CS228 Tutorial Solutions

Arpon Basu

March 22, 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 $\Sigma 0\Sigma^* 0\Sigma + \Sigma 0\Sigma + 0^2$, 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.

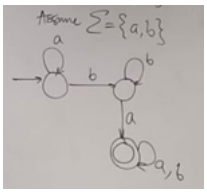

Figure 1: 2(2)(c)

(3)

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

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

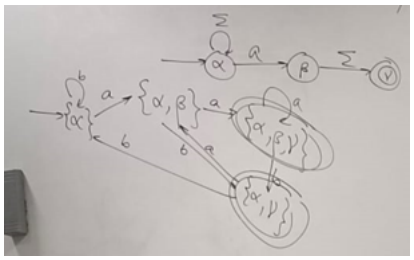

Figure 2: 2(3)(c)

(c)

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

(4)

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

(b)

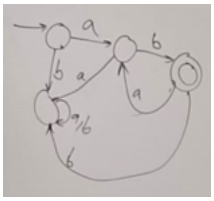

Figure 3: 2(4)(c)

(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-deterministic, 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 condition then ensures its 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. Its 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 $\mathcal{B} := (Q \cup \{d\}, q_0, \Sigma, \delta', \mathcal{F})$, 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.

3 Tutorial 3

3.1 Q1

1. The formulation of a counting language is

$$\forall n_0 \exists n \geq n_0 \exists u, v, w \in \Sigma^*. uv^n w \in L \iff \neg(uv^{n+1}w \in L)$$

2. $(aa)^+$ is a counting language: Indeed, let n_0 be any natural number. Then for $u = \varepsilon, v = a, w = \varepsilon$, $uv^{n_0}w$ and $uv^{n_0+1}w$ can't both belong to $(aa)^+$.

3. We shall prove that $(ab)^+$ is non-counting by showing that for all $u, v, w \in \Sigma^*$, we have for $L := \{uv^n w : n \geq 2\}$, $L \cap (ab)^+ = L$ or $\emptyset$. Then the definition of non-counting would be satisfied by the witness $n_0 = 2$.
   If $v = \varepsilon$, the statement is trivially true, so assume $v \neq \varepsilon$. If $|v| = 1$, $L \cap (ab)^+ = \emptyset$, since $uv^n w$, for every $n \geq 2$, contains two consecutive alphabets which are the same. Similarly, even for any v such that $|v| \geq 2$, if v contains two consecutive alphabets then $L \cap (ab)^+ = \emptyset$. Similarly, for v with $|v| \geq 2$, $L \cap (ab)^+ = \emptyset$ if the start and end letter of v are the same. Consequently, we assume that v is of the form $(ab)^t$ or $(ba)^t$ for some $t \geq 1$.
   If $v = (ab)^t$, then for any $n \in \mathbb{N}$ if we have $uv^n w \in (ab)^+$, we must also have $u, w \in (ab)^*$. But then L is composed of strings of the form $(ab)^*(ab)^{tn}(ab)^*$, which is a subset of $(ab)^+$, thus showing that $L \cap (ab)^+ = L$, as desired.
   If $v = (ba)^t$, then for any $n \in \mathbb{N}$ if we have $uv^n w \in (ab)^+$, we must also have $u \in (ab)^* a, w \in b(ab)^*$. But then L is composed of strings of the form $(ab)^* a (ba)^{tn} b (ab)^*$, which is a subset of $(ab)^+$, thus showing that $L \cap (ab)^+ = L$, as desired.

3.2 Q2

(a) Let S be the second order variable denoting the set of vertices reachable from s, and let S' be the second order variable denoting the set of vertices **not** reachable from s. Then we define

$$\text{Def}(S) := S(s) \wedge \forall x.(S(x) \wedge E(x, y) \implies S(y))$$

$$\text{Def}(S') := \forall x.(S'(x) \implies (\forall y.(S(y) \implies \neg E(y, x))))$$

$$\text{Partition}(S, S') := \forall x.(S(x) \iff \neg S'(x))$$

Our final sentence is

$$\exists S, S'.(\text{Def}(S) \wedge \text{Def}(S') \wedge \text{Partition}(S, S') \wedge S(t))$$

(b) We define a few formulae first:

$$\text{NonEmpty}(X) := \exists x.X(x)$$

$$\text{UpperBound}(X, y) := \forall z.(X(z) \implies z \leq y)$$

$$\text{Bounded}(X) := \exists y.\text{UpperBound}(X, y)$$

$$\text{LeastUpperBound}(X, y) := \text{UpperBound}(X, y) \wedge \forall y'((\forall z.(X(z) \implies z \leq y')) \implies y \leq y')$$

Then our desired MSO sentence is

$$\forall X.(\text{NonEmpty}(X) \wedge \text{Bounded}(X) \implies \exists y.\text{LeastUpperBound}(X, y))$$

3.3 Q3

MSO and MSO_0 are equi-expressive.
Consider a MSO_0 formula. Wherever the formula contains the following MSO_0 atomic formulae, replace them with the following MSO formulae [2]. In this conversion, ensure that the variables introduced (free or quantified) are not present anywhere else in the formula, quantified or otherwise.

1. $\text{Sing}(X)$: $\exists x.(X(x) \wedge \forall y.(X(y) \implies (x = y)))$.
2. $X \subseteq Y$: $\forall x.(X(x) \implies Y(x))$
3. $X < Y$: First replace this by $\text{Sing}(X) \wedge \text{Sing}(Y) \wedge \forall x, y.(X(x) \wedge Y(y) \implies x < y)$, and then replace the Sing predicates according to the first rule.
4. $S(X, Y)$: First replace this by $\text{Sing}(X) \wedge \text{Sing}(Y) \wedge \forall x, y.(X(x) \wedge Y(y) \implies S(x, y))$, and then replace the Sing predicates according to the first rule.
5. $Q_a(X)$: $\forall x.(X(x) \implies Q_a(x))$

If we are given a formula in MSO, we can convert it to MSO_0 as follows:

1. $x = y$: $\text{Sing}(X) \wedge \text{Sing}(Y) \wedge Y \subseteq X$
2. $x < y$: $\text{Sing}(X) \wedge \text{Sing}(Y) \wedge X < Y$
3. $S(x, y)$: $\text{Sing}(X) \wedge \text{Sing}(Y) \wedge S(X, Y)$
4. $X(x)$: $\text{Sing}(Y) \wedge Y \subseteq X$. Here Y plays the role of x in the original formula.
5. $Q_a(x)$: $\text{Sing}(X) \wedge Q_a(X)$

[2] these MSO_0 atomic formulae are replaced sequentially, in any arbitrary order

3.4 Q4

$$\exists X.\forall Y.(\text{Sing}(X) \wedge \text{Sing}(Y) \wedge (X < Y \implies Q_a(Y)))$$

3.5 Q5

Note that $L(N) = b(a^+b^3)^*$. We shall use two SO variables X_0, X_1 such that $X_0 \xrightarrow{b} X_1, X_1 \xrightarrow{a^+b^2} X_0$. Then, our sentence is

$$\exists X_0, X_1 \cdot$$

$$\forall x.(X_0(x) \iff \neg X_1(x)) \bigwedge$$

$$X_0(\text{first}) \wedge X_1(\text{last}) \wedge Q_b(\text{last}) \bigwedge$$

$$\forall x, y.S(x, y) \wedge X_0(x) \wedge Q_b(x) \implies X_1(y) \bigwedge$$

$$\forall y.X_1(y) \implies \exists x.S(x, y) \wedge X_0(x) \wedge Q_b(x) \bigwedge$$

$$\forall x, y.$$

$$x < y \wedge X_1(x) \wedge \forall z.(x < z < y \implies (\neg X_0(z) \wedge \neg X_1(z))) \bigwedge$$

$$\exists y_1, y_2.S(y_2, y_1) \wedge S(y_1, y) \wedge Q_b(y_2) \wedge Q_b(y_1) \wedge \forall z.(x < z < y_2 \implies Q_a(z)) \wedge ((\exists z.x < z < y_2) \implies X_0(y))$$

A very troll-like answer to this question would be to note that $L(N)$ is FO-definable: Consequently, write the FO formula to describe $L(N)$ (which we already did in a previous tutorial), and introduce two dummy SO variables to sit and do nothing, thus constructing a MSO formula with two SO variables describing $L(N)$.