CS228 Tutorial Solutions

Arpon Basu

March 15, 2023

Contents

1 Tutorial 1

1.1 Q1

Note that FO-definable languages are a (strict) subset of regular languages. Consequently, if a language is FO-definable then it is automatically regular.

(a) The language is $a\Sigma^*a+b\Sigma^*b+a+b+\varepsilon$, which is regular and FO definable by the formula $\varphi := \exists x, y.(\forall z.(x \leq z \leq y) \wedge (Q_a(x) \iff Q_a(y))) \vee \forall x.(x \neq x)$, assuming $\Sigma = \{a, b\}$.

(b) The language is $a^*\#b^*$, which is regular and FO definable by the formula $\varphi := \exists x.(Q_\#(x) \wedge \forall y.(y < x \implies Q_a(y)) \wedge \forall y.(x < y \implies Q_b(y)))$.

(c) The language is a^*b^*, which is regular and FO definable by the formula $\varphi := \forall x, y.(S(x, y) \wedge Q_b(x) \implies Q_b(y))$.

(d) The language is $\Sigma 0\Sigma^* 0\Sigma + \Sigma 0\Sigma$, which is regular and FO definable by the formula $\varphi := \exists x.((\forall y.y \geq x) \wedge \exists t.(S(x, t) \wedge Q_0(t))) \wedge \exists x.((\forall y.y \leq x) \wedge \exists t.(S(t, x) \wedge Q_0(t)))$.

(e) Note that our alphabet here is $\Sigma = \{\binom{0}{0}, \binom{0}{1}, \binom{1}{0}, \binom{1}{1}\}$. If the top row is larger than the bottom row, then we must have a $\binom{1}{0}$ somewhere, preceding which all digits should be the same. Thus the language is $(\binom{0}{0}+\binom{1}{1})^*\binom{1}{0}\Sigma^*$, which is FO definable by the formula $\varphi := \exists x.(Q_{\binom{1}{0}}(x) \wedge \forall y.(y < x \implies (Q_{\binom{0}{0}}(y) \vee Q_{\binom{1}{1}}(y))))$.

1.2 Q2

(1)

(a) $\mathcal{L}(\varphi) = \{\varepsilon\}$

(b) $\overline{\mathcal{L}(\varphi)} = \Sigma^* \setminus \{\varepsilon\}$, ie:- the set of all non-empty words.

(c) Yes. Consider a DFA $\mathcal{A}$ with only two states α, β, where α is the start state, and the only accept state. The transitions are $\delta(\alpha, \sigma) = \delta(\beta, \sigma) = \beta$ for every $\sigma \in \Sigma$. $\mathcal{A}$ accepts $\{\varepsilon\}$.

(d) The complement of a regular language is regular, so yes.

(2)

(a) $\mathcal{L}(\varphi) = \Sigma^*(ba^+)\Sigma^*$

(b) $\overline{\mathcal{L}(\varphi)} = a^*b^*$

(c)

(d) The complement of a regular language is regular, so yes.

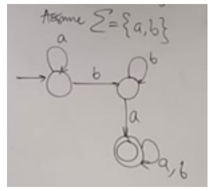

Figure 1: 2(2)(c)

(3)

(a) $\mathcal{L}(\varphi) = \Sigma^* a \Sigma$

(b) $\overline{\mathcal{L}(\varphi)} = \{\varepsilon, a, b\} \cup \Sigma^* b \Sigma$

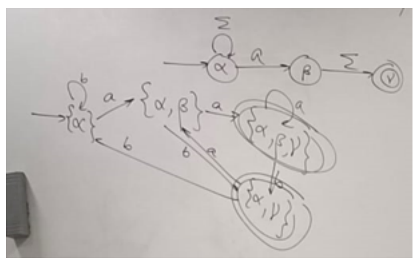

Figure 2: 2(3)(c)

(c)

(d) The complement of a regular language is regular, so yes.

(4)

(a) $\mathcal{L}(\varphi) = (ab)^+$

(b)

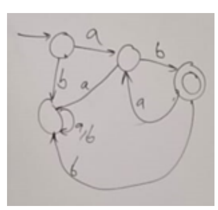

Figure 3: 2(4)(c)

(c)

(d) The complement of a regular language is regular, so yes.

1.3 Q3

The language described by the DFA is $b(a^+b^3)^*$. It is FO describable, with the (conjunction of the) following sentences:-

1. First letter is 'b': $\exists x \forall y (x \leq y \wedge Q_b(x))$
2. If the length of the word is more than 1, then it ends with a "bbb":

   $$(\exists x, y.S(x,y)) \implies \exists x.((\forall t.t \leq x) \wedge Q_b(x) \wedge \exists y.(S(y,x) \wedge Q_b(y) \wedge \exists z.(S(z,y) \wedge Q_b(z))))$$

3. For every "bbb", there is a non-empty block of 'a's before it, preceding which there is a 'b':

   $$\forall x, y, z.((S(x,y) \wedge S(y,z) \wedge Q_b(x) \wedge Q_b(y) \wedge Q_b(z)) \implies ($$

   $$(\exists t.S(t,x) \wedge Q_a(t)) \bigwedge$$

   $$(\exists u.u < x \wedge Q_b(u) \wedge (\forall v.(u < v < x) \implies Q_a(v))))$$

4. We don't have 4 consecutive 'b's:

   $$\neg \exists x, y, z, w.S(x,y) \wedge S(y,z) \wedge S(z,w) \wedge Q_b(x) \wedge Q_b(y) \wedge Q_b(z) \wedge Q_b(w)$$

5. We don't have "aba":

   $$\neg \exists x, y, z.S(x,y) \wedge S(y,z) \wedge Q_a(x) \wedge Q_b(y) \wedge Q_a(z)$$

6. If a 'b' is preceded by an 'a' and succeeded by a 'b', then it is a triplet,

   $$\forall x, y, z.(Q_a(x) \wedge S(x,y) \wedge Q_b(y) \wedge S(y,z) \wedge Q_b(z) \implies \exists t.(S(z,t) \wedge Q_b(t)))$$

2 Tutorial 2

Even though ε-NFAs have not been taught in class, they are very useful in practice. In particular, we shall be using them a lot in our tutorial. Please refer to this link for a formal proof of equivalence (of power) of ε-NFAs, NFAs and DFAs.
We shall represent DFAs as 5-tuples $(Q, q_0, \Sigma, \delta, \mathcal{F})$, where Q denotes the set of states, q_0 is the start state, Σ the alphabet, $\delta : Q \times \Sigma \mapsto Q$ the transition function, and $\mathcal{F}$ is the set of final states.
Similarly, NFAs will be represented as $(Q, Q_0, \Sigma, \delta, \mathcal{F})$, where Q denotes the set of states, Q_0 is the set of start states, Σ the alphabet, $\delta : Q \times \Sigma \mapsto 2^Q$ the transition function, and $\mathcal{F}$ is the set of final states.

2.1 Q1

(a) Yes. Given a DFA $(Q, q_0, \Sigma, \delta, \mathcal{F})$, construct the NFA $(Q, \mathcal{F}, \Sigma, \delta', \{q_0\})$, where $\delta'(i, \sigma) = j$ if and only if $\delta(j, \sigma) = i$, for every $i, j \in Q, \sigma \in \Sigma$, ie:- we reverse all the transitions, and make our final states inital states, and the initial state final. It's easy to see that the NFA only accepts words which are the reverse of some word accepted by the DFA, and thus the reverse of a language is also regular.

(b) Let $\mathcal{A}$ be a DFA representing L. Replace all transitions of a in $\mathcal{A}$ to ε-transitions. The resulting ε-NFA recognizes $L \downarrow \{b, c\}$.

(c) Let our NFA be $(Q, Q_0, \Sigma, \delta, \mathcal{F}), |\mathcal{F}| > 1$. Construct a new NFA $(Q \cup \{f\}, Q_0', \Sigma, \delta', \{f\})$, where, for every $i \in Q$ and every $\sigma \in \Sigma$ we have

$$\delta'(i, \sigma) = \begin{cases} \delta(i, \sigma), \text{ if } \delta(i, \sigma) \cap \mathcal{F} = \emptyset \\ \delta(i, \sigma) \cup \{f\}, \text{ otherwise} \end{cases}$$

$$\delta'(f, \sigma) = \emptyset$$

$$Q_0' = \begin{cases} Q_0, \text{ if } Q_0 \cap \mathcal{F} = \emptyset \\ Q_0 \cup \{f\}, \text{ otherwise} \end{cases}$$

The new NFA recognizes the same language as the old one, but has only one final state.
Challenge:- Demonstrate a regular language L such that any DFA recognizing L has atleast 2 accepting states.

(d) Yes. The proof of this fact is not so difficult.

(e) Refer this.

2.2 Q2

Fix any $n \geq 2$ (for $n = 1$, $L_1 = \Sigma^* \setminus \{\varepsilon\}$, which is regular). We assume that ε is not divisible by any $n \in \mathbb{N}$.
Construct the DFA $\mathcal{A}_n = (Q, \text{start state}, \Sigma, \delta, \mathcal{F}) = (\{s, q_0, q_1, \ldots, q_{n-1}\}, s, \{0, 1\}, \delta, \{q_0\})$, with the transitions being as follows:

$$\delta(s, 0) = q_0, \delta(s, 1) = q_1$$

$$\delta(q_i, 0) = q_{(2i) \bmod n}, \delta(q_i, 1) = q_{(2i+1) \bmod n}$$

We finish the proof that L_n is regular by noting that $\mathcal{L}(\mathcal{A}_n) = L_n$ [1].

[1] Note that in the exam you would be expected to formally argue why $\mathcal{A}_n$ accepts every word in L_n, and doesn't accept any word in $\overline{L_n}$, for full credit.

2.3 Q3

Let $\mathcal{A} = (Q, Q_0, \Sigma, \delta, \mathcal{F})$ be our NFA. Let $\mathcal{B} = (2^Q, q_0', \Sigma, \delta', \mathcal{F}')$ be the DFA constructed from our NFA according to the usual powerset construction, assuming that $\mathcal{A}$ is angelic. Note that $\mathcal{F}' = \{S \in 2^Q : S \cap \mathcal{F} \neq \emptyset\}$.
However, if $\mathcal{A}$ were to be now interpreted with a devilish condition, we show that it would still accept a regular language by claiming that the DFA $\tilde{\mathcal{B}}$ is equivalent to $\mathcal{A}_{\text{devilish}}$, where $\tilde{\mathcal{B}} := (2^Q, q_0', \Sigma, \delta', \tilde{\mathcal{F}}')$ where we now define $\tilde{\mathcal{F}}' := \{S \in 2^Q : S \subseteq \mathcal{F}, S \neq \emptyset\}$.
The reason this construction works can be attributed to the meaning of the "subset of states" in the powerset construction: Since NFAs are non-determinstic, when NFAs read a word, they can land up in multiple different states, and we collect those states to form one state of our DFA. Thus, when a set of states contains a final state, that represents that *some* run of the word through the NFA lands up in a final state, and the angelic ondition then ensures it's acceptance.
However, since the devilish acceptance condition requires that *every* run of our NFA ends up in a final state, we correspondingly set our final state condition in the constructed DFA to reflect that.
Also note that the question *does not* claim that $\mathcal{L}(\mathcal{A}_{\text{angelic}}) = \mathcal{L}(\mathcal{A}_{\text{devilish}})$. Indeed, all the question asks us to show is that $\mathcal{L}(\mathcal{A}_{\text{devilish}})$ is regular. In fact it's easy to see that in general, $\mathcal{L}(\mathcal{A}_{\text{angelic}}) \neq \mathcal{L}(\mathcal{A}_{\text{devilish}})$.

2.4 Q4

Let $\mathcal{A} = (Q, q_0, \Sigma, \delta, \mathcal{F})$ be a DFA recognizing L.
Note that a proper prefix of some word w can belong to L if and only if while reading w, we pass through some final state of $\mathcal{A}$ (and don't stop there).
We shall prove L' is regular by constructing a DFA recognizing (exactly) L'. Indeed, consider the DFA $\mathcal{B} := (Q \cup \{d\}, q_0, \Sigma, \delta', Q)$, with the transition function being given by

$$\forall \sigma \in \Sigma, \forall n \in Q \setminus \mathcal{F},\ \delta'(n, \sigma) = \delta(n, \sigma)$$
$$\forall \sigma \in \Sigma, \forall f \in \mathcal{F},\ \delta'(f, \sigma) = \delta'(d, \sigma) = d$$

We finish the proof that L' is regular by noting that $\mathcal{L}(\mathcal{B}) = L'$.
Note:- In this question, we've assumed that ε is a proper prefix of every non-empty word.

2.5 Q5

Assume for the sake of contradiction that there exists $n \in \mathbb{N}$ such that L_n is the language of a DFA $\mathcal{A}$ with $< 2^{n-1}$ states. Let K_n be the set of all strings with length $\mathbf{n-1}$. For any $w \in \Sigma^*$, denote s_w to be the state $\mathcal{A}$ reaches on reading w.
Since $|K_n| = 2^{n-1}$, and since $\mathcal{A}$ has $< 2^{n-1}$ states, by the pigeonhole principle there exist two distinct words $u, v \in K_n$ such that $s_u = s_v$. Since $u \neq v$, there

exists $\ell \in [n-1]$ such that $u_\ell \neq v_\ell$. WLOG let $u_\ell = 0, v_\ell = 1$. Now, since $s_u = s_v$, we also have $s_{u0^\ell} = s_{v0^\ell}$. Now, since the n^{th} last bit of $v0^\ell$ is the same as $v_\ell = 1$, we get that $v0^\ell \in L_n$, and consequently $s_{v0^\ell} = s_{u0^\ell}$ is a final state of $\mathcal{A}$, implying that $u0^\ell \in L_n$. However, the n^{th} last bit of $u0^\ell$ is equal to $u_\ell = 0$, which contradicts the definition of L_n.