CS 228 (M) - Logic in CS
Tutorial VII - Solutions

Ashwin Abraham

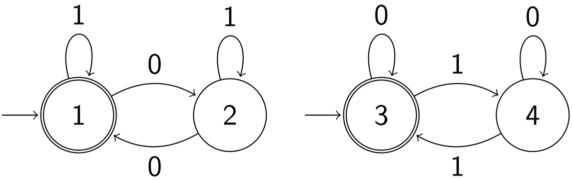
IIT Bombay

10th October, 2023

Table of Contents

Question 1

1. For a word x, $\#_0(x)\#_1(x)$ is even if and only if $\#_0(x)$ is even **or** $\#_1(x)$ is even. We know that if L_A is the language accepted by an NFA A and L_B is the language accepted by an NFA B, the language $L_A \vee L_B$ is the language accepted by the automaton constructed by putting A and B side by side. Therefore, our required automaton would be:



The component on the left accepts words x with even $\#_0(x)$ and the one the right accepts words x with even $\#_1(x)$, and the overall automaton accepts words x with even $\#_0(x)\#_1(x)$.

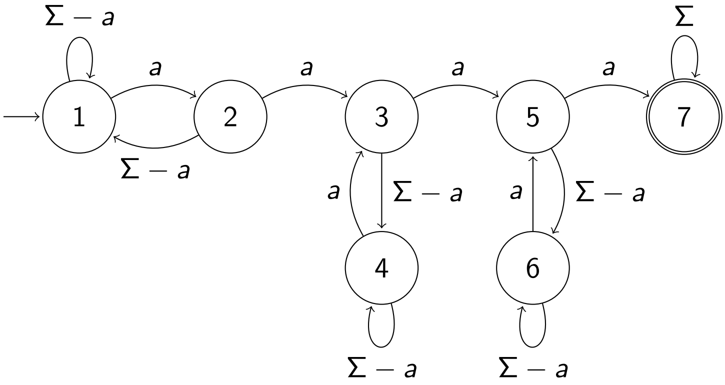
Question 1

2. This language is FO definable, and an FO formula defining this language would be:

$$\exists x_1 \exists x_2 \exists x_3 [(x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_3 \neq x_1) \wedge Q_a(x_1) \wedge \\ \exists y_1 (Q_a(y_1) \wedge S(x_1, y_1)) \wedge Q_a(x_2) \wedge \exists y_2 (Q_a(y_2) \wedge S(x_2, y_2)) \wedge \\ Q_a(x_3) \wedge \exists y_3 (Q_a(y_3) \wedge S(x_3, y_3))]$$

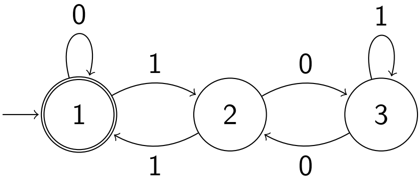
Question 1

A DFA with 7 states accepting this language is:



Question 1

3. A DFA accepting this language is:



Domain Transformations

Please go through this document on domain transformations.

Question 2

Solved in the second page of the document.

Question 3

1. Solved in the second page of the document.
2. We will show that if L_1 and L_2 are regular, then $\texttt{shuffle}(L_1, L_2)$ is regular, from which the desired statement can immediately be proven. Say $L_1 = L(Q_1, \Sigma, \delta_1, q_1, F_1)$ and $L_2 = L(Q_2, \Sigma, \delta_2, q_2, F_2)$. Consider the NFA $A = (Q_1 \times Q_2, \Sigma, \Delta, \{(q_1, q_2)\}, F_1 \times F_2)$ where $\Delta((p, q), a) = \{(\delta_1(p, a), q), (p, \delta_2(q, a))\}$. We assert that $\texttt{shuffle}(L_1, L_2) = L(A)$. To prove this firstly note that $\hat{\Delta}(\{(q_1, q_2)\}, w) = \{(\hat{\delta}_1(q_1, u), \hat{\delta}_2(q_2, v)) : w \in \texttt{shuffle}(u, v)\}$. This can be proven by induction on $|w|$.
Now, $\hat{\Delta}(\{(q_1, q_2)\}, w) \cap (F_1 \times F_2) \neq \emptyset$ iff there exist u and v such that $w \in \texttt{shuffle}(u, v)$, $\hat{\delta}_1(q_1, u) \in F_1$ and $\hat{\delta}_2(q_2, v) \in F_2$, ie $u \in L_1$ and $v \in L_2$ and $w \in \texttt{shuffle}(L_1, L_2)$, ie $w \in \texttt{shuffle}(L_1, L_2)$.
Therefore, our assertion that $\texttt{shuffle}(L_1, L_2) = L(A)$ is proven, which implies that $\texttt{shuffle}(L_1, L_2)$ is regular. $\square$

Question 4

We have $h : \Sigma^* \to \Gamma^*$ such that $\forall x, y \in \Sigma^*\ h(xy) = h(x)h(y)$[1]. Putting $x = y = \varepsilon$, we get $h(\varepsilon) = h(\varepsilon)^2$, ie $h(\varepsilon) = \varepsilon$. It can easily be shown that $h(x_1 x_2 \dots x_n) = h(x_1)h(x_2)\dots h(x_n)$, ie if $x = a_1 a_2 \dots a_n$ for $a_1, a_2, \dots a_n \in \Sigma$, then $h(x) = h(a_1)h(a_2)\dots h(a_n)$. This means that h is wholly determined by its values on Σ.

1. Say $L = L(Q, \Sigma, \delta, q_0, F')$. Consider the ε-NFA $(Q', \Gamma, \Delta, \{(q_0, \varepsilon)\}, F)$ where $Q' = \{(q, w) : q \in Q, w \in \mathtt{pref}(h(a)), a \in \Sigma\}$, $\Delta = \{((q, w), \gamma, (q, w\gamma)) : q \in Q, \gamma \in \Gamma, w\gamma \in \mathtt{pref}(h(a)), a \in \Sigma\} \cup \{((q, h(a)), \varepsilon, (\delta(q, a), \varepsilon)) : q \in Q, a \in \Sigma\}$ and $F' = F \times \{\varepsilon\}$. Note that Q' is finite since $h(a)$ is finite for each $a \in \Sigma$ and Σ is finite. For this ε-NFA, $\hat{\Delta}(\{(q_0, \varepsilon)\}, w) = \{(\hat{\delta}(q_0, x), z) : z \in \mathtt{pref}(h(a)), h(x)z = w\}$. This can be proven by induction on $|w|$. Now, $\hat{\Delta}(\{(q_0, \varepsilon)\}, w) \cap (F \times \{\varepsilon\}) \neq \emptyset$ iff there exists $x \in \Sigma^*$ such that $h(x) = w$, ie $w \in h(L)$. Therefore, the language of this ε-NFA is $h(L)$, which means $h(L)$ is regular. □

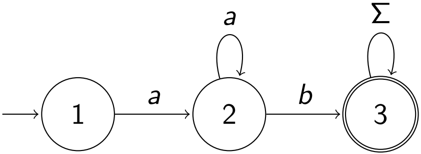
[1] Such a function is called a homomorphism from Σ^* to Γ^*

For $L \subseteq \Gamma^*$, we can also define $h^{-1}(L) = \{x \in \Sigma^* : h(x) \in L\}$. If L is regular, ie it is accepted by the DFA $(Q, \Gamma, \delta, q_0, F)$, then we can show that $h^{-1}(L)$ is also regular. Consider the DFA $A = (Q, \Sigma, \delta', q_0, F)$ where $\delta'(q, a) = \hat{\delta}(q, h(a))$. It can be proved by induction that $\hat{\delta}'(q_0, w) = \hat{\delta}(q_0, h(w))$, ie $w \in L(A)$ iff $h(w) \in L$, which means $L(A) = h^{-1}(L)$, which shows that $h^{-1}(L)$ is regular. $\square$

2. Consider the FO definable language $L = (ab)^*$ over $\Sigma = \{a, b\}$ and the homomorphism $h : \Sigma^* \to \Sigma^*$ such that $h(a) = h(b) = a$. $h(L) = (aa)^*$ which is not FO definable. Therefore, FO definability is not preserved under homomorphism. $\square$

Question 5

- $\varphi = \exists y\,[\neg \mathit{first}(y) \wedge Q_b(y) \wedge \forall x(x < y \implies Q_a(x))]$. Since the outermost quantifier here is existential, φ is clearly not valid, since $\varepsilon \nvDash \varphi$. It is however, satisfiable since $ab \nvDash \varphi$. In fact, $L(\varphi) = a^+ b\Sigma^*$. An NFA accepting this language is:

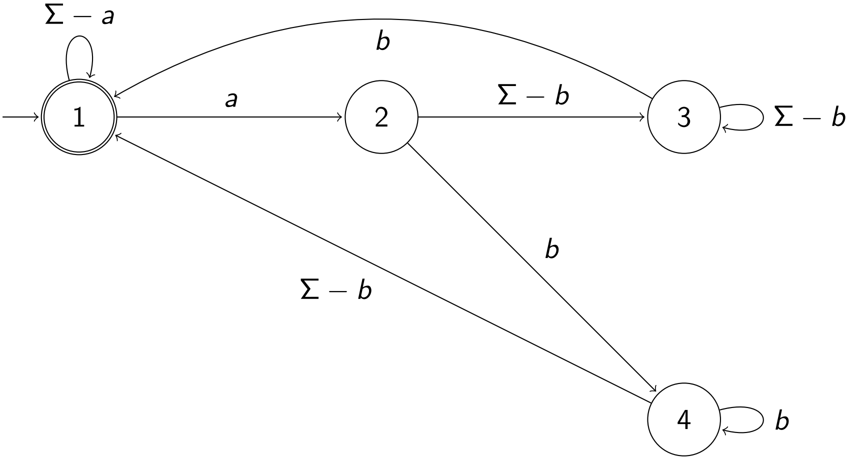


-

$$\varphi = \forall x[Q_a(x) \implies \exists y(x < y \wedge Q_b(y))$$
$$\wedge \exists z_1, z_2[x < z_1 \wedge z_1 < z_2 \wedge \neg(Q_b(z_1) \wedge Q_b(z_2))]])]$$

Firstly, since the outermost quantifier is here is universal, φ is clearly satisfiable as $\varepsilon \vDash \varphi$. It is not valid, as $a \nvDash \varphi$.

A DFA accepting $L(\varphi)$ is:



We have $L(\varphi) = ((\Sigma - a) + a[b^{+}(\Sigma - b) + (\Sigma - b)^{+}b])^{*}$.

Question 6

1. We will construct the set of nodes reachable from s, and assert that t belongs in this set. Let

$$\varphi(X, s) = [X(s) \wedge \forall u, v(X(u) \wedge E(u, v) \implies X(v))]$$

$\varphi(X, s)$ asserts that X contains all the nodes reachable from s - it may contain other nodes also (to see this, note that $\varphi(V, s)$ is always true). We want X to be the set containing precisely the nodes reachable from s. Note that this is the "smallest" set satisfying φ, ie it is a subset of every set satisfying φ. The required formula therefore becomes:

$$\exists X\,[\varphi(X, s) \wedge \forall Y(\varphi(Y, s) \implies \forall u\,[X(u) \implies Y(u)]) \wedge X(t)]$$

2. Let $\varphi(X, t)$ denote that t is an upper bound of X, ie

$$\varphi(X, t) = \forall x\,[X(x) \implies x \leq t]$$

the required sentence is then:

$$\forall X\,[\exists x\; X(x) \implies \exists t(\varphi(X, t) \wedge \forall r\,[\varphi(X, r) \implies t \leq r])]$$

Question 7

We will show that $\mathtt{MSO}$ and $\mathtt{MSO}_0$ are equally expressive. Firstly, note that every production rule of $\mathtt{MSO}_0$ is also a production rule of $\mathtt{MSO}$. The atomic formulae of $\mathtt{MSO}_0$ can also be expressed in $\mathtt{MSO}$ as follows:

- $Sing(X) \equiv \exists x\,[X(x) \wedge \forall y(X(y) \implies y = x)]$
- $X \subseteq Y \equiv \forall x\,[X(x) \implies Y(x)]$
- $X < Y \equiv Sing(X) \wedge Sing(Y) \wedge \exists x, y\,[x < y \wedge X(x) \wedge X(y)]$ where $Sing(X)$ is as defined previously
- $S(X, Y) \equiv Sing(X) \wedge Sing(Y) \wedge \exists x, y\,[S(x, y) \wedge X(x) \wedge Y(y)]$
- $Q_a(X) \equiv \forall x\,[X(x) \implies Q_a(x)]$

Therefore, $\mathtt{MSO}$ is at least as expressive as $\mathtt{MSO}_0$.

Question 7

To show that $\texttt{MSO}_0$ and $\texttt{MSO}$ are equally expressive, we need to show that every $\texttt{MSO}$ formula has an equivalent $\texttt{MSO}_0$ formula. However, $\texttt{MSO}_0$ has no first order variables, and so we have to map the first order variables in the $\texttt{MSO}$ formula to appropriate second order $\texttt{MSO}_0$ variables. We will map the first order variables to second order variables corresponding to singleton sets. The mappings of the atomic MSO_0 formulae are (here X and Y are the second order variables corresponding to the first order variables x and y):

- $x = y \equiv Sing(X) \wedge Sing(Y) \wedge X \subseteq Y \wedge Y \subseteq X$
- $x < y \equiv X < Y$
- $S(x, y) \equiv S(X, Y)$
- $H(x) \equiv Sing(X) \wedge X \subseteq H$
- $Q_a(x) \equiv Sing(X) \wedge Q_a(X)$

All production rules of $\texttt{MSO}$ are also production rules of $\texttt{MSO}_0$, except $\forall x \varphi(x)$, which becomes $\forall X\, [Sing(X) \implies \varphi'(X)]$ where $\varphi'(X)$ is the formula corresponding to $\varphi(x)$. Therefore, every $\texttt{MSO}$ formula has an equivalent $\texttt{MSO}_0$ formula, which implies that they are equally expressive.

Question 8

We can write $\exists x \forall y\, [x < y \implies Q_a(y)]$ as $\neg\forall x \neg\forall y\, [x < y \implies Q_a(y)]$. Using the equivalences discussed in the previous problem, the $\mathtt{MSO}_0$ equivalent of this would be:

$$\neg\forall X\, [Sing(X) \implies \neg\forall Y(Sing(Y) \implies [X < Y \implies Sing(Y) \wedge Q_a(Y)])]$$

This can be simplified to:

$$\neg\forall X\, [Sing(X) \implies \neg\forall Y(X < Y \implies Q_a(Y))]$$

Question 9

The principle we use is similar to that of Question 6a - if $x < y$, then we can reach y from x by moving to the successor of the element you are at a finite number of times. Let

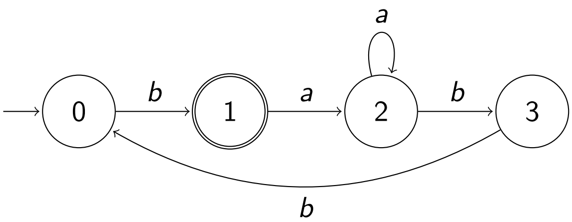
$$\varphi(X, x) = X(x) \wedge \forall u, v\,[X(u) \wedge S(u, v) \implies X(v)]$$

$\varphi(X, x)$ asserts that X contains all numbers greater than or equal to x. Our required formula will therefore be:

$$\exists X\,[\varphi(X, x) \wedge \forall Y(\varphi(Y, x) \implies \forall u\,[X(u) \implies Y(u)]) \wedge X(y)] \wedge \neg(x = y)$$

Question 10

N can be written as:



We have $L(N) = b(a^+b^3)^*$. This language is actually FO definable, as can be seen in Tutorial 5, Question 3 and so we can write it as an MSO formula as well by adding two valid clauses of the form $\exists X\,[\forall y\ \neg X(y)]$ to trivially add two MSO variables.

Question 11

This question was present in this tutorial sheet when this course was run in the Spring Semester. Call $L \subseteq \Sigma^+$ non counting iff

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

That is for all $n \geq n_0$, either all $uv^n w$ are in L, or none of them are.
A language is counting iff it is not non counting.

- Formulate the condition for a counting language
- Is $L = (aa)^+$ counting?
- Is $L = (ab)^+$ counting?

Question 11

- We just negate the condition for a language to be non counting the get the condition for a language to be counting, ie:

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

- $L = (aa)^+$ is counting. This is as for all n_0, we can choose $n = n_0 + 1$, $u = w = \varepsilon$ and $v = a$. We will have $uv^n w \in L \oplus uv^{n+1} w \in L$, as $uv^n w = a^{n_0+1}$ and $uv^{n+1} w = a^{n_0+2}$, and out of these two, exactly one will be in L, satisfying the definition of a counting language.
- $L = (ab)^+$ is a non counting language. We choose $n_0 = 2$. For every $n \geq 2$, if $uv^n w \in L$, $v \in (ba)^* + (ab)^*$. To see this, note that v cannot contain two consecutive identical letters, and it must begin and end with different letters (if it is non-empty). This clearly implies that $uv^{n+1} w \in L$ as well.