Formal Languages and Automata Theory

First Long Test

Time: 50 minutes — 12–February–2021 — Maximum marks: 40

---

1. Consider the following language over the alphabet $\{a,b\}$:

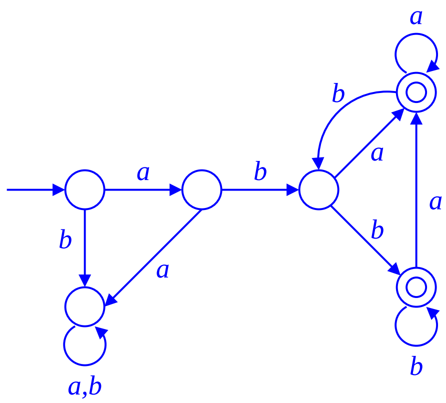
$$L_1 = \left\{ x \in \{a,b\}^* \;\middle|\; x \text{ starts with } ab \text{ but does not end with } ab \right\}.$$

(a) Write a regular expression for L_1. **(4)**

Solution $ab(a+b+(a+b)^*(aa+ba+bb))$.

(b) Design a DFA for L_1. **(6)**

Solution



2. **(a)** Construct a regular expression over the alphabet $\{a,b,c\}$ for **(4)**

$$L_2 = \left\{ x \in \{a,b,c\}^* \;\middle|\; x \text{ has } 4i+1 \; b\text{'s for some integer } i \geqslant 0 \right\}.$$

Solution $((a+c)^*b(a+c)^*b(a+c)^*b(a+c)^*b)^*(a+c)^*b(a+c)^*$.

(b) For a language A over some alphabet Σ, define the language

$$\text{cyclicshift}(A) = \left\{ yx \;\middle|\; xy \in A \text{ for some } x,y \in \Sigma^* \right\}.$$

If A is regular, prove that cyclicshift(A) is regular too. **(6)**

Solution Let $D = (Q,\Sigma,\delta,s,F)$ be a DFA for A. Fix a $q \in Q$, and define the language

$$B_q = \left\{ yx \;\middle|\; \hat{\delta}(s,x) = q \text{ and } \hat{\delta}(q,y) \in F \right\}.$$

We design an ε-NFA $N = (Q_N,\Sigma,\Delta_N,s_N,F_N)$ for B_q as follows. Q_N consists of two copies of Q, that is, we take $Q_N = \{p \mid p \in Q\} \cup \{p' \mid p \in Q\}$. We keep all the transitions of D in both the copies, that is, for every transition $\delta(p,a) = q$ of D, we have $q \in \Delta_N(p,a)$ and $q' \in \Delta_N(p',a)$. For every final state $f \in F$, we add the transition $s' \in \Delta_N(f,\varepsilon)$. Finally, we take $s_N = q$, and $F_N = \{q'\}$.

It therefore follows that each B_q is regular. But then,

$$\text{cyclicshift}(A) = \bigcup_{q \in Q} B_q$$

is the union of finitely many regular languages, and is regular too.

3. One of the following languages over the alphabet $\{0,1\}$ is regular; the other is not. Identify which one is what. Justify. No credit for correct identification only.

(a) $L_{3(a)} = \left\{ \alpha\beta\alpha \mid \alpha \in \{0,1\}^+ \text{ and } \beta \in \{0,1\}^+ \right\}$. **(5)**

Solution *Not regular.* For proving this, suppose that $L_{3(a)}$ is regular. Let $k \in \mathbb{N}$ be a pumping-lemma constant for $L_{3(a)}$. The string $01^k0^21^k$ belongs to $L_{3(a)}$ (here $\alpha = 01^k$ and $\beta = 0$). Using the notations of the pumping lemma, take $x = 0$, $y = 1^k$, and $z = 0^21^k$. The pumping lemma gives a decomposition $y = uvw$ with $|v| > 0$ (so $v = 1^l$ for some $1 \leqslant l \leqslant k$). Pumping out v gives $01^{k-l}0^21^k \in L_{3(a)}$. But this string cannot be written in the form $\alpha\beta\alpha$, and so is not in $L_{3(a)}$, a contradiction.

(b) $L_{3(b)} = \left\{ \alpha\beta\alpha \mid \alpha \in \{0\}^+ \text{ and } \beta \in \{0,1\}^+ \right\}$. **(5)**

Solution *Regular.* Show that $L_{3(b)}$ is the set of all strings over $\{0,1\}$ of length $\geqslant 3$, that start and end with 0, that is, $L_{3(b)} = \mathscr{L}(0(0+1)^+0) = \mathscr{L}(0(0+1)(0+1)^*0)$.

4. Use the Myhill–Nerode theorem (no credit for using any other method) to prove the regularity/non-regularity of the following languages over the alphabet $\{a,b\}$.

(a) $L_{4(a)} = \left\{ x \in \{a,b\}^* \mid \#a(x) - \#b(x) = 2021 \right\}$. **(5)**

Solution *Not regular.* Define the surplus of a's over b's in $w \in \{a,b\}^*$ as $s(w) = \#a(w) - \#b(w)$. We have

$$L_{4(a)} = \left\{ x \in \{a,b\}^* \mid s(x) = 2021 \right\}.$$

Define a relation on $\{a,b\}^*$ as $x \equiv_a y$ if and only if $s(x) = s(y)$. It is easy to see that $\equiv_a$ is an equivalence relation, is a right congruence, and refines $L_{4(a)}$.

In order to show that $\equiv_a$ is the coarsest MN$'$ relation for $L_{4(a)}$, take any MN$'$ relation $\equiv$ for $L_{4(a)}$. Let $x \equiv y$. There exists $z \in \{a,b\}^*$ such that $xz \in L_{4(a)}$. This is because if $s(x) = 2021$, we take $z = \varepsilon$; if $s(x) = 2021 + r$ for some $r > 0$, we take $z = b^r$; and if $s(x) = 2021 - r$ for some $r > 0$, we take $z = a^r$. Since $\equiv$ is a right congruence and refines $L_{4(a)}$, we have $yz \in L_{4(a)}$ for this choice of z. But then, $s(xz) = s(yz) = 2021$, that is, $s(x) + s(z) = s(y) + s(z)$, that is, $s(x) = s(y)$, that is, $x \equiv_a y$.

Finally, note that $\equiv_a$ has infinitely many equivalence classes—one class for each $n \in \mathbb{Z}$.

(b) $L_{4(b)} = \left\{ x \in \{a,b\}^* \mid \#a(x) - \#b(x) \text{ is a multiple of } 2021 \right\}$. **(5)**

Solution *Regular.* Define the surplus function $s : \{a,b\}^* \to \mathbb{Z}$ as in Part (a). We have

$$L_{4(b)} = \left\{ x \in \{a,b\}^* \mid s(x) \equiv 0 \pmod{2021} \right\}.$$

Define a relation on $\{a,b\}^*$ as $x \equiv_b y$ if and only if $s(x) \equiv s(y) \pmod{2021}$. It is easy to see that $\equiv_b$ is an equivalence relation, is a right congruence, and refines $L_{4(b)}$.

In order to show that $\equiv_b$ is the coarsest MN$'$ relation for $L_{4(b)}$, take any MN$'$ relation $\equiv$ for $L_{4(b)}$. Let $x \equiv y$. There exists $z \in \{a,b\}^*$ such that $xz \in L_{4(b)}$. This is because if $s(x) = 2021q + r$ for $q \in \mathbb{Z}$ and $r \in \{0,1,2,\ldots,2020\}$, we can take $z = b^r$. Since $\equiv$ is a right congruence and refines $L_{4(b)}$, we have $yz \in L_{4(b)}$ for this choice of z. But then, $s(xz) \equiv 0 \pmod{2021}$ and $s(yz) \equiv 0 \pmod{2021}$, that is, $s(xz) \equiv s(yz) \pmod{2021}$, that is, $s(x) + s(z) \equiv s(y) + s(z) \pmod{2021}$, that is, $s(x) \equiv s(y) \pmod{2021}$, that is, $x \equiv_b y$.

Finally, note that $\equiv_b$ has only finitely many (2021 to be precise) equivalence classes.