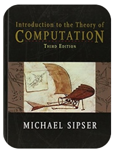
Ciências / Ciência da computação / Introduction to the Theory of Computation (3rd Edition)

Exercício 5

Capítulo 1, Página 84



Introduction to the Theory of Computation

ISBN: 9781133187790

Índice

Solução Certificado Solução fornecida há 1 ano

Passo 1

1 de 10

In part a) we are going to explain the procedure of constructing a DFA for a given language, from the DFA of its complement. In other parts construction is analogous, so we skip it and present final solution. First we consider the general procedure.

Ok, let's think about this for a moment. Suppose we have a DFA $M = (Q, \Sigma, \delta, q_0, F)$ which recognizes language L, and for some strange reason want to transform this machine into new machine M' which recognizes the complement L^c of language L. What are we to do?

First we need to understand what we want our new machine M' to do. By definition of complement, it has to $\textbf{accept}$ those and only those words which machine M $\textbf{does not accept}$.

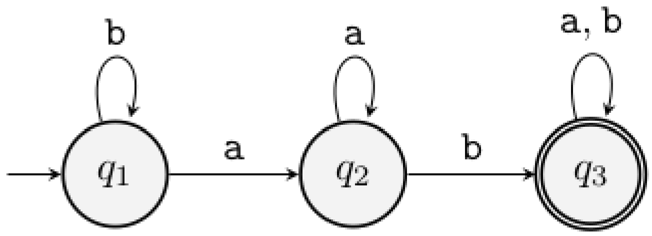
But when does machine M accept some word w? When it ends its computation on w in accept state, of course! So, we will have to play around with accept states, cool.

Here is the big idea: what if we are to trick the machine M by changing every its accept state to non-accepting and vice-versa!? One thing is sure, this would definitely confuse the machine! How much, you ask. Well, exactly as much as we would like to, say I!

Indeed, if we define new machine as $M' = (Q, \Sigma, \delta, q_0, Q \setminus F)$ (note that the only difference from M is the set of accepting states), we see that language which M' recognizes is exactly L^c. Since Q, δ and q_0 are not changed, computation of M' proceeds identically to that of M on any given word. Hence, word which brings M to accepting state (i.e.\ $w \in L(M)$) brings M' to that same, but now non-accepting, state (i.e.\ $w \notin L(M')$).

Passo 2

2 de 10

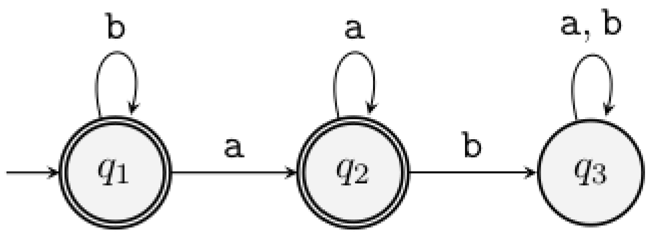


$\boxed{\textbf{Part (a)}}$

Given language is $\{w \mid w \text{ does not contain the substring } \texttt{ab}\}$. We see that this is the complement of language which contains exactly those words which $\textbf{do contain}$ substring **ab**. Machine which recognizes this language is easy to construct, the diagram is presented below.

Passo 3

3 de 10

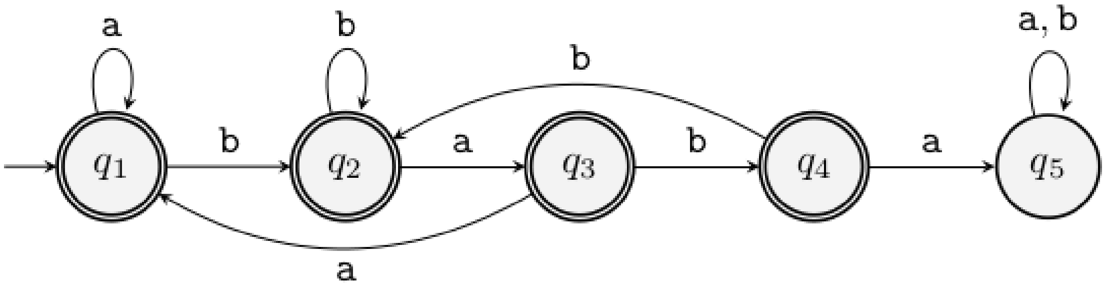


So now, as explained above, we only need to switch the accepting states to non-accepting ones, and vice-versa. Final machine, which recognizes given language, is presented here.

Passo 4

4 de 10

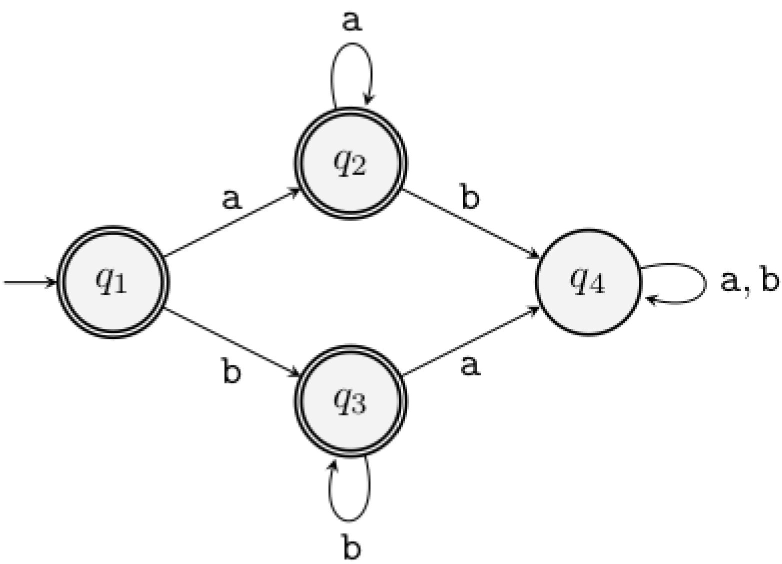
$\boxed{\textbf{Part (b)}}$



Passo 5

5 de 10

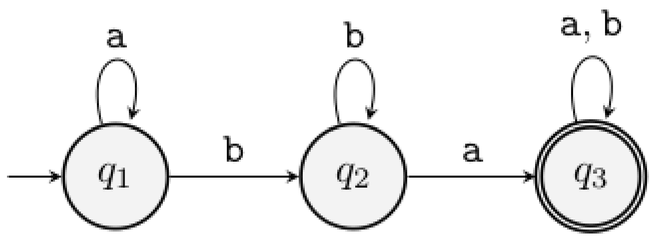
$\boxed{\textbf{Part (c)}}$



Passo 6

6 de 10

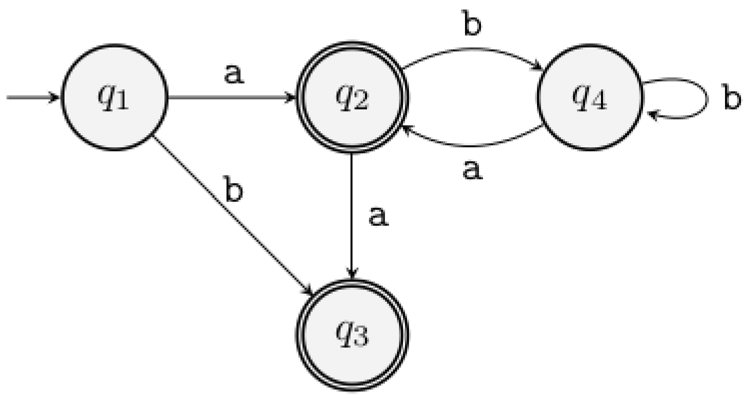
$\boxed{\textbf{Part (d)}}$



Passo 7

7 de 10

$\boxed{\textbf{Part (e)}}$



Passo 8

8 de 10

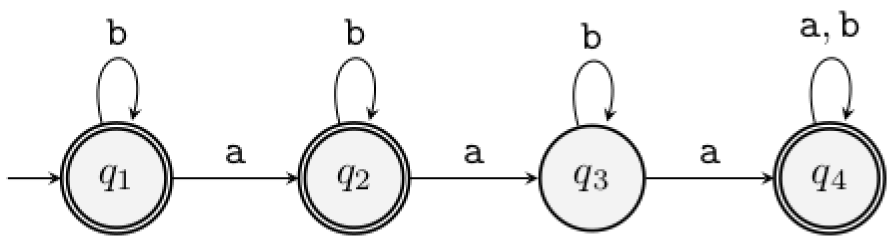
$\boxed{\textbf{Part (f)}}$

Language in this part is the complement of language from part c (can you see why?).

Passo 9

9 de 10

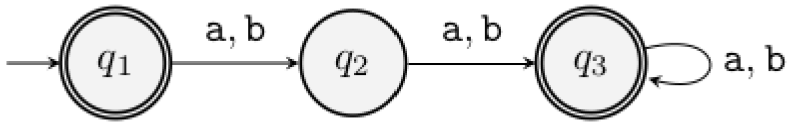
Part (g)



Passo 10

10 de 10

Part (h)



Avaliar esta solução

Exercício 4

Exercício 6

Privacidade Termos de serviço

Português (BR)