CSCI 2400 – Models of Computation, Section 3

Solutions to Homework 6

Problem 1.
(a) $L = \{a^n : \ n \text{ is prime}\}$ (A prime number is divided only by itself and 1.)

Assume for contradiction that L is a context-free language. We apply the pumping lemma. Let m be the parameter of the pumping lemma. Let p be a prime such that $p \geq m$. We choose to pump the string $a^p \in L$. Since $a^p = uvxyz$, we have that $v = a^k$ and $y = a^l$, with $k+l \geq 1$ (since $|vy| \geq 1$). From the pumping lemma we have that $uv^{1+p}xy^{1+p}z \in L$, and therefore $a^{p+kp+lp} \in L$. Subsequently, $a^{p(1+k+l)} \in L$, which is impossible since $p(1+k+l)$ is not a prime. Thus, we have a contradiction and the language L is not context-free.

(b) $L = \{a^n b^j : n \leq j^2\}$

Assume for contradiction that L is a context-free language. We apply the pumping lemma. Let m be the parameter of the pumping lemma. We choose to pump the string $a^{m^2} b^m \in L$. We have that $a^{m^2} b^m = uvxyz$, with $|vxy| \leq m$ and $|vy| \geq 1$.

We examine all the possible cases for the position of string vxy. First we note that the string v cannot span simultaneously both a^{m^2} and b^m, since if we pump up v (repeat v), the resulting string is not in the language (a's are mixed with b's). Therefore, it must be that v is either within a^{m^2} or within b^m. The same holds for y. Below are the rest of the cases. Notice that in all cases we obtain a contradiction, and therefore the language L is not context-free. The most important case is (i).

(i) v is within a^{m^2} and y is within b^m.

We have that $v = a^k$ and $y = b^l$, with $1 \leq k+l \leq m$ (since $|vxy| \leq m$ and $|vy| \geq 1$).

Consider the case where $l \geq 1$. From the pumping lemma we have that $uv^0xy^0z \in L$. Therefore, $a^{m^2-k} b^{m-l} \in L$, and thus, it must be that $m^2 - k \leq (m-l)^2$. However, this is impossible since:

$$\begin{aligned} (m-l)^2 &\leq (m-1)^2 \quad (\text{since } l \geq 1) \\ &= m^2 - 2m + 1 \\ &< m^2 - k \quad (\text{since } k \leq m). \end{aligned}$$

Consider now the case where $l = 0$. It must be that $k \geq 1$ (since $k + l \geq 1$). From the pumping lemma we have that $uv^2xy^2z \in L$. Therefore, $a^{m^2+k} b^m \in L$, which is impossible since $m^2 + k > m^2$.

(ii) v and y are within a^{m^2}.

If we pump up v and y (repeat them) we obtain a string of the form $a^{m^2+k} b^m$, with $k \geq 1$, which obviously is not in the language.

(iii) v and y are within b^m.

If we pump down v and y (remove them) we obtain a string of the form $a^{m^2}b^{m-k}$, with $k \geq 1$, which obviously is not in the language.

(c) $L = \{a^n b^j c^k : k = jn\}$

Assume for contradiction that L is a context-free language. We apply the pumping lemma. Let m be the parameter of the pumping lemma. We choose to pump the string $a^m b^m c^{m^2} \in L$. We have that $a^m b^m c^{m^2} = uvxyz$, with $|vxy| \leq m$ and $|vy| \geq 1$.

We examine all the possible cases for the position of string vxy. First we note that the string v cannot span simultaneously both a^m and b^m, since if we pump up v (repeat v), the resulting string is not in the language (a's are mixed with b's). Similarly, v cannot span both b^m and c^{m^2}. Therefore, it must be that v is either within a^m or b^m or c^{m^2}. The same holds for y. Below are the rest of the cases. Notice that in all cases we obtain a contradiction, and therefore the language L is not context-free. The most important case is (i).

(i) v is within b^m and y is within c^{m^2}.

We have that $v = b^k$ and $y = c^l$, with $1 \leq k + l \leq m$ (since $|vxy| \leq m$ and $|vy| \geq 1$).

Consider the case where $k \geq 1$. It must be that $l < m$ (since $k + l \leq m$). From the pumping lemma we have that $uv^0xy^0z \in L$. Therefore, $a^m b^{m-k} c^{m^2-l} \in L$, and thus, it must be that $m \cdot (m - k) = m^2 - l$. However, this is impossible since:

$$\begin{aligned} m \cdot (m-k) &= m^2 - mk \\ &\leq m^2 - m \quad \text{(since } k \geq 1) \\ &< m^2 - l \quad \text{(since } l < m). \end{aligned}$$

Consider now the case where $k = 0$. It must be that $l \geq 1$ (since $k + l \geq 1$). From the pumping lemma we have that $uv^0xy^0z \in L$. Therefore, $a^m b^m c^{m^2-l} \in L$, which is impossible since $m \cdot m \neq m^2 - l$.

(ii) v and y are within b^m.

If we pump down v and y we obtain a string of the form $a^m b^{m-k} c^{m^2}$, with $k \geq 1$, which obviously is not in the language.

(iii) v and y are within c^{m^2}.

If we pump down v and y we obtain a string of the form $a^m b^m c^{m^2-k}$, with $k \geq 1$, which obviously is not in the language.

(iv) v and y are somewhere within $a^m b^m$.

Similar to cases (ii) and (iii).

(d) $L = \{w : n_a(w) < n_b(w) < n_c(w)\}$

Assume for contradiction that L is a context-free language. We apply the pumping lemma. Let m be the parameter of the pumping lemma. We choose to pump the string $a^m b^{m+1} c^{m+2} \in L$. We have that $a^m b^{m+1} c^{m+2} = uvxyz$, with $|vxy| \leq m$ and $|vy| \geq 1$.

We examine all the possible cases for the position of string vxy. Notice that in all cases we obtain a contradiction, and therefore the language L is not context-free. The most important case is (i).

(i) v is within a^m, and y is within b^{m+1}.

We have that $v = a^k$ and $y = b^l$, with $1 \leq k + l \leq m$ (since $|vxy| \leq m$ and $|vy| \geq 1$). From the pumping lemma we have that $uv^3xy^3z \in L$. Therefore, $a^{m+2k} b^{m+2l+1} c^{m+2} \in L$. Since $k + l \geq 1$, it must be that either $k \geq 1$ or $l \geq 1$. If $k \geq 1$ then $m + 2k \geq m + 2$ and therefore $a^{m+2k} b^{m+2l+1} c^{m+2} \notin L$, a contradiction. If $l \geq 1$ then $m + 2l + 1 > m + 2$ and therefore $a^{m+2k} b^{m+2l+1} c^{m+2} \notin L$, a contradiction.

(ii) v is within a^m, and y spans a^m and b^{m+1}.

Similar to case (i).

(iii) v spans a^m and b^{m+1}, and y is within b^{m+1}.

Similar to case (i).

(iv) v and y are within a^m.

If we pump up v and y we obtain a string of the form $a^{m+k} b^{m+1} c^{m+2}$, with $k \geq 1$, which obviously is not in the language.

(v) v and y are within b^{m+1}.

If we pump up v and y we obtain a string of the form $a^m b^{m+k+1} c^{m+2}$, with $k \geq 1$, which obviously is not in the language.

(vi) v and y are somewhere within $b^{m+1} c^{m+2}$.

The analysis is similar with the above cases.

Problem 2.

(a) $L = \{a^n w w^R a^n : n \geq 0, w \in \{a, b\}^*\}$

Context-free.

The following context-free grammar generates the language.

$$S \to aSa \mid X$$
$$X \to aXa \mid bXb \mid \lambda$$

(b) $L = \{a^n b^j a^n b^j : n, j \geq 0\}$

Not context-free.

We can prove this with the pumping lemma. We choose to pump the string $a^m b^m a^m b^m$, where m is the parameter of the pumping lemma. If v is within the first a^m and y is within the first b^m, when we pump up v and y once (repeat them once) the resulting string has the form $a^{m+k} b^{m+l} a^m b^m$, which obviously is not in the language. Similarly, we can see that for any possible position of string vxy the resulting pumped string is not in the language.

(c) $L = \{a^n b^j a^j b^n : n, j \geq 0\}$
Context-free.
The following context-free grammar generates the language.

$$S \rightarrow aSb \mid X$$
$$X \rightarrow bXa \mid \lambda$$

(d) $L = \{a^n b^n c^j : n \leq j\}$
Not context-free.
We can prove this with the pumping lemma. We choose to pump the string $a^m b^m c^m$, where m is the parameter of the pumping lemma. If v is within b^m and y is within c^m, when we pump down v and y (remove them) the resulting string has the form $a^m b^{m-k} c^{m-l}$, which obviously is not in the language. Similarly, we can see that for any possible position of string vxy the resulting pumped string is not in the language.

Problem 3.

We want to prove that the family of context-free languages is closed under reversal. Namely, if L is a context free language, we want to prove that L^R is also a context-free language.

Let G be the context-free grammar that generates the language L. We construct from G a new grammar G' as follows. For every production $X \rightarrow v$ of G, we add the production $X \rightarrow v^R$ in G', where X is a variable, and v is a string of terminals and variables. It is easy to see that that a string w is generated by grammar G if and only if the string w^R is generated by grammar G'. Therefore, grammar G' generates the language L^R, and thus the language L^R is context-free.

Problem 4.

The solution of this problem is very similar to Example 8.7, page 224.

For the discussion below consider the alphabet $\{a, b\}$. We want to prove that the following language, is context-free.

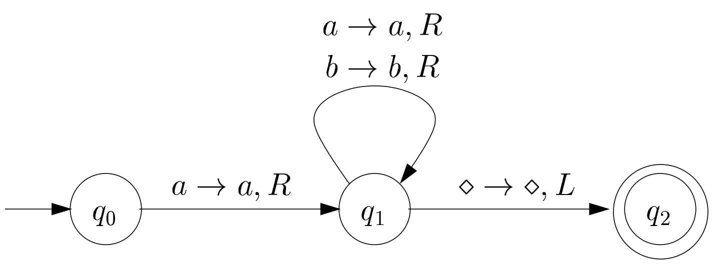
$$L = \{w : n_a(w) = n_b(w),\ w \text{ does not contain the substring } aab\}$$

Consider the following languages.

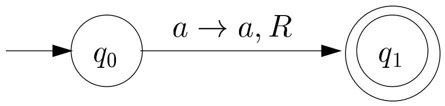
$$L_1 = \{w : n_a(w) = n_b(w)\}$$
$$L_2 = \{w :\ w \text{ contains the substring } aab\}$$

We have that $L = L_1 \cap \overline{L_2}$. We know that language L_1 is context-free (see the grammar of Example 1.12, page 23). It is easy to see that language L_2 is regular, since we can easily construct a nondeterministic finite automaton that accepts all strings which contain substring aab. We know that regular languages are closed under complement (Theorem 4.1, page 103), and therefore the language $\overline{L_2}$ is regular. From Theorem 8.5, page 223, we know that the intersection of a context-free language and a regular language is context-free. Therefore, the language L is context-free.

Problem 5.



(a)



(b)

Figure 1: Turing machines which accept the language $a(a+b)^*$

In Figure 1, we depict two different Turing machines which accept the language $a(a+b)^*$.

The machine of Figure 1.a has three states. The starting state is q_0. The machine reads the first symbol and if it is a the machine goes to state q_1. State q_1 consumes the rest of the input string, and then the machine enters the final state q_2, where it accepts the input.

The machine of Figure 1.b has only two states. The starting state is q_0. The machine reads the first symbol and if this symbol is a it goes to final state q_1, where it accepts the input. Notice that according to the definition of a Turing machine it is not necessary to read all the input string before the machine accepts the string. In the specific machine, by reading only the first symbol we can tell if the input string is in the language or not (if the string starts with an a it is in the language).

Therefore, it is possible to design a Turing machine with only two states that accepts the language.