attribute functions for the grammar of Task 2a. Take into account that \((L_1 \cdots L_n)\) denotes \texttt{cons}(L_1, \texttt{cons}(L_2, \ldots \texttt{nil})). Give a derivation for the list

\[(\texttt{car (quote (1 2 3)))}. \]

(b) Give the attribute functions for the grammar in Task 2d.