Homework 3. Foundation. Languages. Automata.

Submit both PDF and LaTeX file of your solution to Canvas by 11:59pm Friday October 10.

1. You must use LaTeX for your submission.

2. Write a pseudocode algorithm where the input is a regular expression R and the output is the language represented by R.

   Hint: Study the definition of semantics of regular expressions around page 59 of Lecture 3. Your pseudocode algorithm can be based on problem decomposition/divide and conquer using the definition.

   LaTeX sample algorithm format:

Algorithm 1: sort

Input : $A[1..n]$: the array of numbers to sort and n is the number of elements in array A.
output : N.A.
side effect: The numbers in array A is ordered
plan :

```
1 while While condition do
2 |   instructions;
3 |   if condition then
4 |   |   instructions1;
5 |   |   instructions2;
6 |   else
7 |   |   instructions3;
8 |   end
9 end
```

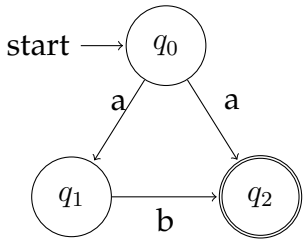
ANSWER:

Algorithm 2: Semantic evaluation $\mathcal{L}(R)$ of a regular expression R

Input : A regular expression R over alphabet Σ
Output: The language $\mathcal{L}(R) \subseteq \Sigma^*$
Plan : Divide-and-conquer on the syntax of R using the semantics of regular expressions

```
1  Function Lang(R):
2      if R = ∅ then
3          return ∅
4      else if R = ε then
5          return {ε}
6      else if R = a for some a ∈ Σ then
7          return {a}
8      else if R = (R1 ∪ R2) then
9          return Lang(R1) ∪ Lang(R2)
10     else if R = (R1 R2) then
11         return { xy | x ∈ Lang(R1) ∧ y ∈ Lang(R2) }
12     else if R = (R1)* then
13         return KleeneClosure(Lang(R1))
14     else
15         return Lang((R')) where R' is the subexpression inside outer parentheses
16     end
17 end
18 Function KleeneClosure(L):
19     return ⋃_{i≥0} L^i where L^0 ≜ {ε} and L^{i+1} ≜ { xy | x ∈ L^i, y ∈ L }
20 end
```

3. Consider the following NFA diagram:



Write a tuple representation of this automaton (using the notation from our definition of NFA).

ANSWER:

Let $N = (\Sigma, K, s, F, \Delta)$ be the NFA, where

$$\Sigma = \{a, b\}, \quad K = \{q_0, q_1, q_2\}, \quad s = q_0, \quad F = \{q_2\}.$$

The transition function $\Delta : K \times \Sigma \to \mathcal{P}(K)$ is

$$\begin{aligned} &\Delta(q_0, a) = \{q_1, q_2\}, \quad \Delta(q_0, b) = \varnothing, \\ &\Delta(q_1, a) = \varnothing, \qquad \Delta(q_1, b) = \{q_2\}, \\ &\Delta(q_2, a) = \varnothing, \qquad \Delta(q_2, b) = \varnothing. \end{aligned}$$

4. The **language** accepted by an NFA N is the set of all strings accepted by the NFA. Write the definition of this concept using set builder notation.

   ANSWER:

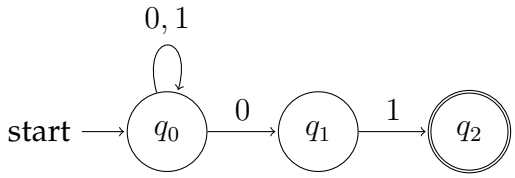
   Let $N = (\Sigma, K, s, F, \Delta)$ and let $\vdash_N^*$ be the reflexive-transitive closure of one-step moves. Then

   $$L(N) \triangleq \{\, w \in \Sigma^* \mid \exists\, q_f \in F \text{ such that } (s, w) \vdash_N^* (q_f, \varepsilon) \,\}.$$

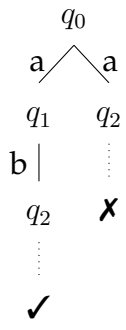
   Equivalently, letting Δ^* be the extended transition function,

   $$L(N) = \{\, w \in \Sigma^* \mid \Delta^*(\{s\}, w) \cap F \neq \varnothing \,\}.$$

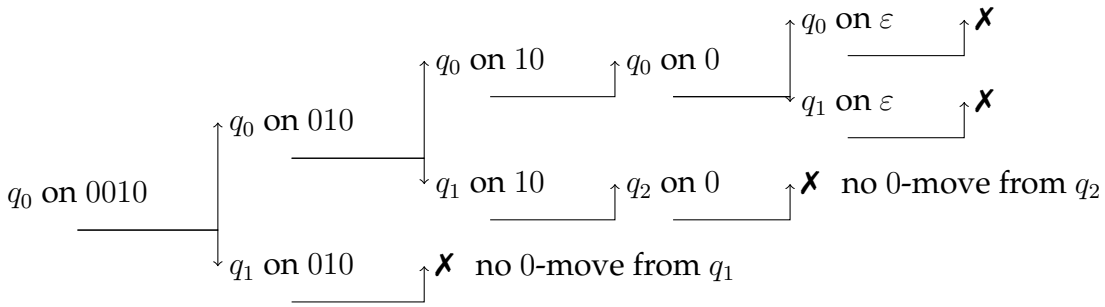
5. Given the automaton diagram below and the string 0010, draw the computation tree for processing 0010 on this automaton.



   LaTeX sample tree format (content is unrelated to the answer):



   ANSWER:



(All leaves reject on 0010, so the string is not accepted.)

6. Consider these definitions:

   A configuration (q, w) **yields in one step** configuration (q', w'), denoted by $(q, w) \vdash_M (q', w')$, if $w = aw'$ and $\delta(q, a) = q'$.

   A configuration (q_1, w_1) **yields** configuration (q_2, w_2), denoted by $(q_1, w_1) \vdash_M^* (q_2, w_2)$, if there is a sequence of configurations, starting with (q_1, w_1) and ending with (q_2, w_2), each of which is yielded in one step from its previous configuration.

   Answer the following: What concepts are being defined? What concepts are used in these definitions? What logical symbols appear? What are the important variables? What is the tree structure of the definition statement for *yields*?

   ANSWER:

   They define (i) the *one-step move* relation $\vdash_M$ and (ii) its *reflexive-transitive closure* $\vdash_M^*$ ("yields") for a machine M. Concepts used: configurations (q, w), the DFA transition function δ, decomposition of an input string $w = aw'$, and finite sequences of configurations. Logical/quantifier symbols: $=$, $\in$, $\exists$ ("there exists a sequence"), indexed conjunction over a sequence ("for each adjacent pair is a one-step move"). Important variables: states q, q', q_1, q_2, strings w, w', w_1, w_2, symbol $a \in \Sigma$, and the machine M. The definition tree for "yields" is:

$$\underbrace{(q_1, w_1) \vdash_M^* (q_2, w_2)}_{\text{goal}} \iff \exists k \geq 0,\ \exists C_0, \ldots, C_k \begin{cases} C_0 = (q_1, w_1), \\ C_k = (q_2, w_2), \\ \forall i \in \{0, \ldots, k-1\} : C_i \vdash_M C_{i+1}. \end{cases}$$

7. Consider an NFA $N = (\Sigma, K, s, F, \Delta)$. Let M be an equivalent DFA for N as defined in Lecture 4.

   Given that $(q_1, aw_1) \vdash_N (q_2, w_1)$ holds under N, prove there exist $S_1, S_2 \in K_M$ such that $(S_1, aw_1) \vdash_M (S_2, w_1)$ holds under M.

   Recall our proof methods including application of the definition of $\vdash_M$, definition of δ of M, and definitions of DFA and NFAs.

   ANSWER (PROOF):

By $(q_1, aw_1) \vdash_N (q_2, w_1)$ we have $q_2 \in \Delta(q_1, a)$ (by the one-step definition for NFAs). In the subset construction, the DFA M has state set $K_M = \mathcal{P}(K)$ and transition function

$$\delta_M(S, a) = \bigcup_{p \in S} \Delta(p, a) \quad \text{for } S \subseteq K,\ a \in \Sigma.$$

Choose $S_1 \triangleq \{q_1\}$ and define $S_2 \triangleq \delta_M(S_1, a) = \bigcup_{p \in \{q_1\}} \Delta(p, a) = \Delta(q_1, a)$. Then $q_2 \in S_2$ by the premise. By the DFA one-step move definition,

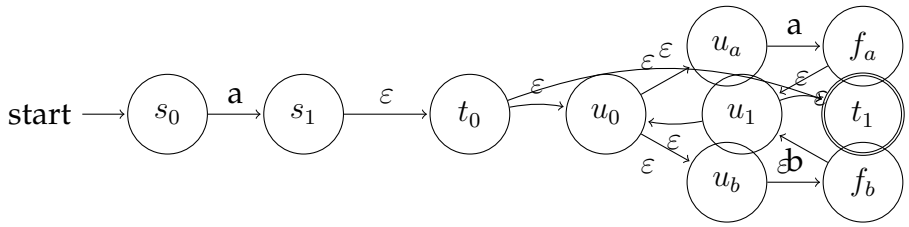
$$(S_1, aw_1) \vdash_M (\delta_M(S_1, a), w_1) = (S_2, w_1).$$

Thus there exist $S_1, S_2 \in K_M$ with $(S_1, aw_1) \vdash_M (S_2, w_1)$, as required.

8. Using the method discussed in class, construct an NFA for the regular expression $(\!|a(\!|a \uplus b|\!)^*|\!)$.

ANSWER:

Below is a Thompson-style ε-NFA for $a\,(a \cup b)^*$.



This NFA accepts exactly $a(a \cup b)^*$.