PART II

Finite and Regular Languages
Solutions

Part II
Finite and Regular Languages

5 Finite Languages

9.

$$\mathbf{card}(L_1 \cdot L_2) \leq \mathbf{card}(L_1 \times L_2) = \mathbf{card}(L_1) \cdot \mathbf{card}(L_2).$$

Why is it $\leq$ and not just $=$?

(Solution)

Because there may be strings $w_1, v_1 \in L_1$ and $w_2, v_2 \in L_2$ such that $w_1 w_2 = v_1 v_2$, e.g., $w_1 = v_2 = \varepsilon$ and $w_2 = v_1$.

10.

$$\mathbf{card}(L_1 \cup L_2) \leq \mathbf{card}(L_1) + \mathbf{card}(L_2).$$

Again, why is it $\leq$ and not $=$?

(Solution)

Because any string in the both L_1 and L_2 shows up only once in their union.

11. Why does this proof not work for infinite languages as well?

(Solution)

Because if the language is not finite then the induction is not well founded. If L is infinite then $L \setminus \{w\}$ is infinite as well. Consequently, the induction never reaches the base case.

12. Recall that we require alphabets to be finite. What happens to this definition if Σ is infinite—does the class it defines include infinite languages?

(Solution)

The proof of Lemma 1 does not depend, in any way, on finiteness of Σ. Thus the lemma is valid whether Σ is finite or not; the class of the definition is still all and only the finite languages over Σ. The finiteness

of the languages in the class is a consequence of the finiteness of the construction. A language is in the class iff it can be constructed in finitely many steps and each step can add only finitely many strings to the languages that have already been constructed.

6 Regular Languages and Regular Expressions

13. Is it possible for a given language to be the denotation of more than one regular expression? If so, give a simple example.

    (Solution)

    This follows immediately from the algebraic properties of the set operations, associativity of concatenation, for instance, or associativity and commutativity of union:

$$\begin{aligned} L(a \cdot (a \cdot a)) &= \{a\} \cdot (\{a\} \cdot \{a\}) = \{a\} \cdot \{aa\} = \{aaa\} \\ &= \{aa\} \cdot \{a\} = (\{a\} \cdot \{a\}) \cdot \{a\} = L((a \cdot a) \cdot a). \\ L(a + b) &= \{a\} \cup \{b\} = \{a, b\} = \{b\} \cup \{a\} = L(b + a). \end{aligned}$$

14. Show that the basis expression 'ε' is redundant; every set that can be defined using it can also be defined without it.

    (Solution)

$$L(\emptyset^*) = (L(\emptyset))^* = (\emptyset)^* = \{\varepsilon\}.$$

    Thus every occurence of 'ε' can be replaced with '$(\emptyset^*)$'. Note that this is another example of distinct regular expressions that denote the same set.

15. What is the denotation of '$(b^*((ab^*)^*a + \varepsilon))^*$'?

    (Solution)

$$(\{b\}^* \cdot ((\{a\} \cdot \{b\}^*)^* \cdot \{a\} \cup \{\varepsilon\}))^*.$$

16. Show that $R^* = R + R^*$.

(Solution)

To show that $L(R^*) = L(R + R^*)$ we will first show that $L(R) \subseteq L(R^*)$:

$$L(R) = L(R)^1 \subseteq L(R)^* = L(R^*).$$

Then

$$L(R + R^*) = L(R) \cup L(R^*) = L(R^*).$$

17. Show that $(R + S)^* = (R^*S^*)^*$.

(Solution)

To show that $L((R + S)^*) \subseteq L((R^*S^*)^*)$, suppose $w \in L((R + S)^*)$. Then $w \in (L(R) \cup L(S))^*$ and, in particular, $w \in (L(R) \cup L(S))^k$ for some $k \geq 0$.
Consequently, we can divide w into $w_1 \cdot w_2 \cdot \cdots \cdot w_k$ where each w_i is either in $L(R)$ or in $L(S)$.
If $w_i \in L(R)$ then $w_i \in L(R)^1 L(S)^0 \subseteq L(R^*S^*)$.
Similarly, if $w_i \in L(S)$ then $w_i \in L(R)^0 L(S)^1 \subseteq L(R^*S^*)$.
Evidently, then,

$$w \in L(R^*S^*)^k \subseteq L((R^*S^*)^*).$$

To show that $L((R^*S^*)^*) \subseteq L((R + S)^*)$, suppose that $w_i \in L((R^*S^*)^*)$ and, in particular, that $w_i \in L((R^*S^*)^k)$ for some k. Again we can divide w into $w_1 \cdot \cdots \cdot w_k$; here each of the w_i are in $L(R^*S^*)$.
Then each w_i is in $L(R)^{l_i} L(S)^{m_i}$ for some $l_i, m_i \in \mathbb{N}$ and

$$w_i \in L(R)^{l_i} L(S)^{m_i} \subseteq L(R+S)^{l_i} L(R+S)^{m_i} = L(R+S)^{l_i+m_i} \subseteq L(R+S)^*.$$

It follows that $w \in (L(R + S)^*)^*$, which, by the identity of the second example of Section 7.1, is just $L(R + S)^* = L((R + S)^*)$.

18. Show that $R^* = R^* + \varepsilon$

(Solution)

$$\begin{array}{rcll} R^* & = & R^*R + \varepsilon & \text{(R12)} \\ & = & R^*R + \varepsilon + \varepsilon & \text{(R3)} \\ & = & R^* + \varepsilon & \text{(R12).} \end{array}$$

Alternatively, working as we did in Exercise 16, since $\varepsilon \in L(R^*)$:

$$R^* + \varepsilon = L(R^*) \cup \{\varepsilon\} = L(R^*).$$

19. Write a regular expression for the language over $\{a, b\}$ in which no string contains the sequence 'bab' as a substring. Prove that the regular expression denotes exactly that language.

(Solution)

Let's call this language $L_{\overline{bab}}$. The main insight is that whenever 'a's occur between 'b's they must occur in blocks of at least two. So strings in this language look like: a block of zero or more 'a's, followed by alternations of blocks of any number of 'b's and blocks of at least two 'a's and ending with a block of any number of 'a's. Thus:

$$R = a^*(bb^*aaa^*)^*(\varepsilon + bb^*a^*).$$

(This can be simplified, but it is useful to choose a form that supports an easy proof.)

Claim 1 $L_{\overline{bab}} = L_(R)$.

Proof: $(L(R) \subseteq L_{\overline{bab}})$

If $w \in L(R)$ then $w = uvx$ where $u \in L(a^*)$, $v \in L((bb^*aaa^*)^*)$ and $x \in L(\varepsilon + bb^*a^*)$. Note that no '$b$'s precede any '$a$' in u, that all 'a's in v occur in blocks of at least two, and that no 'b' follows any 'a' in x. Thus no 'bab' occurs anywhere in w.

$(L_{\overline{bab}} \subseteq L(R))$

Suppose w contains no 'bab'. Divide w into substrings between the 'a' and 'b' in every occurrence of 'ab'. Then $w = w_1w_2 \cdots w_k$, where w_1 is a block of zero or more 'a's, w_k is a block of one or more 'b's followed by any number of 'a's, and, for $1 < i < k$, w_i is a block of one or more 'b's followed by some positive number of 'a's. Note that when w_i is concatenated with w_{i+1} the block of 'a's it contains will be both preceded and followed by a 'b'. Since 'bab' does not occur in w, this block must contain at least two 'a's.
It follows, then, that $w_1 \in L(a^*)$, $w_k \in L(\varepsilon + bb^*a^*)$, and, for $1 < i < k$, each $w_i \in L((bb^*aaa^*)^*)$. Hence $w \in L(R)$. ⊣

7 Deterministic Finite-State Automata (DFAs)

20. Sketch a proof that if $w \in L(\mathcal{A})$ according to Definition 29 then $w \in L(\mathcal{A})$ according to Definition 31. (Just give the base case(s), the IH, and an outline of the inductive step.)

    (Solution)

    (I will give the full proof.)
    To show that $\langle q, w\rangle \stackrel{*}{\vdash}_{\mathcal{A}} \langle p, \varepsilon\rangle \Rightarrow \hat{\delta}(q, w) = p$ (induction on the length of the computation):

    (Basis:)

    Suppose $\langle q, w\rangle \stackrel{0}{\vdash} \langle p, \varepsilon\rangle$. Then $w = \varepsilon$, $p = q$ and $\hat{\delta}(q, \varepsilon) = q$.

    (IH:)

    Suppose $\langle q, w\rangle \stackrel{n}{\vdash} \langle p, \varepsilon\rangle \Rightarrow \hat{\delta}(q, w) = p$.

    (Ind:)

    To show that $\langle q, w\rangle \stackrel{n+1}{\vdash} \langle p, \varepsilon\rangle \Rightarrow \hat{\delta}(q, w) = p$:

    Suppose $\langle q, w\rangle \stackrel{n+1}{\vdash} \langle p, \varepsilon\rangle$. Then $\langle p, \varepsilon\rangle$ is the $(n+1)^{\text{st}}$ successor of $\langle q, w\rangle$. Consequently, there must be an n^{th} successor: $w = v\sigma$ and there is some p' such that

    $$\langle q, v\rangle \stackrel{n}{\vdash} \langle p', \varepsilon\rangle \text{ and } \langle p', \sigma\rangle \vdash \langle p, \varepsilon\rangle$$

    and thus, by the definition of $\vdash$, $\delta(p', \sigma) = p$.
    By the IH, $\hat{\delta}(q, v) = p'$ and, consequently, $\hat{\delta}(q, v\sigma) = p$, by the definition of $\hat{\delta}$.

    Using this result

    $$\begin{aligned} w \in L(\mathcal{A}) \text{ (Def. 29)} &\Rightarrow \langle q, w\rangle \stackrel{n}{\vdash} \langle p, \varepsilon\rangle, \text{ for some } p \in F \\ &\Rightarrow \hat{\delta}(q, w) = p \in F \\ &\Rightarrow w \in L(\mathcal{A}) \text{ (Def. 31)} \end{aligned}$$

21. Sketch a proof of the converse: that if $w \in L(\mathcal{A})$ according to Definition 31 then $w \in L(\mathcal{A})$ according to Definition 29.

(Solution)

(I will, again, give the full proof.)
To show that $\hat{\delta}(q,w) = p \Rightarrow \langle q, w\rangle \stackrel{*}{\underset{\mathcal{A}}{\vdash}} \langle p, \varepsilon\rangle$ (induction on the length of the path, i.e., $|w|$):

(Basis:)

Suppose $w = \varepsilon$. Then $\hat{\delta}(q,\varepsilon) = q$ and $\langle q,\varepsilon\rangle \stackrel{0}{\vdash} \langle q,\varepsilon\rangle$.

(IH:)

Suppose $w = v\sigma$ and, for all strings u of length $|v|$, that

$$\hat{\delta}(q,u) = p' \Rightarrow \langle q,u\rangle \stackrel{*}{\underset{\mathcal{A}}{\vdash}} \langle p',\varepsilon\rangle\,.$$

(Ind:)

Suppose $\hat{\delta}(q,v\sigma) = p$. By the definition of $\hat{\delta}$ it must be the case that $\delta(\hat{\delta}(q,v),\sigma) = p$. Let $p' = \hat{\delta}(q,v)$. Then $\delta(p',\sigma) = p$.
By the IH, then, $\langle q,v\rangle \stackrel{|v|}{\vdash} \langle p',\varepsilon\rangle$ and, by the definition of $\vdash$, $\langle p',\sigma\rangle \vdash \langle p,\varepsilon\rangle$. It follows, then, from the definition of $\stackrel{n}{\vdash}$, that $\langle q,v\rangle \stackrel{|v|+1}{\vdash} \langle p,\varepsilon\rangle$.
Using this result:

$$\begin{array}{rcl} w \in L(\mathcal{A}) \text{ (Def. 31)} & \Rightarrow & \hat{\delta}(q,w) = p \in F \\ & \Rightarrow & \langle q,w\rangle \stackrel{n}{\vdash} \langle p,\varepsilon\rangle\,, \text{ for some } p \in F \\ & \Rightarrow & w \in L(\mathcal{A}) \text{ (Def. 29)} \end{array}$$

22. Prove for all DFAs $\mathcal{A}$, that $\varepsilon \in L(\mathcal{A}) \Leftrightarrow q_0 \in F$. (Do not forget that you must prove both directions of the '$\Leftrightarrow$'.)

(Solution, using computations)

$$\varepsilon \in L(\mathcal{A}) \Leftrightarrow \langle q_0,\varepsilon\rangle \stackrel{*}{\vdash} \langle q,\varepsilon\rangle \text{ for some } q \in F.$$

But $\langle q_0,\varepsilon\rangle \stackrel{*}{\vdash} \langle q,\varepsilon\rangle \Leftrightarrow q = q_0$. Hence, $\varepsilon \in L(\mathcal{A}) \Leftrightarrow q_0 \in F$.

(Solution, using paths)
Since $\hat{\delta}(q_0,\varepsilon) = \varepsilon$,

$$\varepsilon \in L(\mathcal{A}) \Leftrightarrow \hat{\delta}(q_0,\varepsilon) \in F \Leftrightarrow q_0 \in F.$$

23. Our interest, in defining DFAs, is in defining $L(\mathcal{A})$. But $L(\mathcal{A})$ is defined in terms of $\hat{\delta}$ rather than δ. Why, then, don't we define DFAs in terms of $\hat{\delta}$ instead of δ?

    (Solution)

    The edge function δ being of type $Q \times \Sigma \to Q$, is finite. The path function $\hat{\delta}$, on the other hand, is of type $Q \times \Sigma^* \to Q$, which, assuming $\Sigma \neq \emptyset$ has infinite domain. If we were to include $\hat{\delta}$ in the definition of the automaton it would be an infinite object and we may as well just list the language itself. In a very strong sense, automata are finite means of defining infinite languages.

24. Suppose $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ is a DFA and that $\hat{\delta}$ is defined accordingly. Prove that, for any strings x and y in Σ^*,

$$\hat{\delta}(q, xy) = \hat{\delta}(\hat{\delta}(q, x), y).$$

    [Hint: use induction on $|y|$.]

    (Solution)

    (Basis:)

    Suppose $|y| = 0$.

$$\begin{array}{rcll} \hat{\delta}(q, xy) & = & \hat{\delta}(q, x) & \text{since } xy = x \\ & = & \hat{\delta}(\hat{\delta}(q, x), \varepsilon) & \text{def. of } \hat{\delta} \\ & = & \hat{\delta}(\hat{\delta}(q, x), y) & \text{since } y = \varepsilon. \end{array}$$

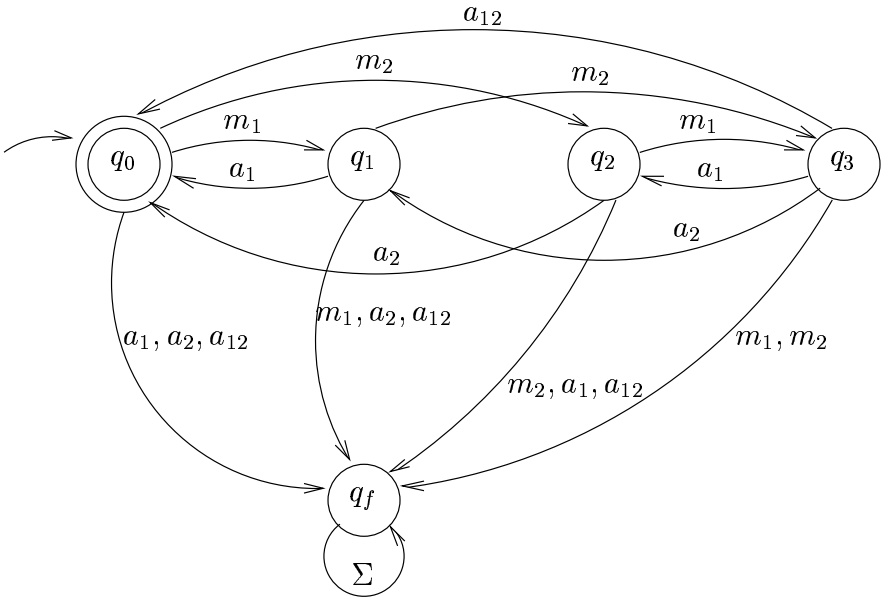
    (Induction:)

    Assume $|y| = n$, $y = y'\sigma$, $\sigma \in \Sigma$ and that the result holds for all $xy' \in \Sigma^*, |y'| < |y|$.

$$\begin{array}{rcll} \hat{\delta}(q, xy) & = & \hat{\delta}(q, xy'\sigma) & \\ & = & \delta(\hat{\delta}(q, xy'), \sigma) & \text{def. of } \hat{\delta} \\ & = & \delta(\hat{\delta}(\hat{\delta}(q, x), y'), \sigma) & \text{IH} \\ & = & \hat{\delta}(\hat{\delta}(q, x), y'\sigma) & \text{def. of } \hat{\delta} \\ & = & \hat{\delta}(\hat{\delta}(q, x), y). & \end{array}$$

25. Show that this is a regular set by providing an automaton and showing that it accepts all and only the strings in L_3.

(Solution)



Invariants:

Q_0 : $\hat{\delta}(q_0, w) = q_0 \Leftrightarrow$ All messages have been ack'd properly
Q_1 : $\hat{\delta}(q_0, w) = q_1 \Leftrightarrow$ Only m_1 is outstanding
Q_2 : $\hat{\delta}(q_0, w) = q_2 \Leftrightarrow$ Only m_2 is outstanding
Q_3 : $\hat{\delta}(q_0, w) = q_3 \Leftrightarrow$ Both m_1 and m_2 are outstanding
Q_f : $\hat{\delta}(q_0, w) = q_f \Leftrightarrow$ Some m_i without a_i or v.v.

To show that these invariants hold for all $w \in \Sigma^*$ (by induction on $|w|$):

(Basis:)

Suppose $w = \varepsilon$, then:

Q_0 : $\hat{\delta}(q_0, \varepsilon) = q_0$ and All messages have been ack'd properly (vacuously).
Q_1 : $\hat{\delta}(q_0, \varepsilon) \neq q_1$ and No m_1 is outstanding.
Q_2 : $\hat{\delta}(q_0, \varepsilon) \neq q_2$ and No m_2 is outstanding.
Q_3 : $\hat{\delta}(q_0, \varepsilon) \neq q_3$ and Neither m_1 or m_2 are outstanding.
Q_f : $\hat{\delta}(q_0, \varepsilon) \neq q_f$ and No m_i without a_i or v.v.

(Induction:)

Suppose $w = w'a,\ a \in \Sigma$ and invariants hold for all w' such that $|w'| < |w|$.

Q_0 : $\hat{\delta}(q_0, w'a) = q_0$ $\Rightarrow$ $\hat{\delta}(q_0, w') = q_1$ and $a = a_1$ or
$\hat{\delta}(q_0, w') = q_2$ and $a = a_2$ or
$\hat{\delta}(q_0, w') = q_3$ and $a = a_{12}$
$\Rightarrow$ Only m_1 was outstanding and a acks it, or
only m_2 was outstanding and a acks it, or
both m_1 and m_2 were outstanding and a acks both.
$\Rightarrow$ All messages have been ack'd properly

Q_1 : $\hat{\delta}(q_0, w'a) = q_1$ $\Rightarrow$ $\hat{\delta}(q_0, w') = q_0$ and $a = m_1$ or
$\hat{\delta}(q_0, w') = q_3$ and $a = a_2$
$\Rightarrow$ All messages were ack'd and a is m_1, or
both m_1 and m_2 were outstanding and a acks m_2.
$\Rightarrow$ Only m_1 is outstanding

Q_2 : $\hat{\delta}(q_0, w'a) = q_2$ $\Rightarrow$ $\hat{\delta}(q_0, w') = q_0$ and $a = m_2$ or
$\hat{\delta}(q_0, w') = q_3$ and $a = a_1$
$\Rightarrow$ All messages were ack'd and a is m_2, or
both m_1 and m_2 were outstanding and a acks m_1.
$\Rightarrow$ Only m_2 is outstanding

Q_3 : $\hat{\delta}(q_0, w'a) = q_3$ $\Rightarrow$ $\hat{\delta}(q_0, w') = q_1$ and $a = m_2$ or
$\hat{\delta}(q_0, w') = q_2$ and $a = m_1$
$\Rightarrow$ Only m_1 was outstanding and a is m_2, or
only m_2 was outstanding and a is m_1.
$\Rightarrow$ Both m_1 and m_2 are outstanding

Q_f : $\hat{\delta}(q_0, w'a) = q_f$ $\Rightarrow$ $\hat{\delta}(q_0, w') = q_0$ and $a \in a_1, a_2, a_{12}$ or
$\hat{\delta}(q_0, w') = q_1$ and $a =\in m_1, a_2, a_{12}$ or
$\hat{\delta}(q_0, w') = q_2$ and $a =\in m_2, a_1, a_{12}$ or
$\hat{\delta}(q_0, w') = q_3$ and $a =\in m_1, m_2$ or
$\hat{\delta}(q_0, w') = q_f$
$\Rightarrow$ All messages ack'd and a is ack, or
only m_1 was outstanding and a is m_1 or acks m_2, or
only m_2 was outstanding and a is m_2 or acks m_1, or
both m_1 and m_2 were outstanding and a is m_1 or m_2, or
Some m_i without a_i or v.v. in w'
$\Rightarrow$ Some m_i without a_i or v.v. in w

To show that $L(M) = L_3$:

$w \in L(M) \Rightarrow \hat{\delta}(q_0, w) = q_0 \Rightarrow$ All messages have been ack'd properly $\Rightarrow w \in L_3$.

$w \notin L(M) \Rightarrow \hat{\delta}(q_0, w) \neq q_0 \Rightarrow$ Some message outstanding or out of order $\Rightarrow w \notin L_3$.