Exercise Compiler Construction (5)

Hans de Nivelle

Date: 08.11.2011

1. Consider the grammar $\mathcal{G} = (\{\hat{S}, S, T, U, \#, +, *, \mathrm{id}, (,)\}, \hat{S}, R)$ with $R =$

   $\hat{S} \to S\#$
   $S \to S + T$
   $S \to T$
   $T \to T * U$
   $T \to U$
   $U \to \mathrm{id}$
   $U \to (S)$

   (a) Draw the prefix automaton for $\mathcal{G}$.
   (b) Identify the states in which reductions are possible. In which of these states is a look ahead necessary?
   (c) Compute the lookaheads, using the algorithm in the slides.

2. (a) Draw the prefix automaton for the language of Task list 4, 2b.
   (b) Identify the states in which reductions are possible. Determine the states in which look ahead is necessary.
   (c) Compute the look ahead sets, using the algorithm on the slides.

3. Consider the following grammar $\mathcal{G} = (\{\hat{S}, S, A, B, a, b, c\}, \hat{S}, R)$ with $R =$

   $\hat{S} \to S\#$
   $S \to cbca$
   $S \to AaAb$
   $A \to B$
   $B \to c$

   (a) Draw the prefix automaton for this language.
   (b) Identify the states in which reductions are possible. Compute the lookaheads, using the algorithm in the slides.
   (c) Actually, this is an example, where the algorithm in the slides computes too big look aheads. What is the consequence of this?
   (d) Determine the correct lookahead sets.