Exercises for FLL, Fall 2018, sheet 5 – Solutions

Return Thursday Oct 18, in class

Exercise 1. Give a CFG for all words over the terminal alphabet $T = \{\boldsymbol{a}, \boldsymbol{b}, +, *, (,), \boldsymbol{\varepsilon}, \varnothing\}$ that are regular expressions over $\Sigma = \{a, b\}$. Adhere to the strict syntax of regular expressions that was given in Definition 3.13 in the lecture notes.

Solution. Put $V = \{E\}$ (which automatically makes E the start variable). Then simply replicate the inductive definition of regexps:

$E \to \boldsymbol{a} \mid \boldsymbol{b} \mid \boldsymbol{\varepsilon} \mid \varnothing \mid (EE) \mid (E+E) \mid (E*)$

Exercise 2. Give a CFG for the language of the regular expression (0*10)*, where your grammar uses at most two variables.

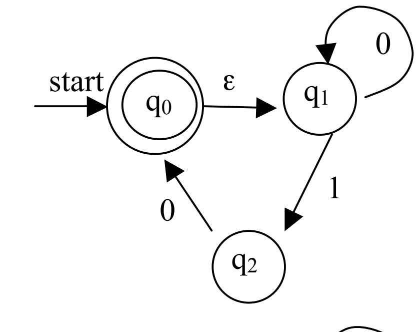
Solution: A set of production rules that works is the following:

$S \to \varepsilon \mid SS$	comment: this takes care of the outer *
$S \to Z10$	
$Z \to \varepsilon \mid ZZ \mid 0$	comment: rules in the last two lines take care of generating the words 0*10

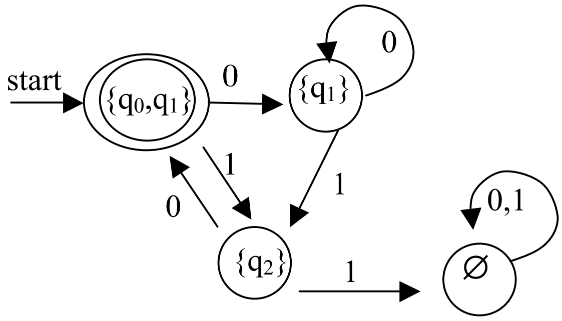
Exercise 3. Give a *right-linear* CFG for the language of the regular expression (0*10)*.

Solution. The mechanical straightforward way to get such a grammar is to (i) find an epsilon-NFA for the language $L((0^*10)^*)$, then (ii) turn this into a DFA, then (iii) transform the DFA to a right-linear grammar using the recipe from Prop. 4.3 in the LN. Here are the three substeps:

(i)



(ii)



(iii) For convenience we rename the DFA states $\{q_0,q_1\}$ -> *S*, $\{q_1\}$ -> *T*, $\{q_2\}$ -> *U*, ∅ -> *V*. Then the recipe from Prop. 4.3 gives the following grammar:

$S \rightarrow 0T \mid 1U \mid \varepsilon$
$T \rightarrow 0T \mid 1U$
$U \rightarrow 0S \mid 1V \mid 0$
$V \rightarrow 0V \mid 1V$