Prof. Javier Esparza
Philipp Czerner

Technical University of Munich
Chair for Foundations of Software Reliability

Automata and Formal Languages

Winter Term 2023/24 – Exercise Sheet 6

Exercise 6.1.

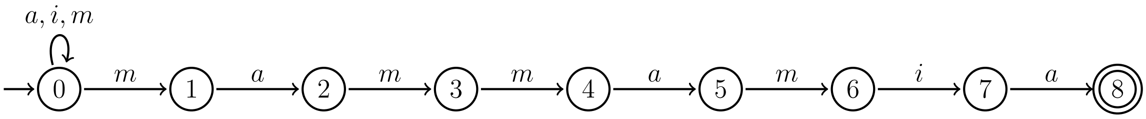
(a) Build the automata B_p and C_p for the word pattern $p = mammamia$.

(b) How many transitions are taken when reading $t = mami$ in B_p and C_p?

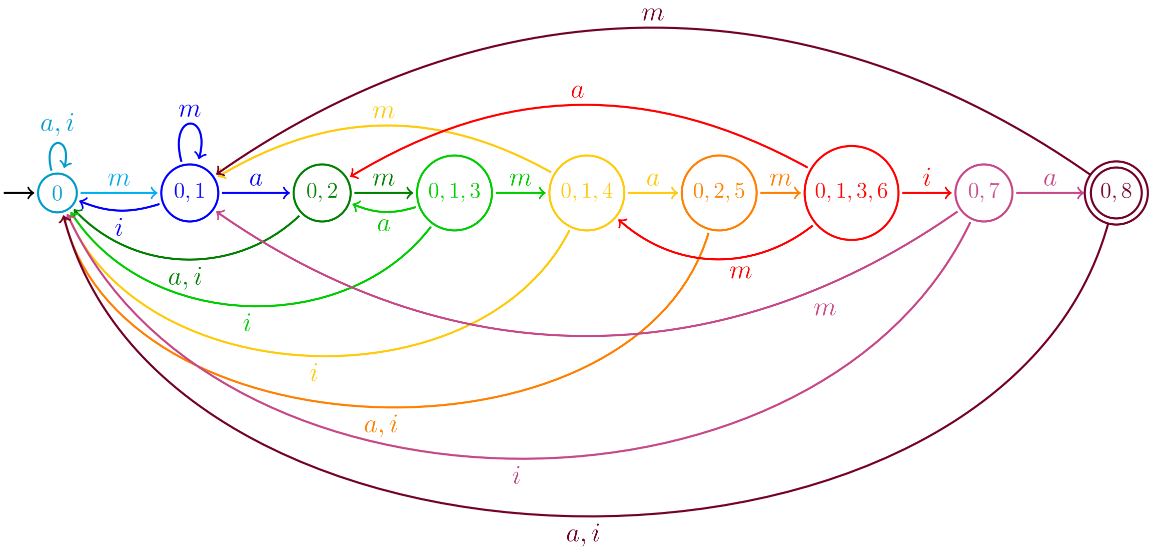
(c) Let $n > 0$. Find a text $t \in \{a, b\}^*$ and a word pattern $p \in \{a, b\}^n$ such that testing whether p occurs in t takes n transitions in B_p and $2n - 1$ transitions in C_p.

Solution.

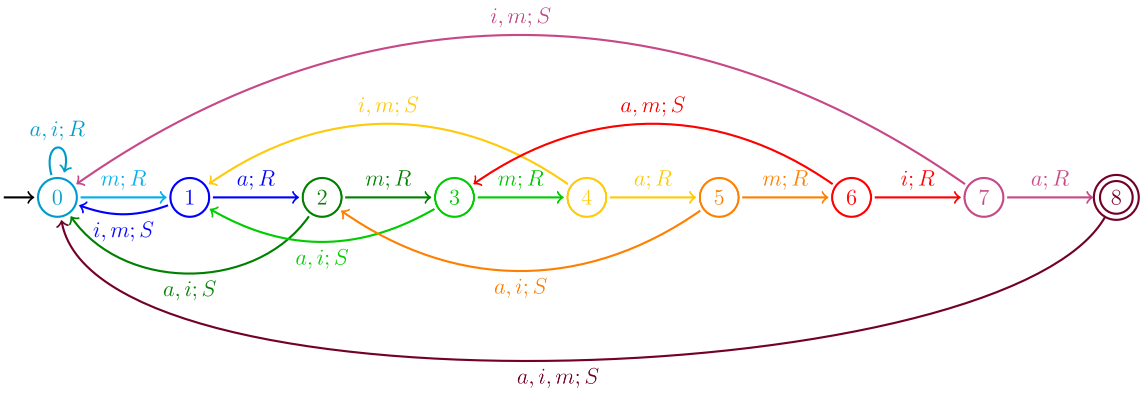
(a) A_p :



B_p :



C_p :

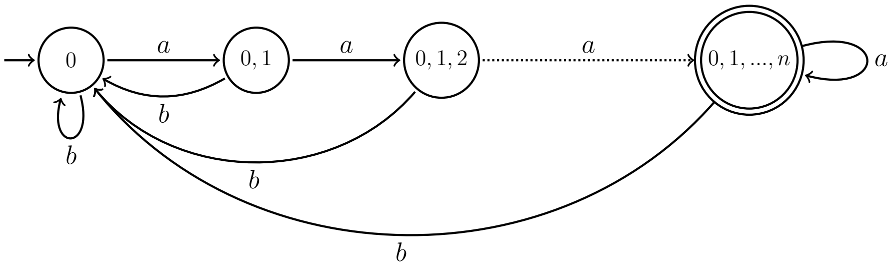


(b) Four transitions taken in B_p: $\{0\} \xrightarrow{m} \{0,1\} \xrightarrow{a} \{0,2\} \xrightarrow{m} \{0,1,3\} \xrightarrow{i} \{0\}$.

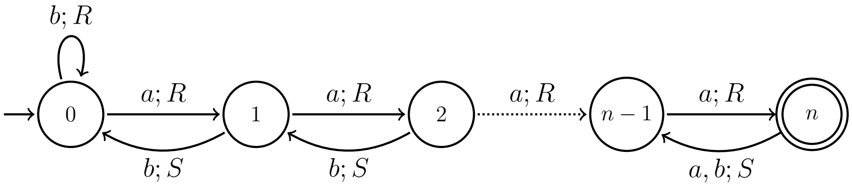
Six transitions taken in C_p: $0 \xrightarrow{m} 1 \xrightarrow{a} 2 \xrightarrow{m} 3 \xrightarrow{i} 1 \xrightarrow{i} 0 \xrightarrow{i} 0$.

(c) $t = a^{n-1}b$ and $p = a^n$. The automata B_p and C_p are as follows:

B_p:



C_p:



The runs over t on B_p and C_p are respectively:

$$\{0\} \xrightarrow{a} \{0,1\} \xrightarrow{a} \{0,1,2\} \xrightarrow{a} \cdots \xrightarrow{a} \{0,1,...,n-1\} \xrightarrow{b} \{0\} \ ,$$

and

$$0 \xrightarrow{a} 1 \xrightarrow{a} 2 \xrightarrow{a} \cdots \xrightarrow{a} (n-1) \xrightarrow{b} (n-2) \xrightarrow{b} (n-3) \xrightarrow{b} \cdots \xrightarrow{b} 0 \ .$$

Exercise 6.2.

In order to make pattern-matching robust to typos we want to include also “similar” words in our results. For this we consider words with a small Levenshtein-distance (edit-distance) “similar”.

We transform a word w to a new word w' using the following operations (with $a_i, b \in \Sigma$):

- *replace* (R): $a_1 \dots a_{i-1} a_i a_{i+1} \dots a_l \to a_1 \dots a_{i-1} b a_{i+1} \dots a_l$
- *delete* (D): $a_1 \dots a_{i-1} a_i a_{i+1} \dots a_l \to a_1 \dots a_{i-1} \varepsilon a_{i+1} \dots a_l$
- *insert* (I): $a_1 \dots a_{i-1} a_i a_{i+1} \dots a_l \to a_1 \dots a_{i-1} a_i b a_{i+1} \dots a_l$

The Levenshtein-distance (denoted $\Delta(w, w')$) of w and w' is the minimal number of operations (R,D,I) needed to transform w into w'. We denote with $\Delta_{L,i} = \{w \in \Sigma^* \mid \exists w' \in L.\, \Delta(w', w) \leq i\}$ the language of all words with edit-distance at most i to some word of L.

(a) Compute $\Delta(\mathit{become}, \mathit{bekommen})$ and $\Delta(\mathit{become}, \mathit{werden})$.

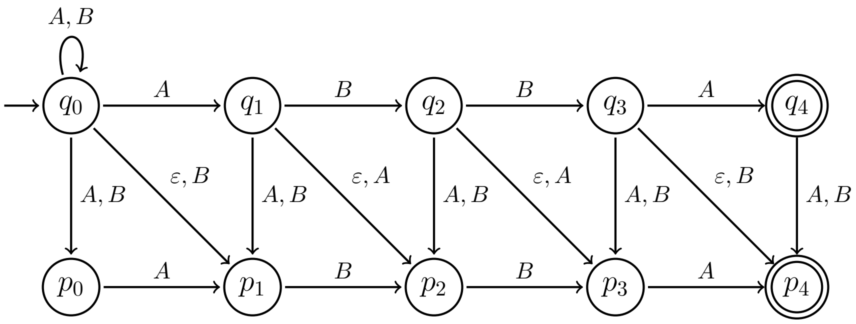
(b) Let p be the pattern $ABBA$. Construct an NFA-ϵ locating the pattern or variations of it with edit-distance 1.

(c) Prove the following statement: If L is a regular language, then $\Delta_{L,n}$ is a regular language.

Solution.

(a) $\Delta(\mathit{become}, \mathit{bekommen}) = 3$, $\Delta(\mathit{become}, \mathit{werden}) = 5$.

(b) We use the automaton A_p for pattern $p = ABBA$ and duplicate it carefully in order to allow up to one "mistake".



(c) Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA for L. We obtain an NFA-ϵ N for $\Delta_{L,n}$ by adding n "error-levels". Formally:

$$N = (Q \times [0, n], \Sigma, \delta', (q_0, 0), F \times [0, n])$$

with

$$\begin{aligned}
\delta' &= \{((q,i), a, (p,i)) \mid q, p \in Q \wedge i \leq n \wedge a \in \Sigma \wedge \delta(q,a) = p\} && \text{no change}\\
&\cup \{((q,i), \varepsilon, (p, i+1)) \mid q, p \in Q \wedge i < n \wedge (\exists a \in \Sigma.\, \delta(q,a) = p)\} && \text{delete}\\
&\cup \{((q,i), a, (q, i+1)) \mid q \in Q \wedge i < n \wedge a \in \Sigma\} && \text{insert}\\
&\cup \{((q,i), b, (p, i+1)) \mid q, p \in Q \wedge i < n \wedge (\exists a \in \Sigma \setminus \{b\}.\, \delta(q,a) = p)\} && \text{replace}
\end{aligned}$$

Let us prove that $\Delta_{L,n} = L(N)$.

$\Delta_{L,n} \subseteq L(N)$. If $w \in \Delta_{L,n}$, it means that there is $w' \in L$ such that $\Delta(w', w) = k \leq n$, or in other words, starting from the word w', we can obtain w by applying k "mistakes" (delete, insert, replace). As $w' \in L$ (accepted by M) and as the 0-level of N is a copy of M, note that w' has a run in N that reaches a final state $(q_f, 0)$. By construction of the automaton N, there is a run of the word w that follows the run of w' where each "mistake" can be seen as moving to the next error-level, using the corresponding transition from δ' (delete, insert, replace) depending on a mistake. It is easy to see that if the word w' reaches a final state $(q_f, 0)$ in N, then w can reach (q_f, k), and thus $w \in L(N)$.

$L(N) \subseteq \Delta_{L,n}$. If $w \in L(N)$, this means there is a run of w in N that reaches a final state $(q_f, k) \in F \times [0, n]$. Intuitively, for each transition of that run that changes the level, we modify w so that it "stays in the same level". Formally, we check the nature of the transition that changes the level and modify w as follows:
(i) If $(p, i) \xrightarrow{a} (p, i+1)$ is an insert edge, this occurrence of the letter a will be removed from w.
(ii) If $(p, i) \xrightarrow{a} (q, i+1)$ is a replace edge, and there exists a $(p, i) \xrightarrow{b} (q, i)$ edge, for some letter b, then we replace this occurrence of a in w with b.
(iii) If $(p, i) \xrightarrow{\epsilon} (q, i+1)$ is a delete edge, and there exists a $(p, i) \xrightarrow{a} (q, i)$ edge, for some letter a, then we add the letter a at this place in w.

Denote the obtained word by w'. It is easy to see that w' is obtained from w by applying mistakes (delete, insert, replace) k times, as in the run of w there are exactly k transitions that change the level. Therefore, $\Delta(w', w) \leq k \leq n$. Moreover, it is easy to see that if w reaches (q_f, k), then w' reaches $(q_f, 0)$. As the 0-level is a copy of M, then $w' \in L$. To summarize, there exists $w' \in L$ such that $\Delta(w', w) \leq n$, that is, $w \in \Delta_{L,n}$.

Exercise 6.3.

Consider transducers whose transitions are labeled by elements of $(\Sigma \cup \{\varepsilon\}) \times (\Sigma^* \cup \{\varepsilon\})$. Intuitively, each transition reads one or zero letter and writes a word of arbitrary length. Such a transducer can be used to perform operations on strings, e.g. upon reading `"singing in the rain"` it could write `Singing In The Rain`.

Sketch such ε-transducers for the following operations, each of which is informally defined by means of three examples. For each example, when the transducer reads the string on the left, it should write the string on the right. You may assume that the alphabet Σ consists of $\{a, b, \ldots, z, A, B, \ldots, Z\}$, a whitespace symbol, and an end-of-line symbol. Moreover, you may assume that every string ends with an end-of-line symbol and contains no other occurrence of the end-of-line symbol.

(a)

Input	Output
`Automata and Formal Languages`	`AFL`
`Technical University of Munich`	`TUM`
`Max Planck Institute`	`MPI`

(b) For this exercise, Σ is extended with $\{,, \cdot\}$.

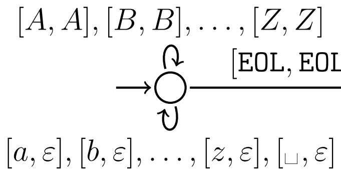
Input	Output
`Ada Lovelace`	`Lovelace, A.`
`Alan Turing`	`Turing, A.`
`Donald Knuth`	`Knuth, D.`

(c) For this exercise, Σ is extended with $\{0, 1, \ldots, 9, (,), +\}$. We want to transform phone-numbers into a normal form, where they are prefixed with a country code.

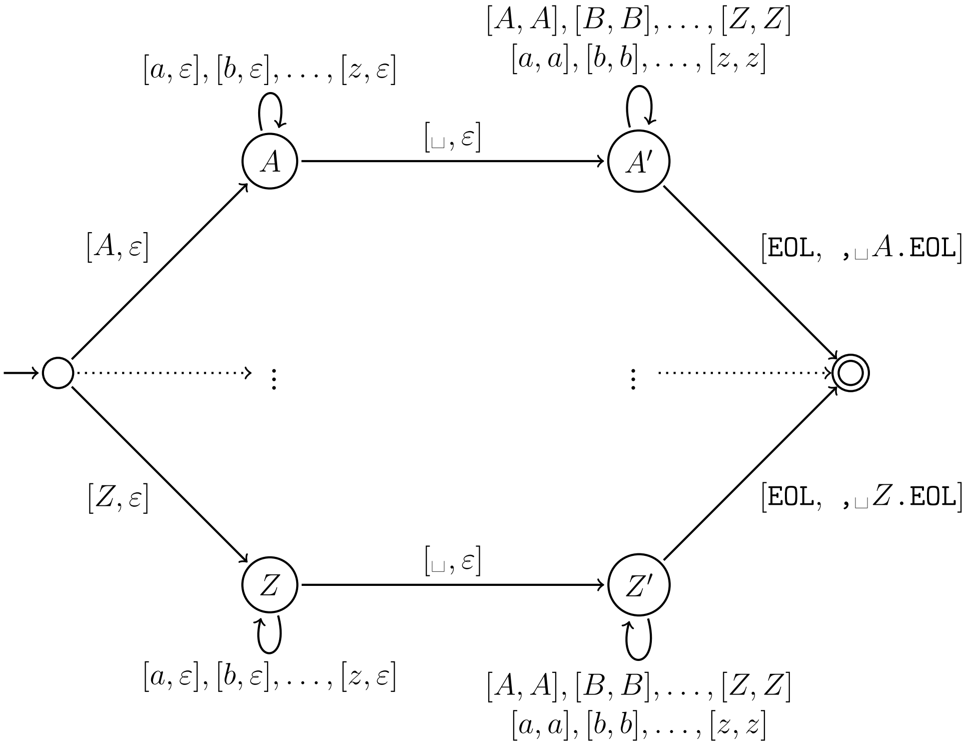
Input	Output
004989273452	+49 89 273452
(00)4989273452	+49 89 273452
273452	+49 89 273452
2 7 3 4 5 2	+49 89 273452
498949	+49 89 498949
+49 89 498949	+49 89 498949

Solution.

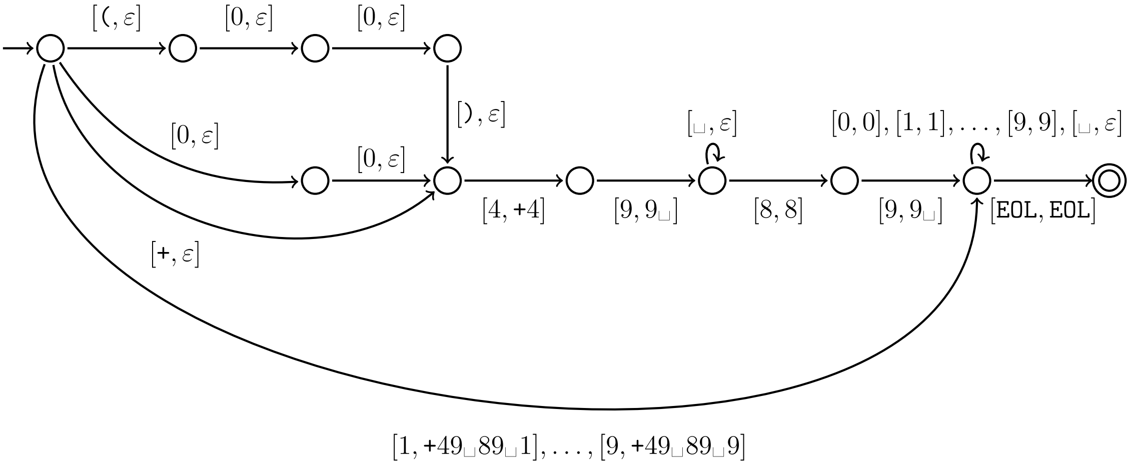
(a)


$[A, A], [B, B], \ldots, [Z, Z]$
$[\texttt{EOL}, \texttt{EOL}]$
$[a, \varepsilon], [b, \varepsilon], \ldots, [z, \varepsilon], [\textvisiblespace, \varepsilon]$

(b)


$[A, A], [B, B], \ldots, [Z, Z]$
$[a, a], [b, b], \ldots, [z, z]$
$[a, \varepsilon], [b, \varepsilon], \ldots, [z, \varepsilon]$
A
$[\textvisiblespace, \varepsilon]$
A'
$[A, \varepsilon]$
$[\texttt{EOL}, \texttt{,}\textvisiblespace A\texttt{.EOL}]$
$[Z, \varepsilon]$
$[\texttt{EOL}, \texttt{,}\textvisiblespace Z\texttt{.EOL}]$
Z
$[\textvisiblespace, \varepsilon]$
Z'
$[a, \varepsilon], [b, \varepsilon], \ldots, [z, \varepsilon]$
$[A, A], [B, B], \ldots, [Z, Z]$
$[a, a], [b, b], \ldots, [z, z]$

(c)


[(, ε]
[0, ε]
[0, ε]
[), ε]
[0, ε]
[0, ε]
[+, ε]
[4, +4]
[9, 9␣]
[␣, ε]
[8, 8]
[9, 9␣]
[0, 0], [1, 1], . . . , [9, 9], [␣, ε]
[EOL, EOL]
[1, +49␣89␣1], . . . , [9, +49␣89␣9]