Solutions to the exercises (FoC 2010)

1. Which of these sentences are propositions? What are the truth values of those that are propositions?

   (a) Liverpool is the capital of the UK.

   (b) $x + 2 = 11$.

   (c) $2 + 3 = 5$.

   (d) Answer this question.

   Solution. (a) and (c) are propositions.

2. What is the negation of each of these propositions?

   (a) Today is Thursday.

   (b) There is no pollution in London.

   (c) $2 + 3 = 5$.

   (d) The summer in London is hot and sunny.

   Solution. (a) Today is not Thursday. (b) There is pollution in London. (c) $2 + 3 \neq 5$. (d) The summer in London is not hot and sunny.

3. Construct a truth table for each of the following Boolean formulas.

   (a) $A \wedge \neg A$.

   (b) $A \vee \neg A$.

   (c) $(A \vee \neg B) \rightarrow B$.

   (d) $(A \rightarrow B) \leftrightarrow (\neg B \rightarrow \neg A)$.

   (e) $(A \rightarrow B) \rightarrow (B \rightarrow A)$.

   (f) $(A \oplus B) \rightarrow (A \oplus \neg B)$.

   Solution. The truth-tables:

A	B	(a)	(b)	(c)	(d)	(e)	(f)
0	0	0	1	0	1	1	1
0	1	0	1	1	1	0	0
1	0	0	1	0	1	1	0
1	1	0	1	1	1	1	1

4. Which of the following pairs of formulas are logically equivalent?

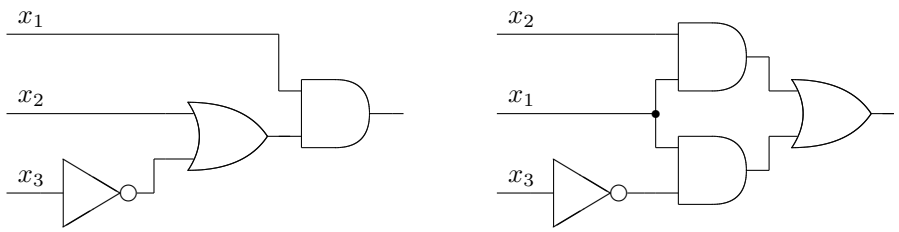
   (a) $(A \rightarrow B) \wedge (A \rightarrow C)$ and $A \rightarrow (B \wedge C)$.

   (b) $(A \rightarrow B) \rightarrow C$ and $A \rightarrow (B \rightarrow C)$.

   (c) $(A \wedge B) \rightarrow C$ and $(A \rightarrow C) \wedge (B \rightarrow C)$.

   Solution. The formulas in (a) are equivalent. The first formula in (b) is false when $A = 0$, $B = 0$, $C = 0$, but the second formula is true. The first formula in (c) is true when $A = 1$, $B = 0$, $C = 0$, but the second formula is false.

5. Consider the Boolean functions $f(x_1, x_2, x_3)$ and $g(x_1, x_2, x_3)$ realised by the following digital circuits:



Construct the truth-tables for these functions and represent them as formulas over $\{\wedge, \vee, \neg\}$.

Solution. The truth-tables:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	$g(x_1, x_2, x_3)$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

$f(x_1, x_2, x_3)$ is realised by the formula $(x_2 \vee \neg x_3) \wedge x_1$.

$g(x_1, x_2, x_3)$ is realised by the formula $(x_1 \wedge \neg x_3) \vee (x_1 \wedge x_2)$.

You see that the *same function* can be realised by different circuits and formulas. That is why it is important to know how to construct as simple realisations as possible. (See also the next problem.)

6. Construct a Boolean formula that realises the Boolean function given by the following truth-table:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Can you simplify it? (Hint: use $\rightarrow$ and $\neg$.)

Solution. The straightforward method (explained in lecture slides) gives the formula

$$(\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \neg x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_3).$$

However, the following observations show that we can construct a much simpler formula. If x_1 is 0 then $f(0, x_2, x_3) = x_2$, and if x_1 is 1 then $f(1, x_2, x_3) = x_3$. Therefore, we can represent $f(x_1, x_2, x_3)$ as the formula

$$(\neg x_1 \rightarrow x_2) \wedge (x_1 \rightarrow x_3).$$

7. Construct the simplest Boolean formula that realises the Boolean function given by the following truth-table:

x_1	x_2	$f(x_1, x_2)$
0	0	0
0	1	0
1	0	1
1	1	1

Solution. The column for $f(x_1, x_2)$ coincides with the column for x_1. So the simplest formula is x_1.

8. Consider two Boolean formulas and show that they realise the same Boolean function:

$$x_1 \wedge (x_1 \vee x_2) \qquad \text{and} \qquad x_1 \vee (x_1 \wedge x_2).$$

Find the simplest Boolean formula representing this Boolean function.

Solution. Construct the truth-table for each of the functions and see that they are precisely the same:

x_1	x_2	$x_1 \wedge (x_1 \vee x_2)$	$x_1 \vee (x_1 \wedge x_2)$
0	0	0	0
0	1	0	0
1	0	1	1
1	1	1	1

According to the previous solution, the simplest formula is x_1.

9. Show that the Boolean function given by the formula

$$((x_1 \to x_2) \to x_1) \to x_1$$

is equal to 1.

Solution. The most straightforward (but time consuming) way is to construct the truth-table for the formula. Another, more interesting, way is as follows. Suppose that there exists a row in the truth-table where the formula (the function realised by the formula, to be more precise) is 0. Now recall that an implication is 0 if and only if its premise (left-hand side) is 1, while its conclusion (right-hand side) is 0. Then $(x_1 \to x_2) \to x_1$ is 1 and x_1 is 0. But the former is impossible because (since x_1 is 0) $x_1 \to x_2$ is 1. This contradiction shows that our assumption was wrong, and therefore, the function is identically equal to 1.

10. Design a Boolean circuit to input a 3-bit value and output its two's complement value (see lecture 2, page 9).

Solution. The two's complement of an n-bit binary number N is $2^n - N$. To compute it, use the algorithm on page 9, lecture 2: invert and add 1. The truth-table for $n = 3$ is as follows:

x_1	x_2	x_3	y_1	y_2	y_3
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	0	1	0
1	1	1	0	0	1

Given the input 3-bit number x_1, x_2, x_3, the outputs y_1, y_2 and y_3 can be computed as follows:

$$y_3 = x_3, \quad y_2 = x_2 \oplus x_3, \quad y_1 = x_1 \oplus (x_2 \vee x_3).$$

11. Is it possible to realise $\neg x$ as a formula with $\wedge$ and $\vee$ only?

    Solution. It is impossible: every formula of one variable x over $\{\wedge, \vee\}$ is equivalent to x. But we need $\neg x$.

12. Given the machine 32-bit word

    1011 1111 1110 0000 0000 0000 0000 0000

    find the decimal number represented by this word assuming that it is

    - a two's complement integer
    - an unsigned integer
    - a single precision IEEE 754 floating-point number.

    Solution:

    - two's complement integer: $-2^{31} + 2^{29} + 2^{28} + 2^{27} + 2^{26} + 2^{25} + 2^{24} + 2^{23} + 2^{22} + 2^{21}$
    - unsigned integer: $2^{31} + 2^{29} + 2^{28} + 2^{27} + 2^{26} + 2^{25} + 2^{24} + 2^{23} + 2^{22} + 2^{21}$
    - single precision IEEE 754 floating-point number: $-1.11_2 \times 2^{(127-127)} = -1.11_2 = -1.75_{10}$.

13. Represent the number -12 as a two's complement 32-bit binary number and as an IEE 754 32-bit floating-point number. Represent -0.125 as an IEE 754 32-bit floating-point number.

    Solution: $-12_{10} = -16_{10} + 4 = -2^4 + 2^2$. So the two's complement representation of -12 is 10100_2 if we have 5-bit words and, for 32-bit words:

    1111 1111 1111 1111 1111 1111 1111 0100

    The IEEE 754 representation is defined by $-12_{10} = (-1)^1 \times 1.1_2 \times 2^3$:

    1 10000010 10000000000000000000000

    -0.125 as an IEEE 754 floating-point number: $-0.125 = -1 \times 2^{-3}$. So Sign = 1, Fraction = 0, Exponent $= -3 + 127 = 124$, and

    1011 1110 0000 0000 0000 0000 0000 0000

14. Draw a Boolean circuit for the logical equation

    $$C \;=\; (A \wedge \neg B) \vee (A \wedge B).$$

    What does this circuit compute?

    Solution: It computes $C = A$ because

    $$(A \wedge \neg B) \vee (A \wedge B) \;\equiv\; A \wedge (\neg B \vee B) \;\equiv\; A \wedge 1 \;\equiv\; A.$$

15. Convert the following decimal numbers to their binary equivalents: 1.5_{10}, 1.1_{10}, $(1/3)_{10}$.

    Convert the following binary numbers to their decimal equivalents: 1.1_2, 101.101_2.

    Solution: $1.5_{10} = 1.1_2$, $1.1_{10} = 1.00011001100110011\ldots$, $1/3$ is tough:

    $1/3 = 0\times(1/2)+1\times(1/4)+0\times(1/8)+1\times(1/16)+0\times(1/32)+1\times(1/64)+\cdots = 0.01010101\ldots$

    $1.1_2 = 1 \times 2^0 + 1 \times 2^{-1} = 1.5_{10}$,
    $101.101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 5 + 5/8 = 5.625_{10}$

16. What decimal number does this two's complement binary number represent?

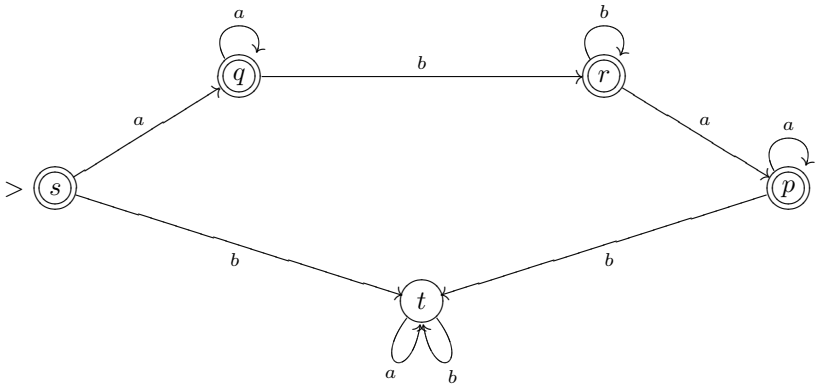
    1111 1111 1111 1111 1111 1110 0000 1100

    Solution: Suppose it represents N. Then $-N$ is represented by

    $$0000\ 0000\ 0000\ 0000\ 0000\ 0001\ 1111\ 0100_2$$

    which is 500_{10}. So $N = -500$.

17. Consider the following DFA:



    (a) Give the formal description of the automaton (using a transition table).

    (b) Find the computation of the automaton on the input string *aaabba* and determine if the string is accepted.

    (c) Find the computations of the automaton on the input strings *aabaab* and *aabbaab* and determine if the strings are accepted.

    (d) Describe (informally) the language accepted by the automaton.

    Solution:

    (a) $(Q, \Sigma, \delta, s, F)$ where $Q = \{s, q, r, p, t\}$, $\Sigma = \{a, b\}$, $F = \{s, q, r, p\}$ and the transition function δ is as follows:

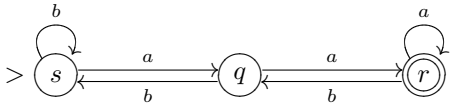
δ	a	b
s	q	t
q	q	r
r	p	r
p	p	t
t	t	t

    (b) $(s, aaabba)$, $(q, aabba)$, $(q, abba)$, (q, bba), (r, ba), (r, a), (p, ε); the string is accepted.

    (c) $(s, aabaab)$, $(q, abaab)$, $(q, baab)$, (r, aab), (p, ab), (p, b), (t, ε); the string is not accepted. $(s, aabbaab)$, $(q, abbaab)$, $(q, bbaab)$, $(r, baab)$, (r, aab), (p, ab), (p, b), (t, ε); the string is not accepted.

    (d) The language of this automaton consists of the empty string ε and all the strings of the form "a string of as followed by a (possibly empty) string of bs followed by a (possibly empty) string of as".

18. Consider the following DFA:



(a) Give the formal description of the automaton (using a transition table).

(b) Find the computations of the automaton on the input strings *aababa* and *bbaababaaa* and determine if the strings are accepted. Is it true that the automaton accepts all strings of length ≥ 2 ending with an a?

(c) Find the computations of the automaton on the input strings *aabaab* and *bbaababb* and determine if the strings are accepted.

(d) Describe the language accepted by the automaton.

Solution:

(a) $(Q, \Sigma, \delta, s, F)$ where $Q = \{s, q, r\}$, $\Sigma = \{a, b\}$, $F = \{r\}$ and the transition function δ is as follows:

δ	a	b
s	q	s
q	r	s
r	r	q

(b) $(s, aababa)$, $(q, ababa)$, $(r, baba)$, (q, aba), (r, ba), (q, a), (r, ε); the string is accepted.
$(s, bbaababaaa)$, $(s, baababaaa)$, $(s, aababaaa)$, $(q, ababaaa)$, $(r, babaaa)$, $(q, abaaa)$, $(r, baaa)$, (q, aaa), (r, aa), (r, a), (r, ε); the string is accepted.
No, it is not true, e.g. *aba* is not accepted.

(c) $(s, aabaab)$, $(q, abaab)$, $(r, baab)$, (q, aab), (r, ab), (r, b), (q, ε); the string is not accepted.
$(s, bbaababb)$, $(s, baababb)$, $(s, aababb)$, $(q, ababb)$, $(r, babb)$, (q, abb), (r, bb), (q, b), (s, ε); the string is not accepted.

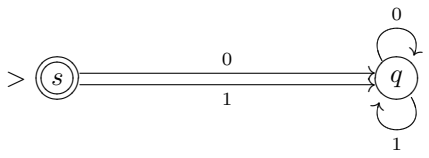
(d) The language of this automaton is

$$\{waa(ba)^n a^m \mid w \text{ is any word of } a\text{s and } b\text{s, and } n, m \geq 0\}.$$

In other words, this language consists of all the strings of the following form: "take any word of as and bs which is followed by aa, then by any (possibly zero) number of repetitions of ba, then by a (possibly empty) string of as".

19. Is there a deterministic finite automaton A accepting the empty word ε? Is there an A over the alphabet $\{0, 1\}$ such that $L(A) = \{\varepsilon\}$? Explain your answers.

Solution: Every automaton whose initial state is a favourable state accepts ε. For example, the following automaton is such that its language is $\{\varepsilon\}$:



20. For each of the following languages, construct (in the form of a state transition diagram) a deterministic finite automaton accepting it.

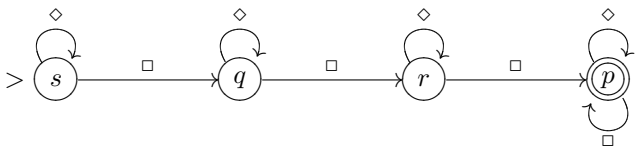
(a) all strings over $\{\square, \diamond\}$ containing at least three $\square$s

(b) all strings over $\{\square, \diamond\}$ that begin with a $\square$ and end with a $\diamond$

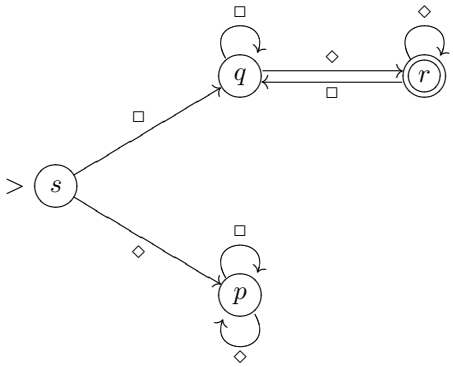
(c) all strings over $\{0, 1\}$ except the empty string ε.

Solution:

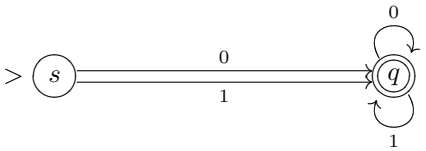
(a)



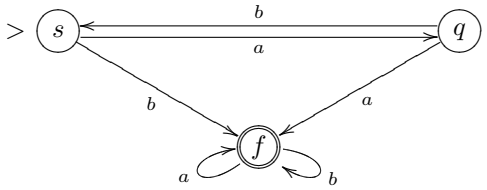
(b)



(c)



21. Consider the following finite automaton:



(a) Is the automaton deterministic or nondeterministic?

(b) Find the computation(s) of the automaton on the input strings *ababab* and *ababa*, and determine if the strings are accepted.

(c) Find the computation(s) of the automaton on the input strings *ababb*, *ba*, and *aa*, and determine if the strings are accepted.

(d) Describe (informally) the language accepted by the automaton.

Solution:

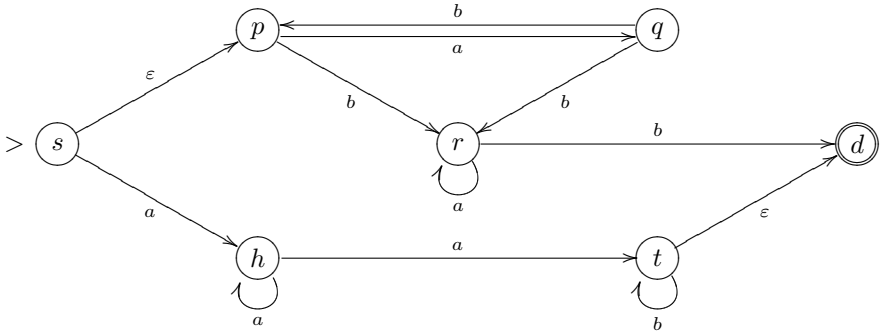
(a) It is deterministic.

(b) $(s, ababab)$, $(q, babab)$, $(s, abab)$, (q, bab), (s, ab), (q, b), (s, ε); the string is not accepted.
$(s, ababa)$, $(q, baba)$, (s, aba), (q, ba), (s, a), (q, ε); the string is not accepted.

(c) $(s, ababb)$, $(q, babb)$, (s, abb), (q, bb), (s, b), (f, ε); the string is accepted.
(s, ba), (f, a), (f, ε); the string is accepted.
(s, aa), (q, a), (f, ε); the string is accepted.

(d) The language consists of all the strings that either start with a b, or contain two consecutive a's or two consecutive b's.

22. Determine if the nondeterministic automaton



accepts the strings

(a) *abb*

(b) *abbbb*

(c) *aabb*

In each case, if the string is accepted, give a corresponding computation.

Solution:

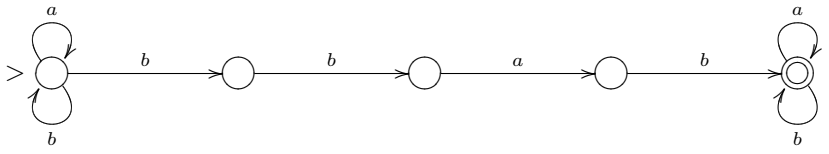
(a) *abb* is accepted. A corresponding computation on input *abb* is:
(s, abb), (p, abb), (q, bb), (r, b), (d, ε)

(b) *abbbb* is not accepted (every computation on it is stuck somewhere).

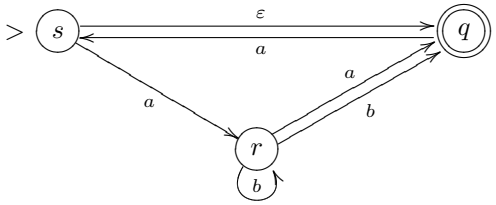
(c) *aabb* is accepted. A corresponding computation on input *aabb* is:
$(s, aabb)$, (h, abb), (t, bb), (t, b), (t, ε), (d, ε)

23. Design a (deterministic or nondeterministic) finite automaton A such that $L(A)$ consists of all the strings of a's and b's that contain *bbab* as a substring.

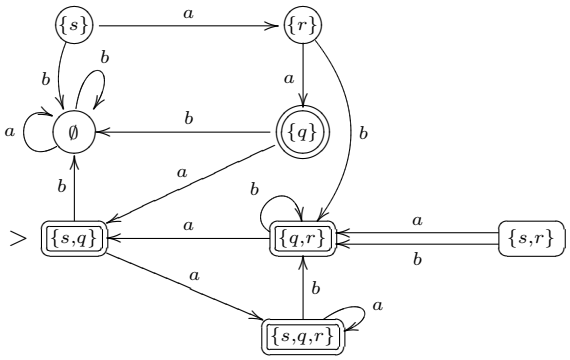
Solution:



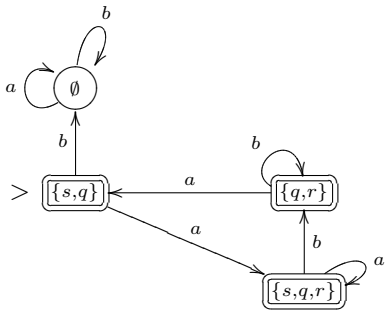
24. Transform, using the subset construction, the following nondeterministic finite automaton into an equivalent deterministic finite automaton. Then delete the unreachable states. Show your working.



Solution:



After removing the unreachable states, we obtain:



25. For each of the following regular expressions, find three of the shortest strings in the corresponding regular language:

    (i) $a^*(b \cup abb)b^*b$

    (ii) $(a \cup ab)(a^* \cup ab)^*b$

    Solution:

    (i) bb, bbb, abb

    (ii) ab, abb, aab

26. For each of the following regular expressions, list three of the shortest strings that are **not** in the corresponding regular language:

    (i) a^*aabb^*

    (ii) $(aa)^*(bba)^*(bb)^*$

    (iii) $a^*(bba)^*b^*$

    Solution:

    (i) ε, a, b

    (ii) a, b, ab

    (iii) ba, aba, bab

27. Let $\Sigma = \{\square, \diamond\}$. For each of the following languages over Σ, find a regular expression representing it:

    (i) All strings that contain exactly one $\square$.

(ii) All strings that begin with □□.

(iii) All strings that begin with □□ and end with ◇◇.

(iv) All strings that contain either the substring □□□ or the substring ◇◇◇.

Solution:

(i) $\diamond^* \square \diamond^*$

(ii) $\square\square(\square \cup \diamond)^*$

(iii) $\square\square(\square \cup \diamond)^*\diamond\diamond$

(iv) $(\square \cup \diamond)^*(\square\square\square \cup \diamond\diamond\diamond)(\square \cup \diamond)^*$

28. Let $\Sigma = \{0, 1\}$. Show that the language

$$L = \{w \mid 0110 \text{ is a suffix of } w\}$$

is regular.

Solution: A regular expression representing L is $(0 \cup 1)^*0110$.

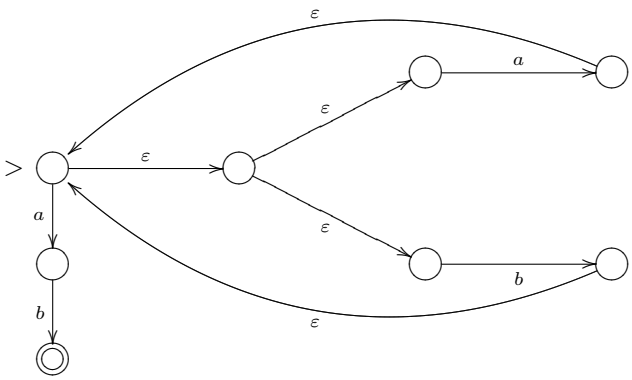
29. Apply the procedure given in the lectures to convert each of the following regular languages to a finite automaton accepting it:

(i) $L[(a \cup b)^*ab]$

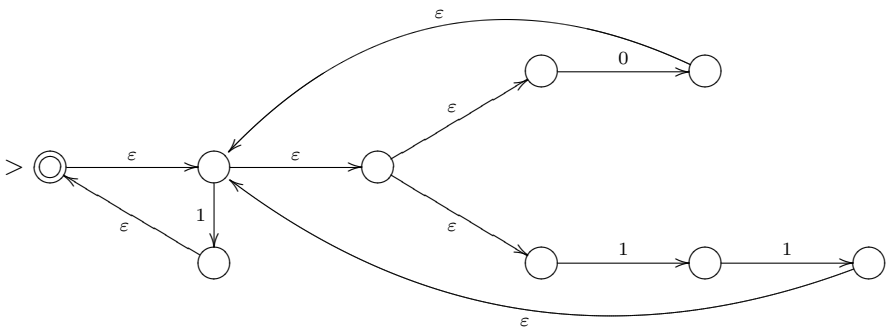
(ii) $L[((0 \cup 11)^*1)^*]$

Solution:

(i)



(ii)



30. Consider the following transition table of a Turing machine:

s	$\sqcup$	h	$\sqcup$
s	a	q	b
s	b	q	a
q	a	s	$\rightarrow$
q	b	s	$\rightarrow$
q	$\sqcup$	q	$\sqcup$
s	$\triangleright$	s	$\rightarrow$
q	$\triangleright$	s	$\rightarrow$

(i) Give the computation of the machine starting with the configuration $(s, \triangleright\underline{a}bbabba)$.

(ii) Give the computation of the machine starting with the configuration $(s, \triangleright\underline{b}aaba)$.

(iii) Experiment with more input words. Then describe in English what this Turing machine does.

Solution:

(i)

$(s, \triangleright\underline{a}bbabba),\ (q, \triangleright\underline{b}bbabba),\ (s, \triangleright b\underline{b}babba),\ (q, \triangleright b\underline{a}babba),\ (s, \triangleright ba\underline{b}abba),\ (q, \triangleright ba\underline{a}abba),$
$(s, \triangleright baa\underline{a}bba),\ (q, \triangleright baa\underline{b}bba),\ (s, \triangleright baab\underline{b}ba),\ (q, \triangleright baab\underline{a}ba),\ (s, \triangleright baaba\underline{b}a),$
$(q, \triangleright baaba\underline{a}a),\ (s, \triangleright baabaa\underline{a}),\ (q, \triangleright baabaa\underline{b}),\ (s, \triangleright baabaab\underline{\sqcup}),\ (h, \triangleright baabaab\underline{\sqcup})$

(ii)

$(s, \triangleright\underline{b}aaba),\ (q, \triangleright\underline{a}aaba),\ (s, \triangleright a\underline{a}aba),\ (q, \triangleright a\underline{b}aba),\ (s, \triangleright ab\underline{a}ba),\ (q, \triangleright ab\underline{b}ba),\ (s, \triangleright abb\underline{b}a),$
$(q, \triangleright abb\underline{a}a),\ (s, \triangleright abba\underline{a}),\ (q, \triangleright abba\underline{b}),\ (s, \triangleright abbab\underline{\sqcup}),\ (h, \triangleright abbab\underline{\sqcup})$

(iii) The Turing machine scans the input step by step from left to right and replaces every symbol a by b and every symbol b by a.

31. Let $\Sigma = \{a, b, \triangleright, \sqcup\}$. Give the complete transition tables for the following Turing machines having tape alphabet Σ:

(i) The machine scans the tape to the right until it finds the first blank cell, then halts.

(ii) The machine scans the tape to the right until it finds the first pattern ab (i.e., an a followed immediately by a b), then halts. If there is no such pattern, then the machines does not terminate.

Solution: There are several possible solutions, here are just some:

(i)

s	a	s	$\rightarrow$
s	b	s	$\rightarrow$
s	$\sqcup$	h	$\sqcup$
s	$\triangleright$	s	$\rightarrow$

(ii)

s	a	q	$\rightarrow$
s	$\sqcup$	s	$\rightarrow$
s	b	s	$\rightarrow$
q	b	h	b
q	a	s	a
q	$\sqcup$	s	$\rightarrow$
s	$\triangleright$	s	$\rightarrow$
q	$\triangleright$	q	$\rightarrow$

32. Give the complete transition table of a Turing machine which computes the function

$$f(w) = \begin{cases} w & \text{if } |w| \text{ is even,} \\ wa & \text{if } |w| \text{ is odd,} \end{cases}$$

defined for all strings of a's. (Note that 0 counts as even.)

Solution: Again, there are several possible solutions, here is one:

s	$\sqcup$	h	$\sqcup$
s	a	q_{odd}	$\rightarrow$
q_{odd}	a	s	$\rightarrow$
q_{odd}	$\sqcup$	h	a
s	$\triangleright$	s	$\rightarrow$
q_{odd}	$\triangleright$	s	$\rightarrow$

33. (i) Give an implementation level description in English of a Turing machine that computes the $\mathbb{N} \rightarrow \mathbb{N}$ function

$$f(n) = \begin{cases} n & \text{if } n \geq 4 \\ 0 & \text{otherwise.} \end{cases}$$

(ii) Give the complete transition table of this Turing machine.

Solution:

(i) There are many possible solutions, here is just one:

(1) It checks whether the leftmost symbol of the input is 0. If yes, halts.

(2) If not, it scans the tape to the right until it reaches a blank. In the meanwhile, it memorises (with the help of a state) whether the length of the input string is > 2 (this would correspond to the input number being ≥ 4).

(3) If after reaching the first blank the length of the input is > 2, then halts.

(4) If after reaching the first blank the length of the input is ≤ 2, then erases the input, writes 0, and halts.

(ii) The transition table corresponding to the above informal description is:

s	0	h	0
s	1	q_1	$\rightarrow$
q_1	$\sqcup$	$back$	$\leftarrow$
q_1	0	q_2	$\rightarrow$
q_1	1	q_2	$\rightarrow$
q_2	$\sqcup$	$back$	$\leftarrow$
q_2	0	q_3	$\rightarrow$
q_2	1	q_3	$\rightarrow$

q_3	0	q_3	$\rightarrow$
q_3	1	q_3	$\rightarrow$
q_3	$\sqcup$	h	$\sqcup$
$back$	0	$back$	$\sqcup$
$back$	1	$back$	$\sqcup$
$back$	$\sqcup$	$back$	$\leftarrow$
$back$	$\triangleright$	$back^+$	$\rightarrow$
$back^+$	$\sqcup$	h	0

The other transitions are arbitrary (of course $\ldots, \triangleright$ should always go to $\ldots, \rightarrow$).