Computability and Complexity

Exam

Regular and context-free languages

Guidelines: read before anything!

This exam is divided in three sections, and will give you two grades: one for regular languages and one for context-free languages.

1. All exercises in Section 1 will count towards your grade for regular languages;
2. All exercises in Section 3 will count towards your grade for context-free languages;
3. The exercise in Section 2 will count towards both grades.

You can do any exercises as long as you are sure to get a passing grade for each part.

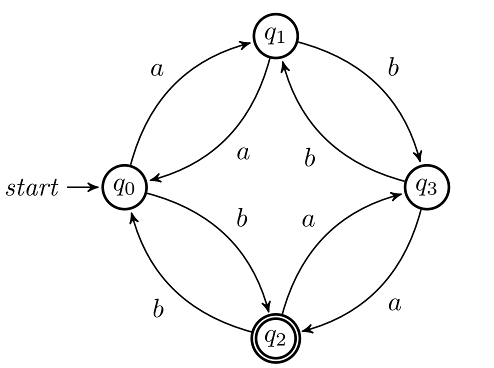
1 Regular languages

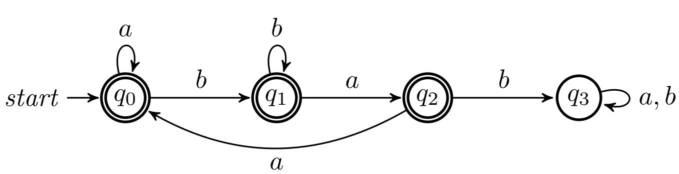
Exercise 1 : ***Warmup (3 points).***
For each of the following languages, construct a finite automata that recognizes them:

1. $\{w \in \{a, b\}^* \mid w$ contains an even number of a and an odd number of $b\}$;
2. $\{w \in \{a, b\}^* \mid w$ does not contain $bab\}$.

<u>*Answer*</u>*:*

1. *In the following automata, q_0 is the state where there is an even number of both a's and b's, q_1 is the state where there is an odd number of a's and an even number of b's, q_2 is the state where there is an even number of a's and an odd number of b's, and q_3 is the state where there is an odd number of both a's and b's:*





2.

□

Exercise 2 : ***Opening by closing (2 points).***
In the next questions, assume that all languages are over a common alphabet Σ.

1. Prove that the class of regular languages is closed under difference (*i.e.*, if L_1 and L_2 are regular, then $L_1 \setminus L_2$ is regular).
2. Deduce from the previous result that the class of regular languages is closed under complementation.

Answer:

1. *Let $A_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $A_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be two nondeterministic finite automatas that recognize, respectively, L_1 and L_2. We construct $A = (Q, \Sigma, \delta, q_0, F)$ that recognizes $L_1 \setminus L_2$:*
   - *$Q = Q_1 \times Q_2$;*
   - *$\delta((q, r), a) = (\delta(q, a), \delta(r, a))$;*
   - *$q_0 = (q_1, q_2)$;*
   - *$F = (q, r)$ such that $q \in F_1$ and $r \notin F_2$.*
2. *We know that $\overline{L} = \Sigma^* \setminus L$. Since Σ^* and L are both regular, by the above observation, $\overline{L}$ is regular. Alternatively, given an automata $(Q, \Sigma, \delta, q_0, F)$ that recognizes L, the automata $(Q, \Sigma, \delta, q_0, Q \setminus F)$ recognizes $\overline{L}$.*

□

Exercise 3 : ***Finition (2 points).***
Prove that if L is a finite language, then L is regular.

Answer: Let $L = \{w_1, \ldots, w_n\}$ with $w_i \in \Sigma^*$. For every i, we denote $w_i = w_i^1 w_i^2 \ldots w_i^{|w_i|}$. We construct a finite automata $A = (Q, \Sigma, \delta, q_0, F)$ that recognizes L.
We construct $\sum_{i=1}^{k} |w_i|$ states, that we denote by q_i^j with $j \in \{1, \ldots, |w_i|\}$, and let

$$Q = \bigcup_{i=1}^{k} \{q_i^j \mid j \in \{1, \ldots, |w_i|\}\} \cup \{q_0\}$$

and

$$F = \bigcup_{i=1}^{k} \{q_i^{|w_i|}\}.$$

As for the transition function, for every i and for every $j \in \{1, \ldots, |w_i| - 1\}$, we let $\delta(q_i^j, w_i^j) = q_i^{j+1}$. It is easy to verify that every word from L will be accepted by A: when reading the input w_i, the automata will follow the states $w_i^1, w_i^2, \ldots$ until reaching the final state $w_i^{|w_i|}$ and accepting the word. Thus, L is recognized by a (nondeterministic) finite automata, which implies it is regular.

□

Exercise 4 : ***Mr. Pump (2 points).***
Prove that the following languages are not regular:

1. $\{ww \mid w \in \{a, b\}^*\}$;
2. $\{a^{2^n} \mid n \geq 0\}$ (the chains of 2^n consecutive a's for any nonnegative integer n).

Answer:

1. *Assume by contradiction that L is regular. We apply the pumping lemma. Let p be the pumping length of L. We consider $w = a^p b a^p b = xyz$. Since $y = a^k$, we have $xy^2z = a^{p+k} b a^p b \notin L$, a contradiction.*
2. *Assume by contradiction that L is regular. We apply the pumping lemma. Let p be the pumping length of L. We consider $w = a^{2^p} = xyz$. We have $y = a^k$, with $1 \leq k \leq p$, and as such $xy^2z = a^{2^p+k}$. However, $2^p < 2^p + 1 \leq 2^p + k \leq 2^p + p < 2^{p+1}$ since $p < 2^p$. Thus, $xy^2z \notin L$, a contradiction.*

□

2 Transition

Exercise 5 : ***An irregular with no context (3 points).***
Let $L = \{a^m b^n \mid m \neq n;\ m, n \geq 0\}$.

1. Prove that L is not regular.

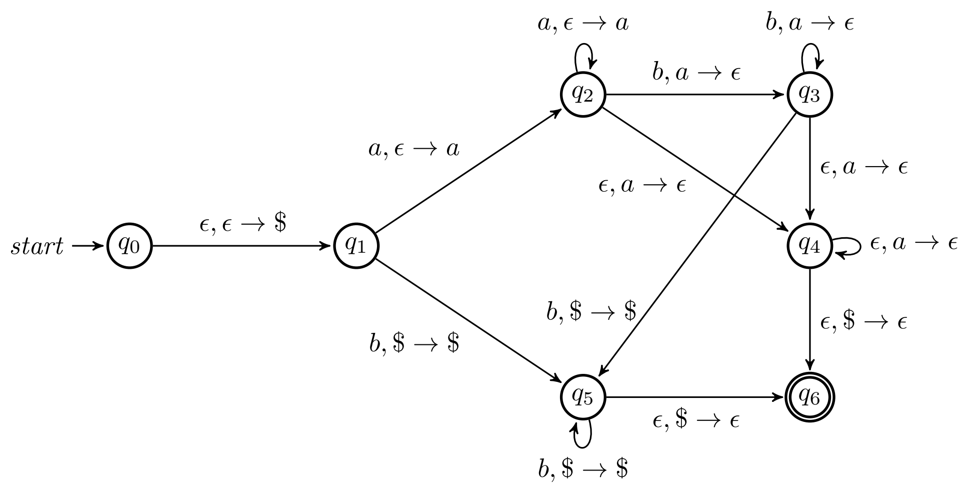
2. Prove that L is context-free by constructing a pushdown automata that recognizes it.

<u>Answer:</u>

1. *Assume by contradiction that L is regular, we use the pumping lemma. Let p be the pumping length of L. Let $w = a^p b^{p!+p} = xyz$. Since $|y| = k \in \{1, \ldots, p\}$, we have that k divides $p!$. Now, consider the word $w' = xy^{k(\frac{p!}{k}+1)}z$. By definition, $x = a^{p-k}$ and $y = a^k$, which implies that $xy^{k(\frac{p!}{k}+1)} = a^{p-k}a^{p!+k} = a^{p!+p}$, but then $w' \notin L$, a contradiction.*

   Alternatively, let $L_0 = \{a^n b^n \mid n \geq 0\}$. It is easy to see that $\overline{L_0} = \{w \mid ab \in w \text{ or } ba \in w\} \cup L$. By Exercise 2, $\overline{L_0}$ cannot be regular (since otherwise, L_0 would be regular). Now, it is easy to construct a finite automata that recognizes $\{w \mid ab \in w \text{ or } ba \in w\}$. Hence, if L was regular, then $\overline{L_0}$ would be the union of two regular languages, and thus would be regular. So L cannot be regular.

2. *In the following automata, if there are no a then we will go through q_5; if there are no b then we will go through q_2 and q_4; if there are more a than b then we will go through q_2, q_3 and q_4; and if there are more b than a we will go through q_2, q_3 and q_5. Also note that if there are as many a as b, then we will either reach q_6 without having finished reading the input (leading to rejecting it) or be stuck in q_3.*



□

3 Context-free languages

Exercise 6 : ***Elementary grammars (2 points).***
For each of the following languages, give a context-free grammar that generates it:

1. $\{a^m b^{m+n} a^n \mid m, n \geq 0\}$;

2. $\{w \in \{a, b\}^* \mid w \text{ contains more } a\text{'s than } b\text{'s}\}$;

3. The complement of $\{a^n b^n \mid n \geq 0\}$.

<u>Answer:</u>

1. $S \to \epsilon \mid AB$
   $A \to \epsilon \mid aAb$
   $B \to \epsilon \mid bBa$

2. $S \to TaT$
   $T \to \epsilon \mid a \mid TT \mid aTb \mid bTa$

3. *It is easy to see that this language is the union of three languages: $a^m b^n$ with $m > n$ or $m < n$, and any word with a ba. So we have to generate those languages:*

   $S \to A \mid B \mid RbaR$
   $A \to a \mid aA \mid aAb$
   $B \to b \mid Bb \mid aBb$
   $R \to \epsilon \mid aR \mid bR$

□

Exercise 7 : *Regulars don't have context (3 points).*
Prove that every regular language is context-free, by showing how to convert a regular expression to an equivalent context-free grammar.
Answer: *Let e be a regular expression over an alphabet Σ. We use induction on its number of symbols. If e contains 1 symbol a, then it is equivalent to the context-free grammar with rule $S \to a$. If e contains 2 or 3 symbols, then it is of the form a^*, $a \cup b$ or ab. For each of those, we can construct the rules of a context-free grammar generating them: $S \to \epsilon \mid aS$ for a^*, $S \to a \mid b$ for $a \cup b$ and $S \to ab$ for ab.*
Now let e be a regular expression of length k with $k \geq 4$. There are three possible forms for e, for which we will construct a rule for a context-free grammar generating the same language:

1. *If $e = e_1^*$ for some expression e_1, then by induction hypothesis there is a context-free grammar G_1 equivalent to e_1. Let S_1 be the starting rule of G_1. We add the following rule to G: $S \to \epsilon \mid S_1 S$, and let S be the new starting rule. Now G is equivalent to e.*

2. *If $e = e_1 \cup e_2$ for some expressions e_1 and e_2, then by induction hypothesis there are two context-free grammars G_1 and G_2 equivalent to, respectively, e_1 and e_2. Let S_1 and S_2 be their starting rules. We create G as the union of G_1 and G_2, and adding the following starting rule: $S \to S_1 \mid S_2$. Now G is equivalent to $e_1 \cup e_2$.*

3. *If $e = e_1 e_2$ for some expressions e_1 and e_2, then by induction hypothesis there are two context-free grammars G_1 and G_2 equivalent to, respectively, e_1 and e_2. Let S_1 and S_2 be their starting rules. We create G as the union of G_1 and G_2, and adding the following starting rule: $S \to S_1 S_2$. Now G is equivalent to $e_1 e_2$.*

This concludes the induction.
Thus, we can represent a regular language by a regular expression, that we can then convert into a context-free grammar, which proves that regular languages are also context-free.

□

Exercise 8 : *Pumping in context (2 points).*
Prove that the language $\{ww \mid w \in \{a, b\}^*\}$ is not context-free.
Answer: *Assume by contradiction that L is context-free. We apply the pumping lemma. Let p be the pumping length, and let us consider $s = a^p b^p a^p b^p = uvxyz$. Note that vxy has to contain the middle point of s. Indeed, if vxy is only in the first part of s, then in $w' = uv^2xy^2z$ the first letter after the middle point is necessarily a b, so $w' \notin L$, a contradiction. A symmetric argument proves that vxy cannot be only in the second part of s.*
So, vxy contains the middle part of s. Now, consider $w' = uv^0xy^0z = uxz$. We have removed some a and b in the middle, but not in the edges. Furthermore, we necessarily removed a positive number of letters. Thus, $w' = a^p b^i a^j b^p$ where either $i < p$, $j < p$ or both. This implies that $w' \notin L$, a contradiction.

□

Exercise 9 : *Challenging grammars (4 points).*
For each of the following languages, give a context-free grammar that generates it:

1. $\{w \in \{a, b\}^* \mid w = xy$ with $|x| = |y|$ but $x \neq y\}$;
2. $\{w\#x \mid w^R$ is a substring of $x;\ w, x \in \{a, b\}^*\}$.

Answer:

1. *Let G be the context-free grammar with the following rules:*

$$\begin{array}{rcl} S & \rightarrow & AB \mid BA \\ A & \rightarrow & a \mid aAa \mid aAb \mid bAa \mid bAb \\ B & \rightarrow & b \mid aBa \mid aBb \mid bBa \mid bBb \end{array}$$

It is easy to see that $L(G)$ is comprised of all words that can be written $v_1 \ell v_2 w_1 \ell' w_2$, with $\ell \neq \ell'$ and $|v_1| + |w_2| = |v_2| + |w_1|$. Now, we can arrange the characters in $v_2 w_1$ such that ℓ and ℓ' are in the same position in their respective parts, which are each of length half the total length of the word. This proves that $L(G) = L$, and thus that L is context-free.

2. *This is more convoluted: we create a rule that will ensure that w^R is a substring of x, and a rule that allows us to add whatever we want in x. We also need to make sure that the two substrings are separated by a #.*

$S \rightarrow TX$
$T \rightarrow aTa \mid bTb \mid \#X$
$X \rightarrow \epsilon \mid aX \mid bX$

It is easy to check that, when doing a derivation, we will have w^R in x thanks to the use of the rule T, that they will be separated by # (after which it will be impossible to change w), and that we can add whatever characters we want in x. Also notice that this works even if $w = \epsilon$.

□