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 bs 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 xy^iz are part of L for all $i \in \mathbb{N}$. Independent of how m is chosen, if y contains more a's than b's, which according to the language definition is possible, the resulting strings for xy^iz $i \in \mathbb{N}$ are for large enough i 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 $n-m$ will be a prime. Such a number does not exist. Hence, L can not be a regular language.

The above is my initial solution. After some research on the net I happend to find an exact formal solution to the problem [2]. It took some time until I understood it but now I do and therefore I reproduce it here, though, admitting that I probably could not have come up with it by myself.

1. Assume L is regular, hence there has to be an m
2. choose a word $w = a^n$ where n is a prime and $|xyz| = n > m+1$ (comment: The '$m+1$' part seems to be the critical assumption which I hardly could have come up with).
3. as $|xy| \leqslant m$ it follows that $|z| > 1$
4. as $|z| > 1$ it follows that $|xy| > 1$. Now choose $i = |xy|$. Then $|xy^iz| = |xy| + |y||xz| = (1+|y|)$. As $(1+|y|)$ and $|xy|$ are both greater than 1, the product must be a composite number. Hence$|xy^iz|$ is a composite number and not a prime.

Problem 3

Conversion of context free grammar into Chomsky normal form:

First Rule 1 is applied;

$$S \rightarrow aAB$$
$$A \rightarrow aAa$$
$$A \rightarrow bb$$
$$B \rightarrow a$$

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

No application for the epsilon and unit rule. The last shown steps are according rule four:

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

$$
\begin{aligned}
S_0 &\rightarrow BU \\
S &\rightarrow BU \\
A &\rightarrow BU|bb \\
U &\rightarrow AB
\end{aligned}
$$

$$
\begin{aligned}
S_0 &\rightarrow BU \\
S &\rightarrow BU \\
A &\rightarrow BU|bb \\
U &\rightarrow AB \\
V &\rightarrow b
\end{aligned}
$$

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 $u_1 \ldots u_{|w1|}$ respectively $v_1 \ldots v_{|w2|}$ 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 (18 + start and accepted end state). All inbound transitions are the same as in the

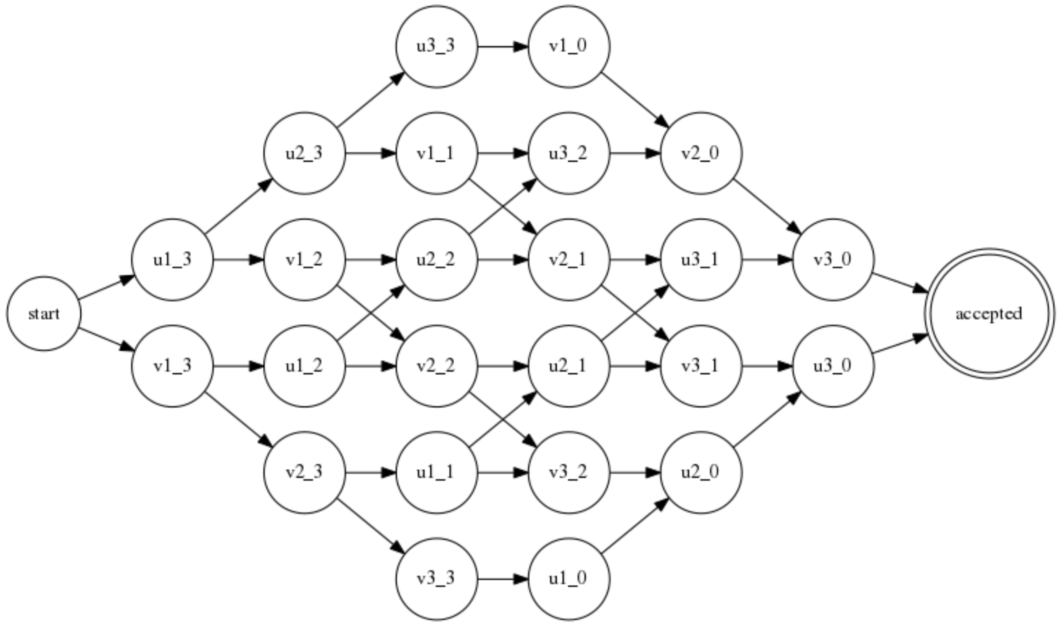

Figure 1: *Non-Deterministic Finite Automaton of shuffle between L1 and L2, generated with graphviz.*

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.

The solution is shown in figure 1 produced with graphviz [1]. Figure 2 shows a hand draft of the same figure, but with a more clear grouping.

References

[1] Graphviz - graph visualization software, 2016. accessed: 2016-09-20.

[2] Penn engineering, course homepage, theory of computation, david matuszek. `http://www.seas.upenn.edu/~cit596/notes/dave/pumping6.html/`, 1996. accessed: 2016-09-20.

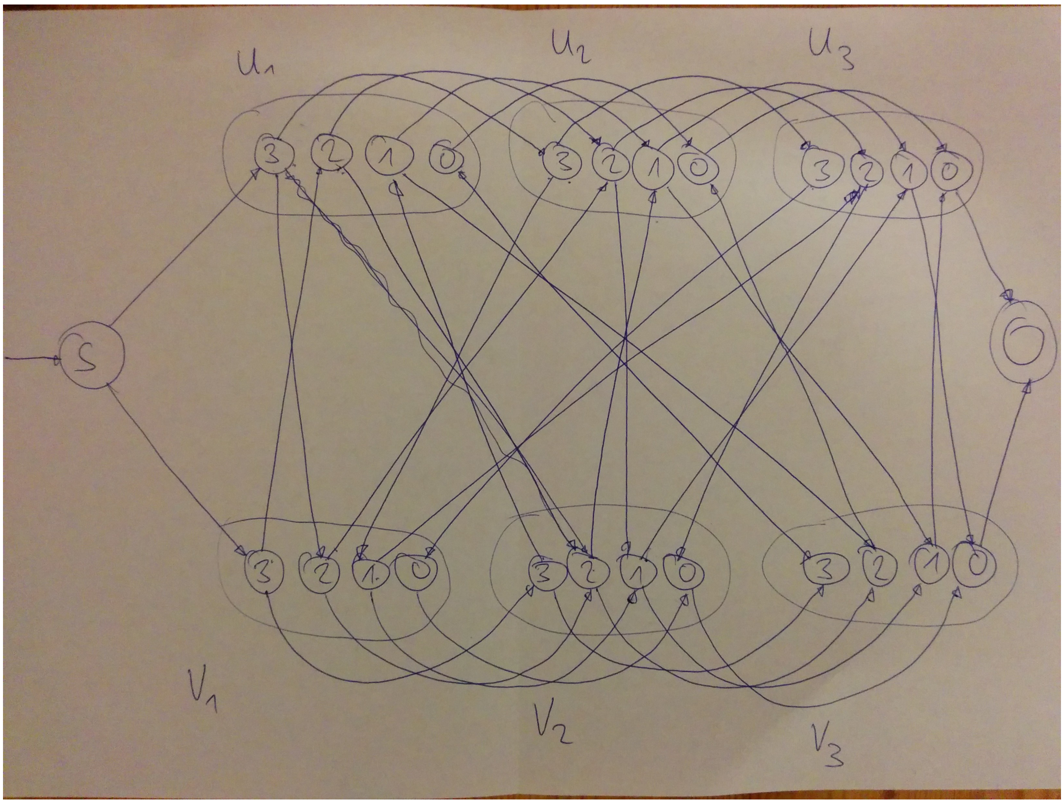

Figure 2: *Non-Deterministic Finite Automaton of shuffle between L1 and L2, hand draft.*