

Exercise 1 (Yes/No):**(0 Points)**

1. Deterministic finite automata (DFA) are strictly less expressive than regular expressions.
2. Non-deterministic finite automata (NFA) are strictly more expressive than deterministic finite automata.
3. The languages of regular expressions are closed under...
 - a) union,
 - b) intersection,
 - c) complement,
 - d) concatenation,
 - e) Kleene closure.
4. Context Free Languages (CFL) are closed under...
 - a) union,
 - b) intersection,
 - c) complement,
 - d) concatenation,
 - e) Kleene closure.
5. DCFL is the set of context free languages that are accepted by deterministic push down automata. Is DCFL = CFL?

Lösung:

1. No. One can be effectively converted to another.
 2. No. NFA can be converted to a DFA using the powerset construction (Rabin-Scott construction).
 3. Yes. Each operation can implemented by manipulating a NFA.
 4. Context Free Languages (CFL) is closed under the following.
 - a) Yes. S_1 and S_2 are starting Non-terminal for two Grammar. $S \rightarrow S_1 \mid S_2$.
 - b) No. $L_1 = \{a^i b^j c^j \mid i, j \in \mathbb{N}\}$ and $L_2 = \{a^i b^j c^j \mid i, j \in \mathbb{N}\}$
 - c) No. From above.
 - d) Yes. $S \rightarrow S_1 S_2$.
 - e) Yes. Let S be the starting non-terminal. New Grammar $S' \rightarrow SS' \mid \epsilon$.
 5. Just an intuitive example: Palindrome
-

Exercise 2 (Regular Expressions and Grammars):**(0 Points)**

- Describe the language of the following regular expression in words:

$$r = (0 + 1)^*0(0 + 1)^*0(0 + 1)^*.$$

- Construct the regular expression for...

- the set of all strings with at most one pair of consecutive 0's and at the most one pair of consecutive 1's,
- the set of all strings with equal number of 0s and 1s such that no prefix has two more 0s than 1s nor two more 1s than 0s.

- Construct a context free grammar (CFG) for a set of strings of $\{(,)\}^*$ such that every string of the set has equal number of left and right parenthesis, and every prefix has at least as many left parenthesis as right parenthesis.

Lösung:

- The set of all strings containing at least two 0's.

- a)

$$(0 + \epsilon)(10)^*(0 + \epsilon)(10)^*(1 + \epsilon)(10)^*(1 + \epsilon) \\ + (1 + \epsilon)(01)^*(1 + \epsilon)(01)^*(0 + \epsilon)(01)^*(0 + \epsilon)$$

- b)

$$((01)^* + (10)^*)^*$$

-

$$S \rightarrow (S) S \mid \epsilon$$

Exercise 3 (Proofs):**(0 Points)**

- Let r and s be regular expressions. Consider the equation $X = r.X + s$. Under the assumption that the language of r ($L(r)$) does not contain ϵ , find X and show that it is unique. What happens if $\epsilon \in L(r)$?
- One can construct a regular expression from a finite automaton by solving a system of linear equation of the form

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & \dots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$$

Given an NFA provide suitable definition of the vectors above. It might be helpful to assume an annihilator ϕ for regular expression, i.e., $\alpha.\phi = \phi, \phi.\alpha = \phi$ and $\phi + \alpha = \alpha$

3. a) An equivalence relation $R \subseteq \Sigma^* \times \Sigma^*$ is said to be *right invariant* when,

$$\forall z \in \Sigma^* \quad (x, y) \in R \iff (xz, yz) \in R.$$

Show that if $L \subseteq \Sigma^*$ is regular then it is the union of some of the equivalence classes of R .

- b) Let R_L be an relation defined as: $(x, y) \in R_L$ if and only if for all $z \in \Sigma^*$ $xz \in L$ exactly when $yz \in L$. Show that R_L is an equivalence relation. Show that number of equivalence classes of R_L is finite if and only if L is regular.
4. a) Show that the language $L = \{0^{i^2} \mid i \in \mathbb{N}\}$ is not regular.
- b) Show that the language $L = \{a^i b^j c^i \mid i \in \mathbb{N}\}$ is not a CFL.

Lösung:

1. Guess the solution to be r^*s (because it feels right).

$$X = r.X + s \tag{1}$$

By substituting r^*s for X in the equation (1), we observe that r^*s satisfies this property. Thus, $L(r^*s) \subseteq L(X)$. Translating the equation to the topology of strings,

$$X = R.X \cup S$$

where $R = L(r)$ and $S = L(s)$. Let $\epsilon \notin L(r)$. Suppose the solution of the above equation is $X = R^*S \cup C$, where $R^*S \cap C = \emptyset$. This new set must also satisfy the equation. Thus, $R^*S \cup C = R^*S \cup RC$. Since $R^*S \cap C = \emptyset$, $C \subseteq RC$. Now we can see why the condition $\epsilon \notin R$ is important, mainly $C \subseteq RC$ has no solution for C when $\epsilon \notin R$.

The proof also shows what happens if $\epsilon \in R$, $r^*s + c$ is a solution of equation (1) for any c .

2. Let $M = (S, s_0, \Sigma, \delta, F)$ be a NFA. x_i is the regular expression for the set of strings that are accepted by automaton M is s_i is the starting state. Let $a_{ij} = b_1 + \dots + b_k$ is defined such that $s_j = \delta(s_i, b_l)$ for all $l \in [1, k]$, if no transition exists from state s_i to s_j then $a_{ij} = \emptyset$. $c_i = \epsilon$ if $s_i \in F$ else it is $\emptyset$.

3. a) We need to find a right invariant relation R for a given regular language. There exists a NFA $M = (S, s_0, \Sigma, \delta, F)$ such that $L(M) = L$. Define $xRy \iff \delta(s_0, x) = \delta(s_0, y)$. Observe that it is a right invariant equivalence relation. $L = \bigcup_{\delta(s_0, x) \in F} [x]$.

- b) Show R_L is reflexive, symmetric and transitive.

$\Leftarrow$ The equivalence classes of R_L , $S = \{[x_1], [x_2], \dots, [x_n]\}$. Observe, if $x \in L$ then $[x] \subseteq L$, thus L is the union of some of the equivalence classes of R_L . Define an automaton, $M = (S, [\epsilon], \Sigma, \delta, F)$ where $\delta([x], a) = [xa]$. The set of final states F is defined such that $L = \bigcup_{\delta(s_0, x) \in F} [x]$. Easy to see

$L(M) = L$. (Good time to say that this construction gives canonical (the smallest) DFA for a regular language)

$\Rightarrow$ L is a regular set and $M = (S, s_0, \Sigma, \delta, F)$ is the NFA such that $L(M) = L$. Define R_M as $xR_My \iff \forall z, \delta(s_0, x) = \delta(s_0, y)$. Note that $R_M \leq R_L$, that is, R_M is a refinement of R_L . This follows from the implication, if xR_My then xR_Ly . As we know that the number of equivalence class of R_M is finite (equal to the number of states of M), the number of equivalence classes R_L is finite as well.

4. a) Let L be regular and n be the integer in the pumping lemma. Let $z = 0^{n^2}$. By pumping lemma $z = uvw$ with $v \geq 1$, $|uv| \leq n$ and $uv^i w$ is in L for all i . for $i = 2$, $n^2 < |uv^2 w| \leq n^2 + n$. Or $|uv^2 w|$ is strictly between n^2 and $(n+1)^2$.
- b) Assume L is defined by a CFG. Let n be the pumping integer. We choose a string $z = a^n b^n c^n$. We write $z = uvwx y$. By the premises of the pumping lemma $|vwx| \leq n$, thus it cannot contain both a and c part of z . So pump away and prove that the resulting string though in the language of the CFL does not belong to L .
-