This test consists of four exercises over 2 pages. It is advised to explain your approach and to check your answers carefully. You can score a maximum of 100 points. Each question indicates how many points it is worth. The test is closed book. You are NOT allowed to use a calculator, a computer or a mobile phone. You may answer in Dutch or in English. Please write clearly, and do not forget to put on each page: your name, your student number, and your werkcollege group. Put your student-card clearly visible at the corner of your table for inspection.

**Notation**  Throughout the test, we denote for any alphabet $A$ and $a \in A$ by $|w|_a$ the number of $a$’s in the word $w \in A^*$, as it was introduced in the exercises.

# 1 Non-deterministic Finite Automata

a) Let $N$ be the NFA given by the following diagram

![NFA Diagram](image)

i) Give a $\lambda$-NFA $N'$ with one final state that accepts the same language as $N$.  

Solution: .................................................................

ii) Construct from $N'$ a regular expression that generates the language accepted by $N'$, using the procedure from the lecture. All intermediate steps belong to your answer.

Solution: .................................................................

There are two possible ways of collapsing states.

1. First collapse to

\[ \lambda \]
and then to
\[ b + aba + a(baa + bb)^*(a + ba) \]

This results into the regular expression
\[ (b + aba + a(baa + bb)^*(a + ba))^{*}(\lambda + a(baa + bb)^*b). \]

2. The alternative is to first collapse \( q_1 \). This gives

\[ b + aa \]

and then
\[ b + aa + ab(bb)^*(a + ba) \]

and this results into
\[ (b + aa + ab(bb)^*(a + ba))^{*}(\lambda + ab(bb)^*). \]

\( \square \)

b) Let \( \mathcal{N} \) be the NFA over the alphabet \{a, b\} be given by the following diagram.
Use the subset construction to obtain a DFA that accepts the same language as $\mathcal{N}$. Leave out unreachable states and clearly mark the states by the set of states they are generated from.

Solution: .................................................................

\[ \begin{align*}
&\{q_0\} \xrightarrow{a} \{q_0, q_1\} \\
&\{q_0, q_1\} \xrightarrow{b} \{q_0, q_1, q_2\} \\
&\{q_0, q_1, q_2\} \xrightarrow{b} \{q_0, q_1, q_3\}
\end{align*} \]

2 Pumping Lemma for Regular Languages

Let $A$ be the alphabet $\{a, b\}$ and $L$ the language

$$L = \{vv^R \mid v \in A^*, |v|_a + |v|_b = 2k + 1, k \in \mathbb{N}\}.$$

Take care that the word, you choose in the contradiction, is indeed in $L$.

Solution: .................................................................

We assume that $L$ is regular and has the pumping length $p > 0$. The word we use in the contradiction is $w = vv^R = a^n bba^n$ with

$$n = \begin{cases} p, & p \text{ even} \\ p + 1, & p \text{ odd} \end{cases}.$$

Clearly, $w \in L$, as $|v|_a + |v|_b$ is odd. By the pumping lemma, there are $x, u, y$ with $w = xuy$, $|xu| \leq p$ and $|x| = s > 0$, such that $xu^iy \in L$ for all $i \in \mathbb{N}$. Observe that we have $xu = a^k$ for some $k > 0$, as $|xu| \leq p$. Thus $xu^2y \neq v^i v^R$ for any $v^i \in A^*$, since $xu^2y = a^{k+s}a^2a^{n-k}bba^n = a^{n+s}bba^n$ and $s > 0$, and $xu^2y$ cannot be in $L$, which is a contradiction. Hence $L$ is not regular.

3 Context Free Grammars

Let $A$ be the alphabet $\{a, b\}$ and $L$ again be the language

$$L = \{vv^R \mid v \in A^*, |v|_a + |v|_b = 2k + 1, k \in \mathbb{N}\}.$$
a) Give a grammar $G$ that generates the language $L$. (15pt)

Solution: ............................................................... 

A possible grammar $G$ is given by

$$
S \rightarrow aXa | bXb \\
X \rightarrow aSa | bSb | \lambda
$$

b) Show that the word $abbbba$ is generated by $G$. (5pt)

Solution: ............................................................... 

$$
S \rightarrow aXa \rightarrow abSba \rightarrow abbbXba \rightarrow abbbba
$$

c) Show that the words $aba$ and $abba$ are not generated. (15pt)

Solution: ............................................................... 

- To generate $aba$, we need to use $S \rightarrow aXa$ to generate the $a$ at the beginning. But then cannot use any other production, since all productions of $X$ generate either zero or two $b$, contradicting the shape of $aba$. Hence $aba$ is not generated by $G$.

- Here, we need to use again $S \rightarrow aXa$ to generate the $a$ at the beginning. Next, we need to use $X \rightarrow bSb$, to generate the $b$ after the first $a$. But now we are stuck with $abSba$, since there is no production $S \rightarrow \lambda$. Thus $abba$ is not generated.

4 Push Down Automata

a) Let $\Sigma = \{a, b, c\}$ and the language $L$ be given by

$$
L = \{w \in \Sigma^* | \text{ } |w|_a = |w|_b + |w|_c \}. 
$$

Give a PDA with one state that accepts exactly the language $L$. Clearly indicate the stack alphabet you are using. Moreover, give the accepting computation for the word $abca$. (13pt)

Solution: ............................................................... 

Since each $a$ can be accounted for by $b$ or $c$ in words of $L$, we use the stack alphabet $\Gamma = \{a, b \lor c\}$ (or any other two-symbol alphabet). A possible PDA accepting $L$ is then given by
The word *abca* is accepted by

\[(q, abca, \lambda) \rightarrow (q, bca, c \lor b) \rightarrow (q, ca, \lambda) \rightarrow (q, a, a) \rightarrow (q, \lambda, \lambda).\]

b) Let $G$ be the grammar on the alphabet \{a, b\} given by

\[
S \rightarrow \lambda | aS | bSB \\
B \rightarrow b | aS
\]

Construct a two state PDA that accepts the language generated by $G$. (12pt)

Solution: 

\[
\begin{align*}
\lambda, S/\lambda \\
a, S/S \\
b, S/SB
\end{align*}
\]

\[
\begin{align*}
\lambda, \lambda/\lambda \\
a, \lambda/S \\
b, \lambda/SB
\end{align*}
\]