Umeå University
Department of Computing Science

Language and Computation 7.5 p
5DV162

Assignment 1

Submitted
2016-09-16

Authors:
Lorenz Gerber (dv15lgr@cs.umu.se, lozger03@student.umu.se)

Instructors:
Henrik Björklund

Problem 1

Assume that L is a regular language. Strings w of language L are constructed according to the specification where zero, one or several ‘a’s and zero, one or several ‘b’s are mixed freely as long as there are more ‘b’s than ‘a’s in it. Let’s have a look at such a generic string and how it can be split into three parts xyz such that xy is shorter than m, x is 1 or larger and all xyiz are part of L for all i e N. Independent of how m is chosen, if y contains more a than b, which according to the language definition is possible, the resulting strings for xyiz i e N are no longer part of L. Hence, proven by contradiction, L can not be a regular language.

Problem 2

Assume that L is a regular language. Strings w of language L are constructed according to the specification as a^n with n being a prime number equal or larger than 2. For every prime n, there has to be a number m which is smaller than n, and that multiplied with any natural number plus the difference of p and m will be a prime. Such a number does not exist. Hence, L can not be a regular language.

Problem 3

Conversion of context free grammar into Chomsky normal form:

$$\begin{aligned}
&S \to aAB \\
&A \to aAa \\
&A \to bb \\
&B \to a
\end{aligned}$$

$$\begin{aligned}
&S_0 \to aAB \\
&S \to aAB \\
&A \to aAa|bb \\
&B \to a
\end{aligned}$$

$$\begin{aligned}
&S_0 \to BAB \\
&S \to BAB \\
&A \to BAB|bb \\
&B \to a
\end{aligned}$$

$$S_0 \to BU$$
$$S \to BU$$
$$A \to BU|bb$$
$$U \to AB$$

$$S_0 \to BU$$
$$S \to BU$$
$$A \to BU|bb$$
$$U \to AB$$
$$V \to b$$

Problem 4

Proving that the resulting Language Ls from shuffeling the regular languages L1 and L2 is also regular was attempted here by construction.

The idea is to construct a NFA from NFA/DFA's that accept L1 and L2. Let's assume The wordlength of L1 and L2 is |w1| and |w2|, hence the NFA/DFA to accept L1 or L2 will have u1...u|w1| respectively v...v|w2| states. Now, for example state u1 is split into |w1| substates to 'remember' how many states of NFA 2 already have been passed. The same is done for all states of the NFA 1 and 2. Hence the new NFA will in this case have 18 + 2 states (start end accept state). All inbound transitions are the same as in the initial NFA's. Hence it has been proven by construction of a new NFA that the shuffle of L1 and L2 is also a regular language.

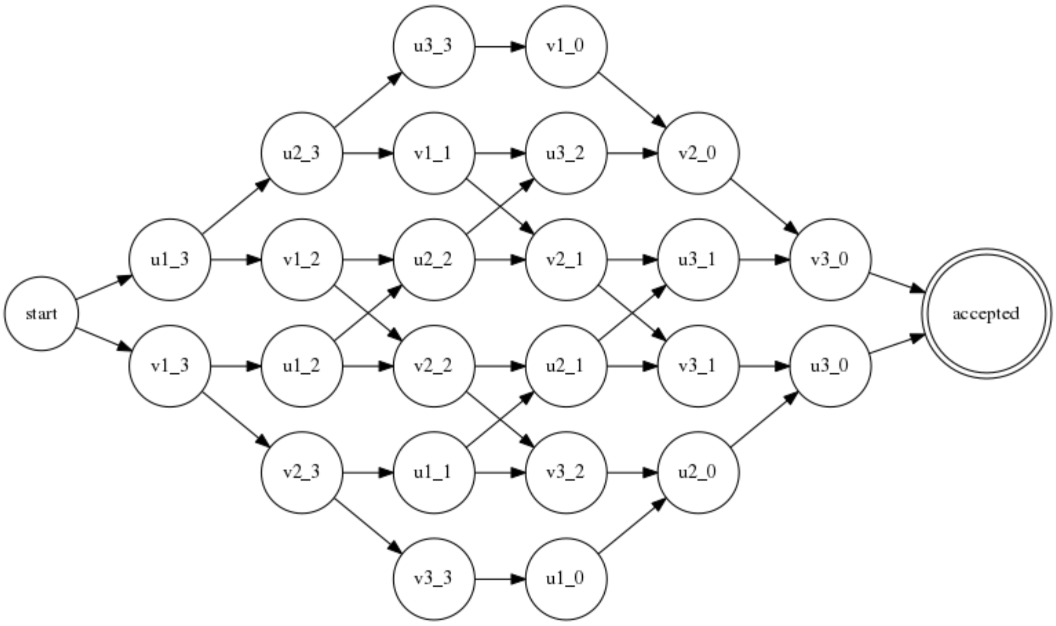

Figure 1: *Non-Deterministic Finite Automaton of shuffle between L1 and L2.*