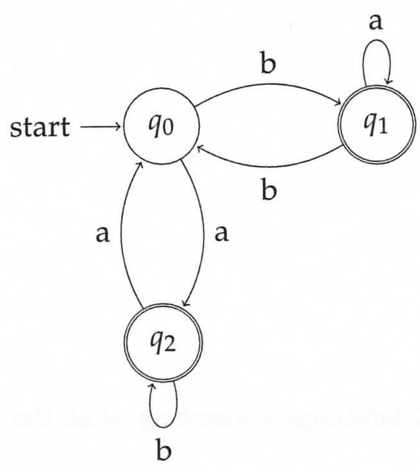
Problem 1 Short answers (16 points, 2 points each)

1) Give a regular expression for the language $L = \{w \in \{a,b,c\}^* \mid \text{the number of } a\text{'s in } w \text{ is a multiple of } 3\}$. For instance $\epsilon \in L$, $abacba \in L$, $aaaaaba \in L$, but $aba \notin L$.

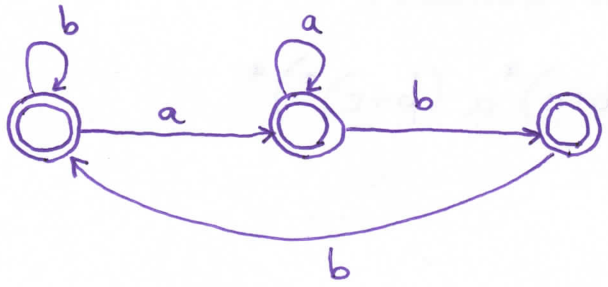
$((b+c)^* a (b+c)^* a (b+c)^* a (b+c)^*)^*$

2) Give a regular expression for the language accepted by the following NFA.



$((ba^*b)^* + (ab^*a))^* (ab^* + ba^*)$

3) Give an NFA having only three states accepting the language $L = \{w \in \{a,b\}^* | w$ does not contain the substring $aba\}$.



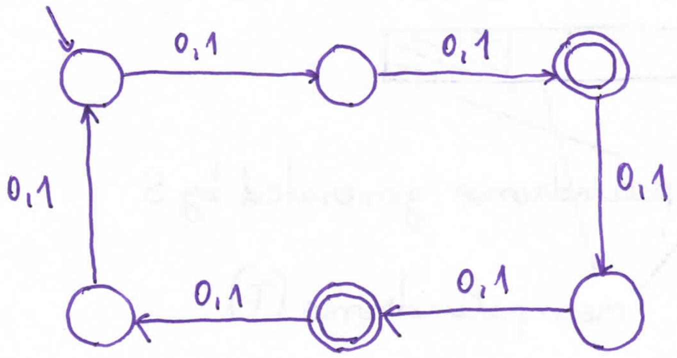
4) Give a context-free grammar for the language $L = \{a^n b^n c^k |\ n \geq 0,\ k \geq 3\}$.

$S \to T\,ccc\,W$
$T \to aTb \mid \varepsilon$
$W \to cW \mid \varepsilon$

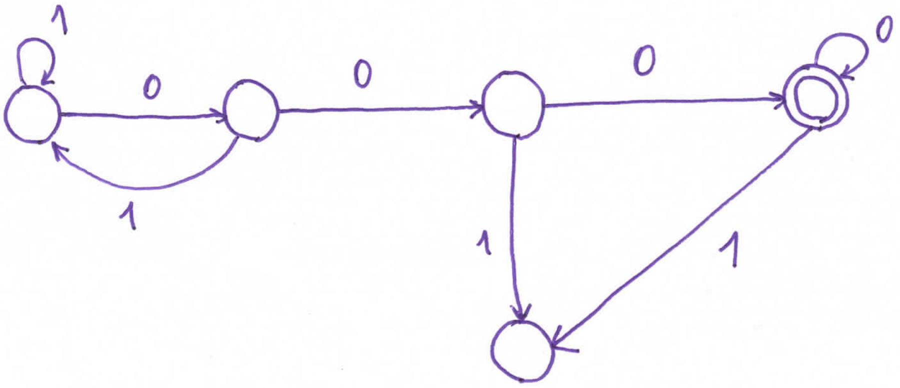
5) Give a regular expression for the language consisting of all the bitstrings over $\{0,1\}$ that have both 00 and 11 as substrings.

$(0+1)^* 00 (0+1)^* 11 (0+1)^* + (0+1)^* 11 (0+1)^* 00 (0+1)^*$

6) Give a DFA accepting the language $L = \{w \in \{0,1\}^* \mid |w| \text{ is divisible by 2 but not by 3}\}$. Here $|w|$ denotes the length of the string w. For example, $|\epsilon| = 0$ and $|01| = 2$.

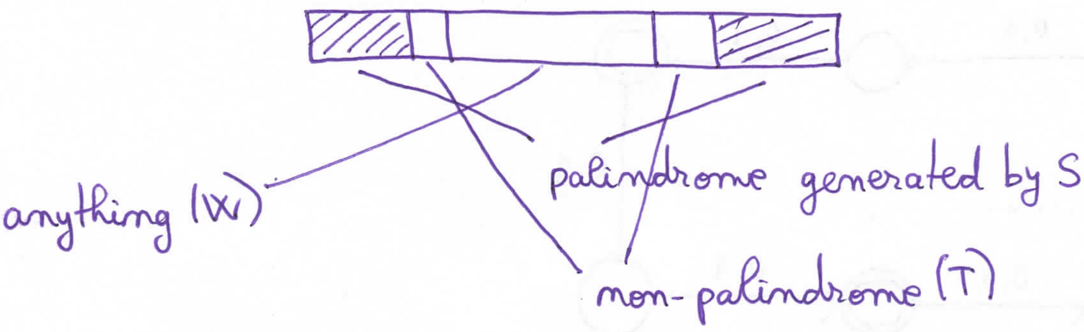


7) Give an NFA accepting the language of all strings over $\{0,1\}^*$ that contain at least one occurrence of the substring 000, but no occurrences of the substring 001.



8) Give a context-free grammar for the language consisting of all the strings in $\{0,1\}^*$ that are not palindromes. Briefly justify.

Any string in the language has the form



$S \to 0S0 \mid 1S1 \mid T$

$T \to 0W1 \mid 1W0 \mid 01 \mid 10$

$W \to 0W0 \mid 0W1 \mid 1W0 \mid 1W1 \mid 1 \mid 0 \mid \varepsilon$

(Blank space for continuing your answers to problem 1).

Problem 2 True/False questions (6 points, 1 point each)

Decide whether each the following statements is true or false. Give a short proof for each statement you believe is true, and give a counter-example to each statement you believe is false.

1) If L is any non-regular language over an alphabet Σ, then the complement of L is also non-regular.

True. Assume that $\bar{L}$ is regular. Then, by the closure properties of regular languages, the complement of $\bar{L}$ is regular. Thus L is regular, contradiction.

2) The class of regular languages is closed under infinite union.

False. The language L_{count} is the infinite union of regular languages $L_m = \{0^m 1^m\}$.

$$L_{count} = \{0^m 1^m \mid m \geq 0\} = \bigcup_{m=0}^{\infty} \{0^m 1^m\}.$$

Each L_m is finite, therefore is regular.

3) The number of regular languages over the alphabet $\Sigma = \{0, 1\}$ is infinite.

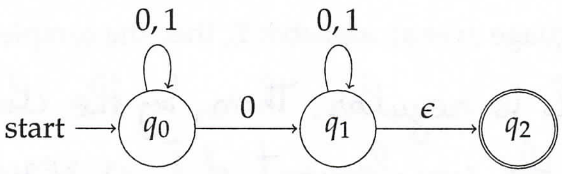
True. Consider the languages

$L_0 = \{\varepsilon\}$,
$L_1 = \{0\}$,
$L_2 = \{00\}$,
...
$L_m = \{0^m\}$
...

Each L_m is finite, therefore is regular.

True/False questions

4) The language of the NFA below is $L = \{w \in \{0,1\}^* \mid w \text{ has odd length and its middle character is } 0\}$



False. The NFA also accepts other strings, for example 0000.

5) If L is a regular language over the alphabet $\{0,1\}$, then the language $\text{Even}(L) = \{w \in L \mid |w| \text{ is even}\}$ is regular. Here $|w|$ denotes the length of the string w. For example, $|\epsilon| = 0$ and $|01| = 2$.

True. $\text{Even}(L) = L \cap L_1$, where $L_1 = \{w \in \{0,1\}^* \mid |w| \text{ is even}\}$.
L_1 is regular, so Even (L) is the intersection of two regular languages, therefore it is regular.

6) The regular expression $0(120)^*12$ generates the same language as the regular expression $01(201)^*2$.

True. $\underbrace{0(120)(120)\ldots(120)}_{k \text{ times}}12 = 01\underbrace{(201)(201)\ldots(201)}_{k \text{ times}}2$

Problem 3 DFA and regex (5 points: 3 + 2) We say that two bitstrings have the same 1-parity if and only if both strings contain an odd number of 1s or both strings contain an even number of 1s. We define over the alphabet $\Sigma = \{0, 1, \#\}$ the following language

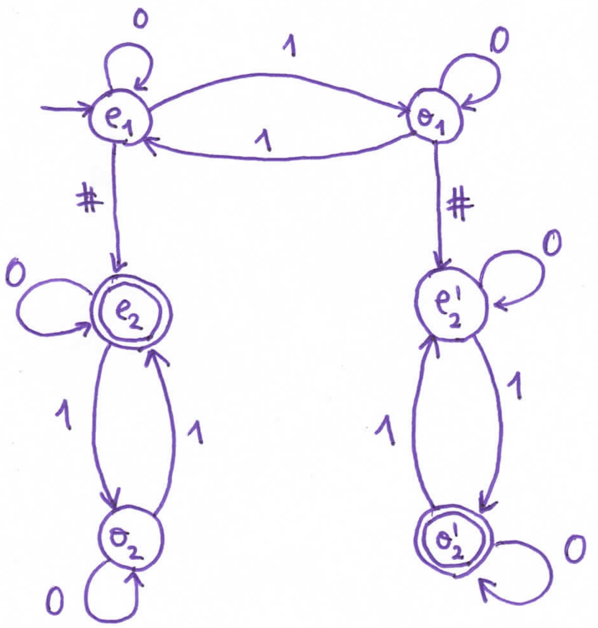
$$1\text{PAR} = \{u\#v | u, v \in \{0,1\}^* \text{ and } u \text{ and } v \text{ have the same 1-parity}\}$$

For example $\# \in$ 1PAR, $01\#111 \in$ 1PAR, and $0011\#11001010 \in$ 1PAR. However $1\#0 \notin$ 1PAR, $\#\# \notin$ 1PAR, and $\epsilon \notin$ 1PAR.

a) Describe a DFA for 1PAR. A formal description or a drawing without an English explanation will receive no credit, even if it is correct.

b) Write a regular expression for 1PAR.

a) We extend the DFAs which recognize strings of odd (or even) length.



e_2 accepts strings even # even, and o'_2 accepts strings odd # odd. The #-transitions omitted from the figure go to a trap state.

b) $(0^*10^*10^*)^*\#(0^*10^*10^*) + 0^*10^*(0^*10^*10^*)^*\#0^*10^*(0^*10^*10^*)^*$

Problem 4 Constructive proof (4 points: 1 + 3).
If $x = a_1a_2\dots a_n$ and $y = b_1b_2\dots b_n$ are two strings of the same length n, define $\mathrm{alt}(x,y)$ to be the string in which the symbols of x and y alternate, starting with the first symbol of x, that is,

$$\mathrm{alt}(x,y) = a_1b_1a_2b_2\dots a_nb_n.$$

If L and M are languages, define $\mathrm{alt}(L,M)$ to be the language of all strings of the form $\mathrm{alt}(x,y)$, where x is any string in L and y is any string in M of the same length.

a) Write a regular expression for $\mathrm{alt}(0^*1, 01^*)$.

b) If L and M are regular languages, prove that $\mathrm{alt}(L,M)$ is regular.

a) $\mathrm{alt}(0^m1, 01^m)$, $m \geq 0$

$\mathrm{alt}(1,0) = 10$

$\mathrm{alt}(01,01) = 0011$

$\mathrm{alt}(001, 0\cancel{1}1) = 000111 = 00(01)^1 11$

$\mathrm{alt}(0001, 0111) = 00010111 = 00(01)^2 11$

$\mathrm{alt}(0^m1, 01^m) = 00(01)^{m-1}11.$

Taking the union over $m = 0, 1, \dots, \infty$, we obtain the result

$$10 + 00(01)^*11.$$

b) Let D_1 be a DFA for L and D_2 be a DFA for M. We construct an NFA N for $\mathrm{alt}(L,M)$.

The states of N have the form (q_1, q_2, b), where q_1 is a state of D_1, q_2 is a state of D_2, and b is either 0 or 1. If $b=0$, we take a transition according to D_1 and make $b=1$. Likewise, if $b=1$, we take a transition according to D_2 and make $b=0$. The final states of N are $(f_1, f_2, 0)$, where f_1 is a final state of D_1 and f_2 is a final state of D_2.

(Blank page for continuing your answers to problem 4).

Formally, if $D_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ and $D_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$, we construct an NFA $N = (Q, \Sigma, q_0, \delta, F)$, where

$$Q = Q_1 \times Q_2 \times \{0,1\},$$

$$q_0 = (q_{01}, q_{02}, 0),$$

$$F = F_1 \times F_2 \times \{0\},$$

$$\delta((q_1, q_2, b), \sigma) = \begin{cases} (\delta_1(q_1,\sigma), q_2, 1) & \text{if } b = 0 \\ (q_1, \delta_2(q_2,\sigma), 0) & \text{if } b = 1 \end{cases}$$

Problem 5 CFG design (3 bonus points).

This problem is intended to be a challenge.

Let $D = \{xy \mid x, y \in \{0,1\}^* \text{ and } |x| = |y| \text{ but } x \neq y\}$. Give a context-free grammar for D.

Let $\Sigma = \{0,1\}$. We denote by Σ^k any bitstring of length k. Any string from D has the form

$$\Sigma^k 0 \Sigma^j \Sigma^k 1 \Sigma^j \quad \text{or} \quad \Sigma^k 1 \Sigma^j \Sigma^k 0 \Sigma^j,$$

for some $k, j \geq 0$. The content of the strings denoted above with Σ^j or Σ^k does not matter, only their length. Thus, instead of

$$\Sigma^k 0 \Sigma^j \Sigma^k 1 \Sigma^j$$

we can generate

$$\Sigma^k 0 \Sigma^k \Sigma^j 1 \Sigma^j.$$

We obtain the following grammar

$$S \to S_1 S_2 \mid S_2 S_1$$
$$S_1 \to 0S_10 \mid 0S_11 \mid 1S_10 \mid 1S_11 \mid 0$$
$$S_2 \to 0S_20 \mid 0S_21 \mid 1S_20 \mid 1S_21 \mid 1$$

The non-terminals S_1 and S_2 generate the two parts of the string of equal sizes. The only way to eliminate them is to replace S_1 with 0 and S_2 with 1. This forces a difference in the two parts of the string we generated.