Theoretical Computer Science (Bridging Course)

Dr. G. D. Tipaldi
F. Boniardi
Winter semester 2014/2015

University of Freiburg
Department of Computer Science

Exercise Sheet 3

Due: 20th November 2014

Exercise 3.1 (Regular languages, Pumping lemma)
Are the following languages regular? Prove it.

(a) $L := \{a^i b^j a^{ij} \mid i, j \geq 0\}$.

Solution:
The language is not regular. To show this, let's suppose L to be a regular language with pumping length $p > 0$. Furthermore, let's consider the string $w = a^p b^p a^{p^2}$. It is apparent that $|w| \geq p$ and $w \in L$. According to the pumping lemma, $w = xyz$ where

- $|xy| \leq p$.
- $y \neq \epsilon$.
- $xy^k z \in L$ for all $k \in \mathbb{N}_0$.

Consequently, $xy^0z = xz$ must belong to L. Since $|xy| \leq p$ and $|y| > 0$, then it is easy to see that $xy^0z = a^{p-|y|} b^p a^{p^2}$ is not a member of L. Thus, L is not regular.

(b) $L := \{b^2 a^n b^m a^3 \mid m, n \geq 0\}$.

Solution:
The Language is regular. Indeed, it can be expressed by the following regular expression:

$$\mathcal{R} := b^2 a^* b^* a^3.$$

(c) $L := \{a^{k^3} \mid k \geq 0\}$.

Solution:
The language is not regular. Again, let's suppose that L is regular with pumping length $p > 0$. The string $w := a^{p^3}$ contradicts the pumping lemma. Indeed, if $w = xyz$ so that the statement of the pumping lemma holds, then it is easy to see that $xy^k z = a^{p^3+(k-1)|y|}$. However, if such y existed, then $p^3 + (k-1)|y| = n(k)^3$ for every $k \geq 0$, where $n(k) \in \mathbb{N}_0$ depends upon k, which is trivially false.

Exercise 3.2 (Pumping Lemma)
Find the minimum pumping length of the languages $L(\mathcal{R})$ where

(a) $\mathcal{R} = \mathcal{R}_1 := 0^*101^*$.

Solution:
The pumping length p must be grater than 2. Indeed, $L(\mathcal{R})$ contains only strings of length at least 2, furthermore $10 \in L(\mathcal{R})$ and cannot be pumped. Let now $w \in L(\mathcal{R}_1)$ so that $|w| \geq 3$, we clain that $p = 3$. To prove this, let's consider three cases:

1. $w = 0 \cdots 010$, i.e. w is 10 anteceded by at least a 0. In such case it is easy to see that we can write w as the concatenation of three strings xyz where $x = \epsilon$, $y = 0$ and z is the remaining substring. It is apparent that x, y and z satisfy the pumping lemma.

2. $w = 101\cdots1$, i.e. w is 10 followed by at least a 1. We can define $x = 10$, $y = 1$ and $z = \epsilon$. Again, x, y and z satisfy the pumping lemma.
3. $w = 0\cdots0101\cdots1$, that is, 10 is both anteceded by at least a 0 and followed by at least a 1. We can choose x, y and z either as in case 1. or in case 2.

(b) $\mathcal{R} = \mathcal{R}_2 := 10^*1$.

Solution:
Strings of length 2 cannot be pumped. However, we claim that the pumping length is 3. Indeed, let $w \in L(\mathcal{R})$ so that $|w| \geq 3$, then $w = 10\cdots01$ (eventually the two 1s bracket a single 0). As a consequence we can select $x = 1$, $y = 0$ and $z = 1$ so the pumping lemma is satisfied.

(c) $\mathcal{R} := \mathcal{R}_1 \cup \mathcal{R}_2$.

Solution:
Given two regular languages $L_1, L_2 \subseteq \Sigma^*$ with minimum pumping length $p_1, p_2 \geq 0$ and set $p_\cup$ to be the minimum pumping length of $L_1 \cup L_2$, it is easy to see that

$$p_\cup = \max\{p_1, p_2\}.$$

To prove this, observe first that $p_\cup \leq \max\{p_1, p_2\}$. Indeed, $\max\{p_1, p_2\} \geq p_1, p_2$ and let $w \in L_1 \cup L_2$ so that $|w| \geq \max\{p_1, p_2\}$, then $|w| \geq p_1, p_2$. Since w belongs to L_1 or to L_2, then by definition of pumping length, w can be pumped in both languages. Furthermore, let's suppose $p_\cup < \max\{p_1, p_2\}$, hence, $p_\cup < p_1$ or $p_\cup < p_2$. Let's assume $p_\cup < p_1$, then all words in $L_1 \cup L_2 \supset L_1$ with length at least $p_\cup$ could be pumped. This would imply that p_1 is not the minimum pumping length for L_1.

Since $L(\mathcal{R}) = L(\mathcal{R}_1) \cup L(\mathcal{R}_2)$, thus $p = 3$.

Exercise 3.3 (Context-free languages)

(a) Provide a context-free grammar $G = (V, \Sigma, R, S)$ that generates the language of palindromes over an alphabet Ξ .

Solution:
For the sake of clearness, say that $\Xi = \{\xi_1, ..., \xi_n\}$. We can define a context-free grammar as follows

- $V = \{S\}$.
- $\Sigma := \Xi$.
- Defining $\xi_1, ..., \xi_n$ to be the symbols in the alphabet, then the set R of production rules can be defined as follows:

$$S \to \epsilon \mid \xi_1 \mid \ \ldots \ \mid \xi_n,$$
$$S \to \xi_1 S \xi_1 \mid \ \ldots \ \mid \xi_n S \xi_n.$$

- S is the start variable.

(b) Prove that $L(G) = L_{pal}$.

Solution: We apply the induction principle on the length of the word. Using strong induction can simplify the proof.

· <u>$n = 0, 1$</u>. The grammar contains all the possible words on Ξ of length at most 1. Such words are trivially palindromes.

– <u>induction</u>. Let's suppose that all words of length k are palindromes for any $k = 0, ..., n-1$. We know that every words of length n is generated as $\xi_j S \xi_j$ where $\xi_j \in \Xi$ is an arbitrary letter and S is a word of length either $n-1$ or $n-2$. By induction hypothesis S is a palindrome and so is $\xi_j S \xi_j$.

The proof is complete.

(c) Consider the context-free grammar $(\{X, Y\}, \{0, 1\}, R, X)$ where R is defined as follows

$$
\begin{aligned}
&X \to \epsilon \mid 1, \\
&X \to 1\,X\,1 \mid Y, \\
&Y \to \epsilon \mid 0, \\
&Y \to 0\,Y\,0.
\end{aligned}
$$

Which language does this context-free grammar generate?

Solution:
It is easy to see that the above grammar generates binary strings as follows:

$$\underbrace{0 \cdots 0}_{m}, \tag{1}$$

$$\underbrace{1 \cdots 1}_{n}, \tag{2}$$

$$\underbrace{1 \cdots 1}_{n}\underbrace{0 \cdots 0}_{m}\underbrace{1 \cdots 1}_{n} \tag{3}$$

with $n, m \geq 0$.
Strings of type (1) can be easily generated by starting from X and applying $[X \to Y]$ followed by an arbitrary sequence of $[Y \to 0\,Y\,0]$ and $[Y \to \epsilon \mid 0]$. Strings of type (2) can be obtained applying either $[X \to \epsilon \mid 1]$ or $[X \to 1\,X\,1]$. Type (3) requires all the generation rules specified by R.