

# LCMC

## DFA

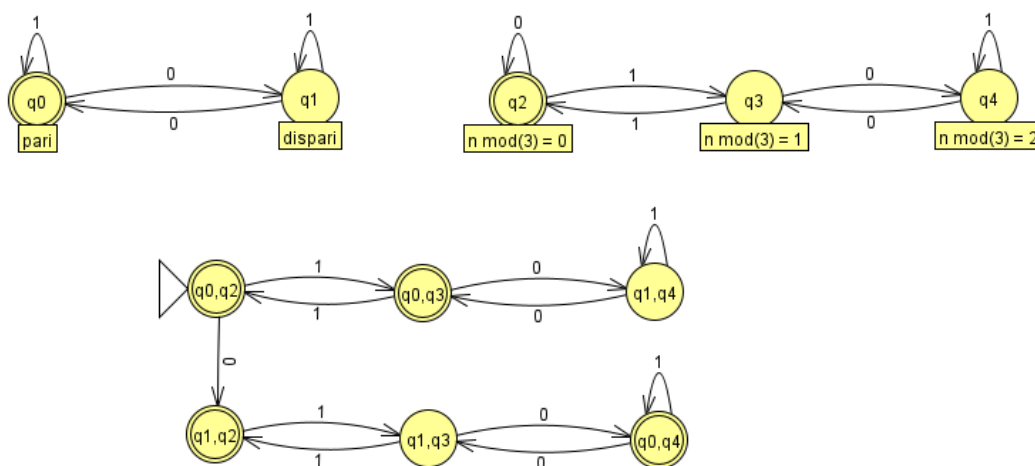
Creazione DFA a partire da un linguaggio

Aggiornamento: 2022-12-14, esercizi fino gennaio 2016

Raccolta di esercizi (prevalentemente di esami) in cui viene richiesto di creare dei DFA.

### Simulazione-Esame-2022-12-14

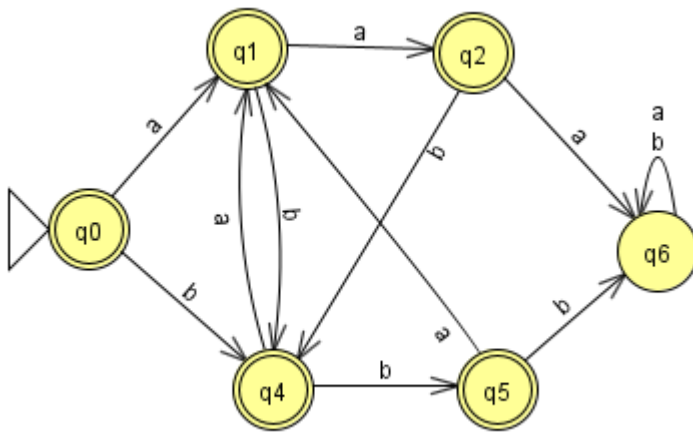
Esercizio 1: Definire un DFA che riconosce il linguaggio delle stringhe binarie che contengono un numero pari di 0 oppure che interpretate come numero in base 2 rappresentano un numero multiplo di 3



In questa risoluzione è mostrato come unire due automi date le specifiche dell'esercizio. **Gli stati iniziali degli automi parziali sono  $q_0$  e  $q_2$ .** L'automata finale racchiude le coppie degli stati di quelli iniziali e ci si muove fra le coppie considerando le transizioni di entrambi per "0" e per "1". Gli stati finali sono quelli che possiedono nella coppia uno degli stati finali degli automi parziali, quindi  $q_0$  e  $q_2$ . Il secondo parziale si muove con gli "0" e "1" tenendo conto che ad un numero binario se aggiungiamo uno "0" allora raddoppieremo il resto precedente andando a raddoppiare il numero; mentre aggiungendo "1" andiamo a raddoppiare e a sommare un'unità.

### Esame-2020-Gennaio-31-dfa

Costruire un DFA per il linguaggio  $L$  su alfabeto  $\{a,b\}$  delle stringhe  $w$  in cui lo stesso simbolo non appare mai tre volte consecutivamente. Ad esempio **aabbaba**, **ababb**,  **$\epsilon$**  appartengono ad  $L$  mentre **aaab**, **aaaa**, **aaababbb** non vi appartengono. Nel realizzare il DFA fare attenzione ad utilizzare un numero minimo di stati (non si richiede, però, di verificare la effettiva minimalità con l'algoritmo di minimizzazione).



## Esame-2020-Settembre-08-dfa

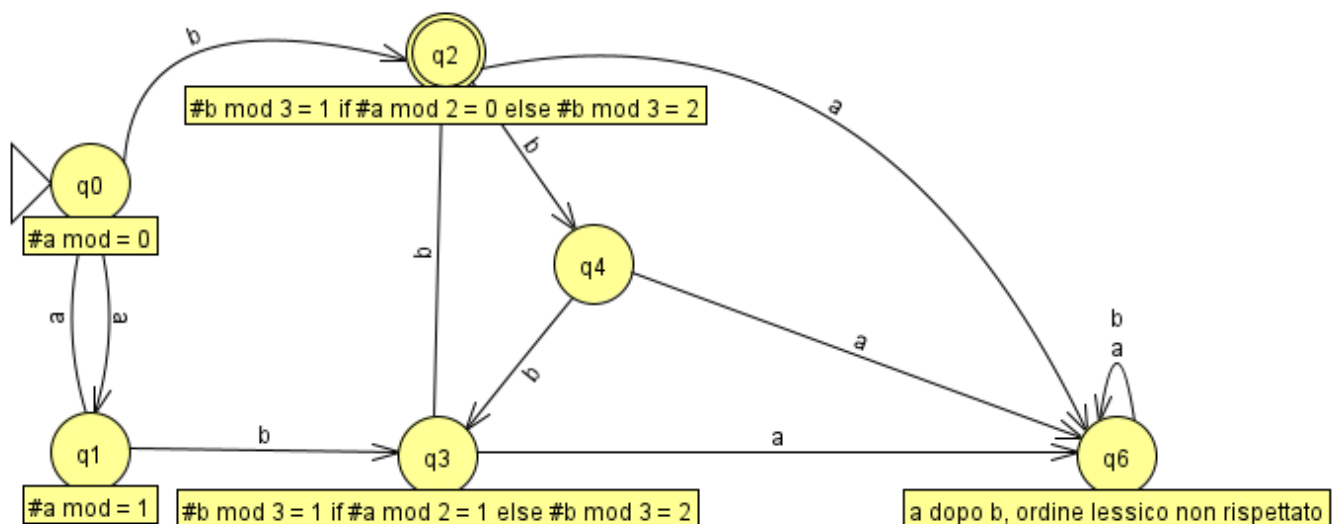
### Esercizio-1

Definire (direttamente, senza fare trasformazioni) un'espressione regolare che rappresenti il linguaggio L su alfabeto  $\{a,b\}$  delle stringhe  $a^n b^m$ , con  $n,m \geq 0$ , per cui valga: se  $n \bmod 2 = 0$  allora  $m \bmod 3 = 1$ , se invece  $n \bmod 2 = 1$  allora  $m \bmod 3 = 2$ . Ad esempio aab, b, bbbb, aaabb, abbbbbb, appartengono al linguaggio L mentre  $\epsilon$ , a, aa, bb, bbb, baa, aabb, aaab, non vi appartengono. Trasformare, poi, l'espressione regolare in  $\epsilon$ -NFA utilizzando la trasformazione vista a lezione (cioè quella definita nel libro di testo).

### Esercizio-2

Costruire (direttamente, senza fare trasformazioni) un DFA per il linguaggio L dell'esercizio 1. Nel realizzare il DFA fare attenzione ad utilizzare un numero minimo di stati: verificare la effettiva minimalità tramite l'algoritmo riempi-tabella (utilizzare una crocetta di tipo diverso ad ogni passata).

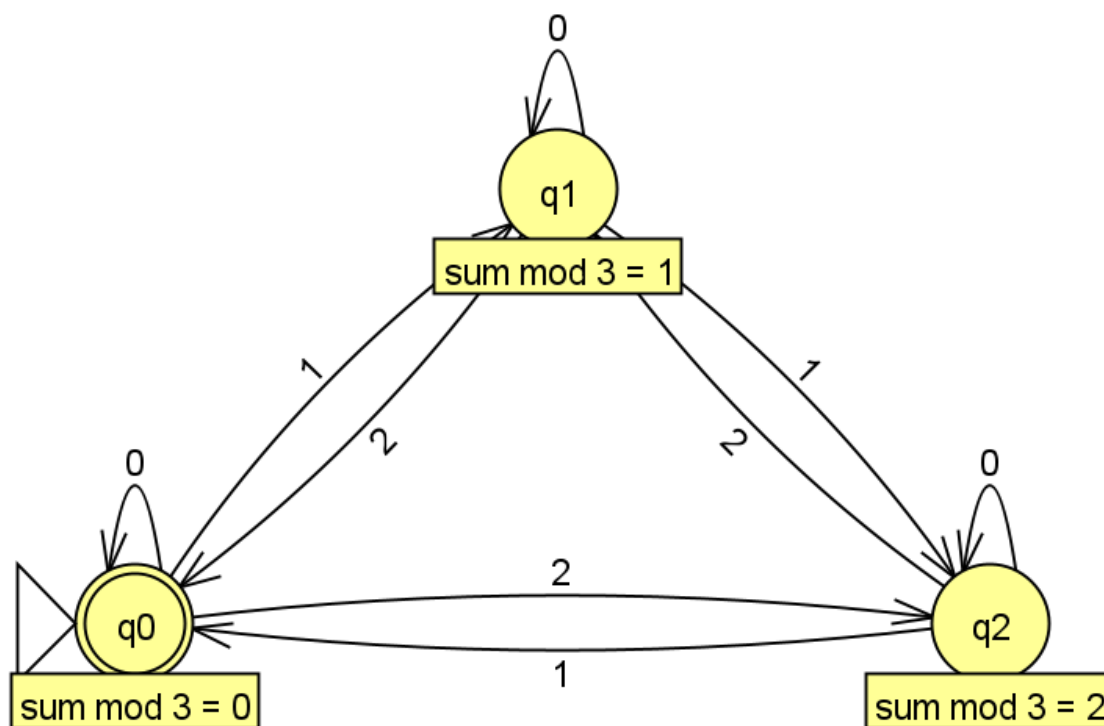
*Sono riportati entrambi gli esercizi dato che le specifiche del linguaggio di quello relativo al DFA era nell'esercizio precedente*



Gli stati di **b** (**q2,q3,q4**) rappresentano i tre possibili moduli differentemente in base a come entriamo nel ciclo: se arriviamo in **q2** da **q0** allora **q2** indicherà una sequenza di **b** con modulo uguale a 1, se arriviamo da **q1** a **q3** questo ruolo lo avrà **q3**. Lo stato **q6** viene raggiunto quando durante le letture di **b** incontriamo una **a** che ci porterebbe ad avere inevitabilmente una stringa che non può più far parte del linguaggio.

### Esame-2019-Luglio-09-dfa

Costruire un DFA per il linguaggio L su alfabeto  $\{0,1,2\}$  che contiene le stringhe **w** in cui la somma dei numeri (0, 1 o 2) contenuti in **w** sia un multiplo di 3. Ad esempio 22101, 1011, 0021, 211110, 00000, 0,  $\epsilon$  appartengono ad L mentre 1, 2, 110, 010, 22 non vi appartengono. Nel realizzare il DFA fare attenzione ad utilizzare il numero minimo di stati. In seguito, verificare che il DFA realizzato sia effettivamente minimo usando l'algoritmo riempi-tabella visto a lezione (cioè quello definito nel libro di testo). Mostrare la tabella, utilizzando una crocetta di tipo diverso ad ogni passata.



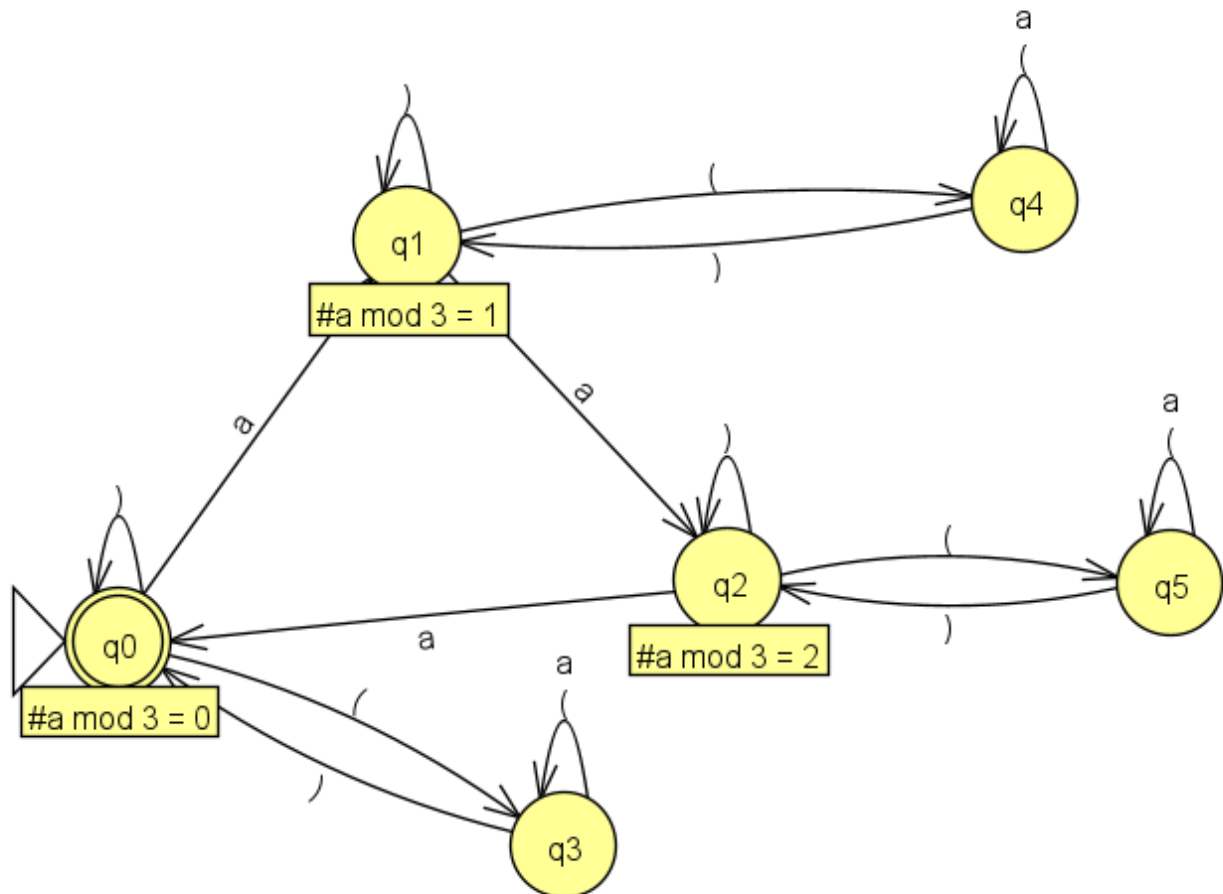
L'esercizio si può risolvere come altri che sono stati visti dove gli stati rappresentano il risultato dei moduli. In questo caso durante la consumazione della stringa ci spostiamo lungo tali risultati possibili 0, 1, 2:

- 0 non porta a cambiare stato dato che la somma non cambia;
- 1 aumenta il risultato del modulo di un'unità
- 2 aumenta il risultato del modulo di due unità

### Esame-2019-Febbraio-02-dfa

Costruire un DFA per il linguaggio L su alfabeto  $\{a,(,)\}$  in cui il numero delle **a** sia un multiplo di 3 ignorando tutto quello che c'è tra una parentesi tonda aperta (incontrata leggendo la stringa da sinistra a destra) e la

prima parentesi tonda chiusa che la segue (o la fine della stringa in caso non ci sia una parentesi tonda chiusa che la segue). Le parentesi tonde, cioè, sono come delimitatori di commenti di un linguaggio di programmazione, dentro cui le *a* non vengono contate: una parentesi tonda aperta apre il commento e la prima parentesi tonda chiusa che lo segue chiude il commento. Ad esempio  $a()a()()a$ ,  $a(a)aaa(a)aa$ ,  $a)a))a$ ,  $\epsilon$ ,  $a((a)aa$ ,  $a(a)aa(a, ((( ))((aa, (a))$  appartengono ad  $L$  mentre  $aa(a, (aa)a$  non vi appartengono (le parti ignorate, che non contribuiscono al conteggio delle *a*, sono in *italic> solo per chiarire il motivo dell'appartenenza o meno).*



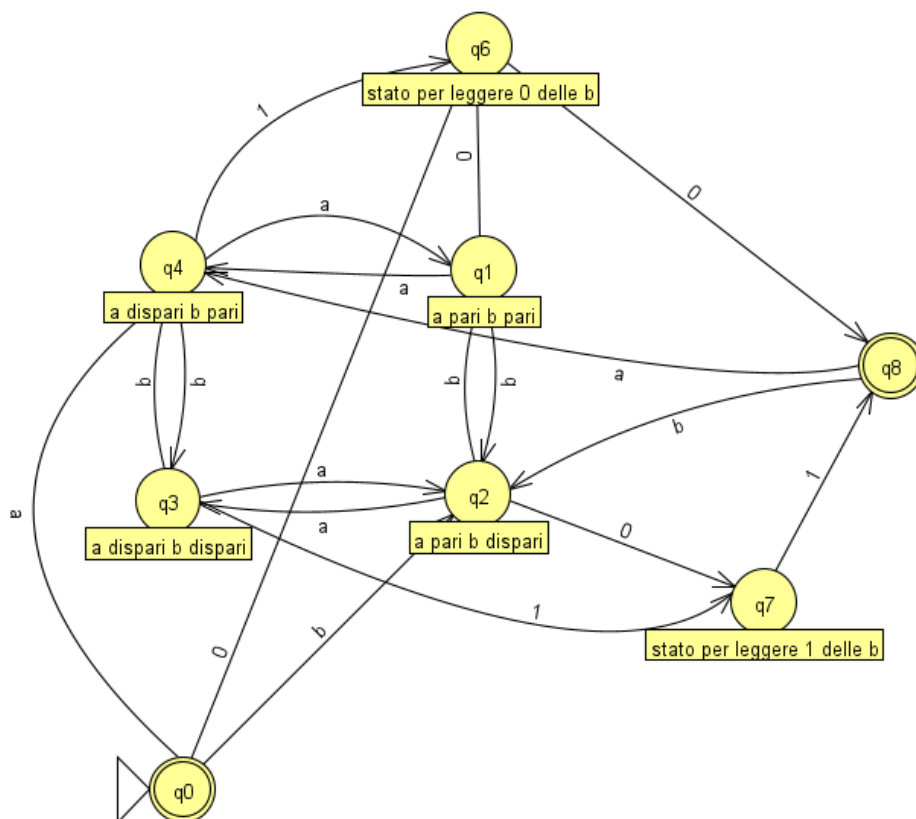
Anche in questo caso come negli esercizi precedenti il concetto di "multiplo" è stato trattato sfruttando i moduli. Seguendo il percorso  $q1 \rightarrow q2 \rightarrow q3$  incrementiamo la cardinalità delle "a" nella stringa e quindi anche il modulo. Da ogni stato se leggiamo un "(" entriamo in un corrispettivo stato che permette di non prendere in considerazione ulteriori "a" e "(" fino a che non troveremo una ")". L'unico differente è  $q3$  che rappresenta anch'esso uno stato finale dato che il numero di "a" che già avevamo prima di arrivarci soddisfaceva le condizioni del linguaggio.

### Esame-2019-Gennaio-15-dfa

Costruire un DFA per il linguaggio  $L$  su alfabeto  $\{a, b, 0, 1\}$  (composto dalle lettere "a" e "b" e dai numeri "0" e "1"), definito come segue. Una stringa  $w \in L$  se e solo se ogni suo gruppo di lettere è seguito da due numeri  $n_1, n_2 \in \{0, 1\}$  tali che, indicato con  $k_a$  il numero di "a" nel gruppo e con  $k_b$  il numero di "b" nel gruppo:  $n_1 = 0$  se  $k_a$  è pari ( $n_1 = 1$  altrimenti) e  $n_2 = 0$  se  $k_b$  è pari ( $n_2 = 1$  altrimenti). Ad esempio  $aaabab10ab11abba00 \in L$  (il primo gruppo di lettere ha un numero dispari di "a" ed un numero pari di "b", come indicato da "10", il secondo un numero dispari sia di "a" che di "b", cioè "11", ed il

terzo un numero pari sia di "a" che di "b", cioè "00"). Altri esempi di stringhe in L sono: aab01, aaa10b01, 0100, ε, abb1011, aab010b01. Esempi di stringhe non in L sono: ab10, aabb0, a, aaab1ab11. Nel costruire il DFA si eviti di introdurre stati tra loro equivalenti: lo si costruisca il più piccolo possibile.

La soluzione presentata tiene conto della struttura del linguaggio e non tanto alle stringhe date come esempio che non si sa se essere effettivamente corrette date le specifiche. Ad esempio 0100, abb1011, aab010b01 non rispettano le regole che traspaiono dal linguaggio.



Le transizioni che non sono presenti nell'automa portano allo stato pozzo, il quale non è stato messo per rendere più leggibile lo schema. Spostandosi fra q1 q2 q3 q4 è possibile ritrovarsi nelle condizioni in cui la stringa, precedente alla coppia di numeri, abbia:

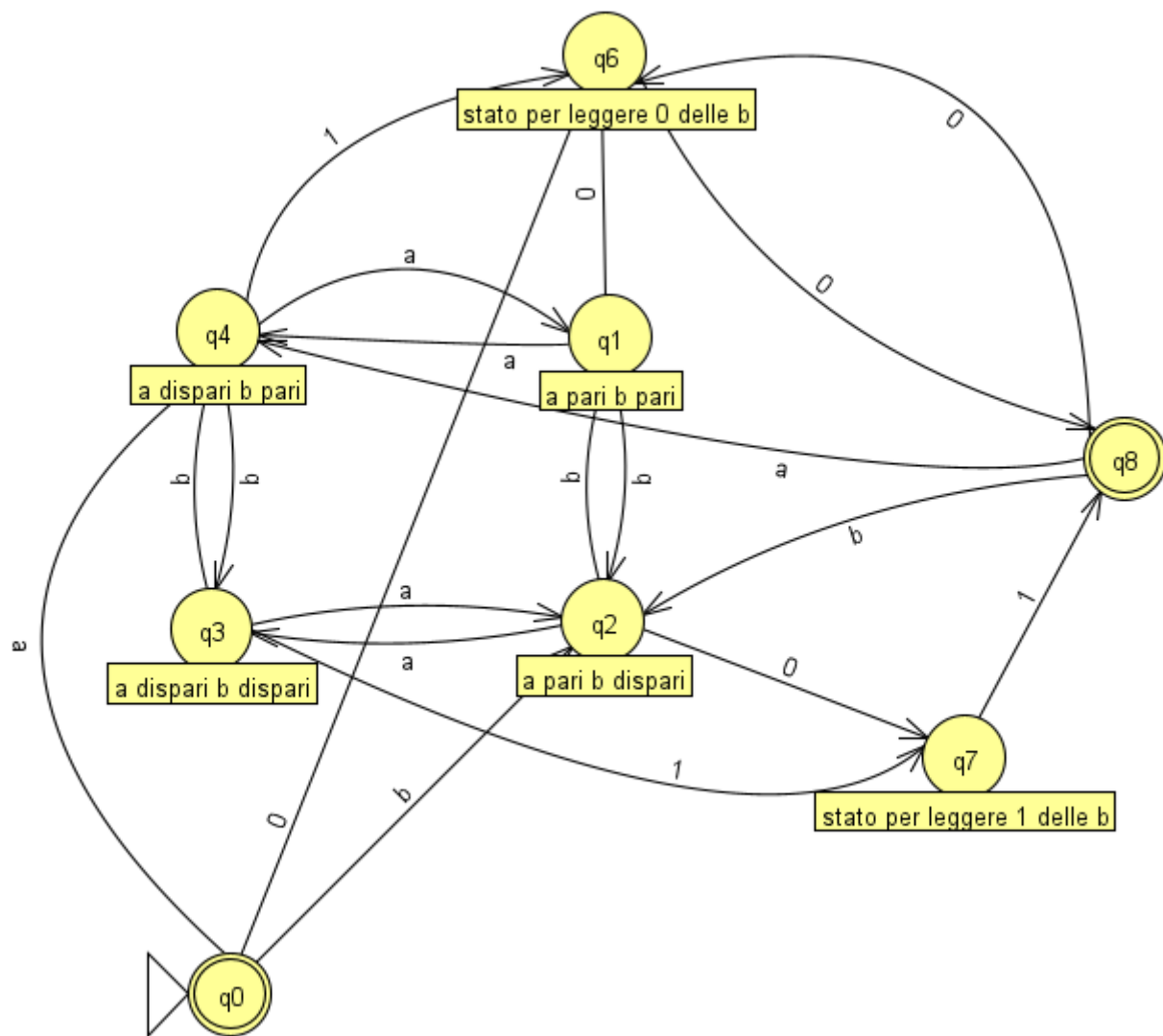
- un numero di "a" pari e "b" pari
- un numero di "a" pari e "b" dispari
- un numero di "a" dispari e "b" pari
- un numero di "a" dispari e "b" dispari

In base allo stato di questi da cui usciamo possiamo andare in q6 o q7:

- q6 sta ad indicare una situazione in cui il numero di "b" è pari e ciò comporta, dopo avere letto il numero relativo ad "a", di dovere leggere uno zero
- q7 è speculare, quindi relativo ad un numero di b dispari In entrambi gli stati possiamo arrivare sia con un 1 che con uno 0 in base al fatto che le "a" siano dispari o pari

Infine, da q8 ci muoviamo come dallo stato iniziale andando in una situazione di "a" o "b" dispari.

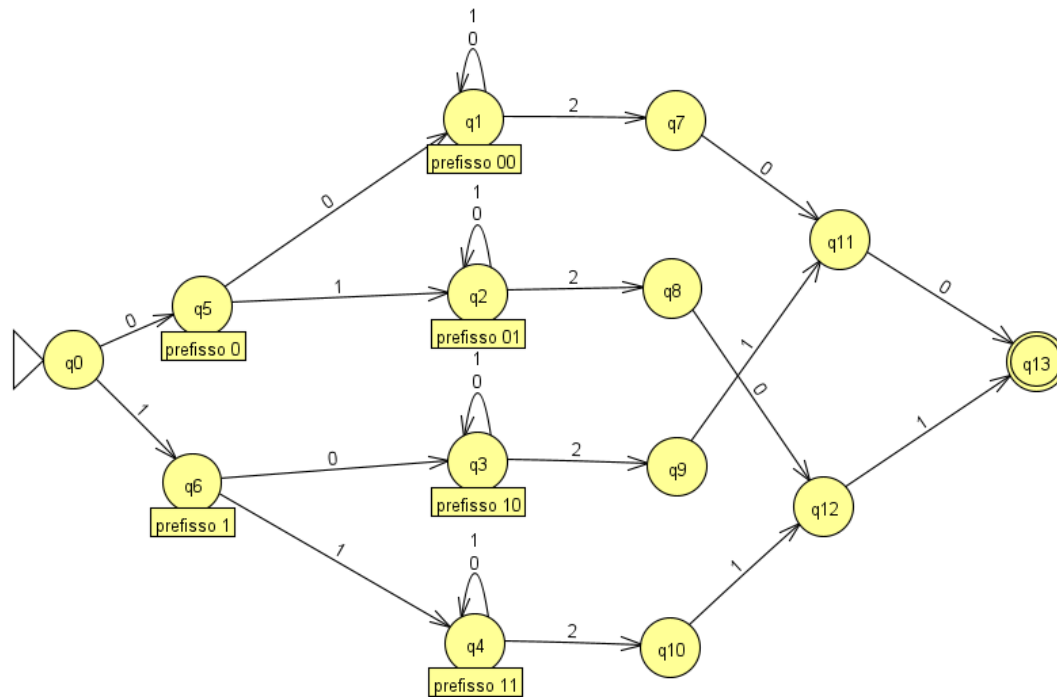
Non è chiaro se una stringa possa avere una coppia di zeri dopo la precedente coppia di numeri come ad esempio ab1100. In caso lo fosse questo secondo automa lo permette.



In questo caso abbiamo un'ulteriore transizione che va da q8 a q6 leggendo uno zero

### Esame-2018-Gennaio-16-dfa

Esercizio 1. Realizzare un DFA che riconosca il linguaggio su alfabeto  $\{0,1,2\}$  che contiene le stringhe  $\$x2y\$$ , dove  $\$x\$$  e  $\$y\$$  sono stringhe su  $\{0,1\}$  tali che:  $\$x\$$  ha lunghezza almeno 2 e  $\$y\$$  è il prefisso di lunghezza 2 di  $\$x\$$ . Ad esempio 1011210, 01201 ed 11000211 appartengono al linguaggio mentre 1210, 110 ed 0120 non vi appartengono.

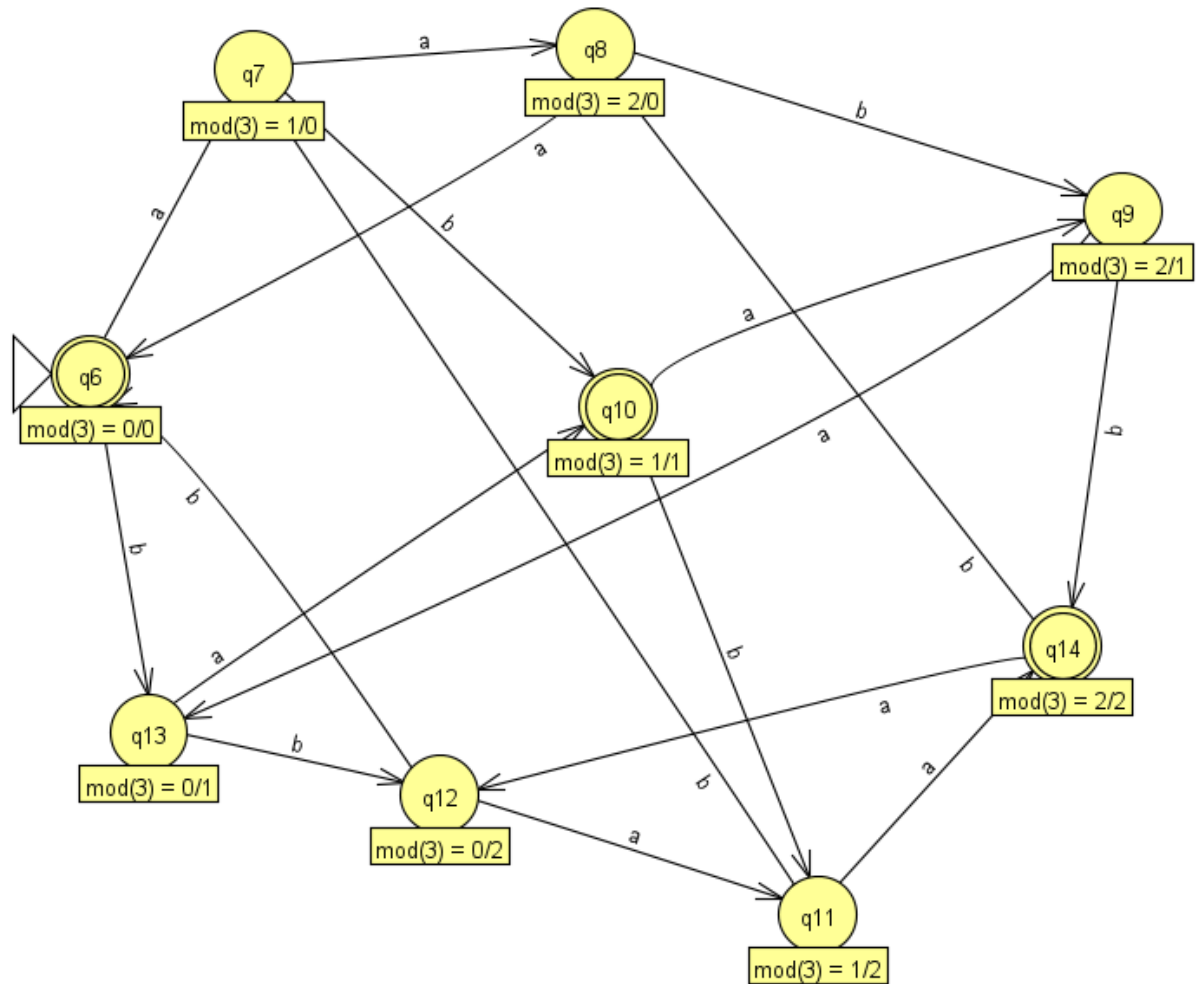


Le transizioni che non sono presenti portano allo stato pozzo. In questo caso abbiamo bisogno di 4 stati per poter memorizzare i 4 possibili stati che ci interessano e che  $y$  dovrà "ricordare". Dopo avere letto i primi caratteri l'automa continua a ciclare su 0 e 1 fino a che non capiterà un 2, solo allora procederà alla sequenza successiva.

### Esame-2017-Giugno-13-dfa

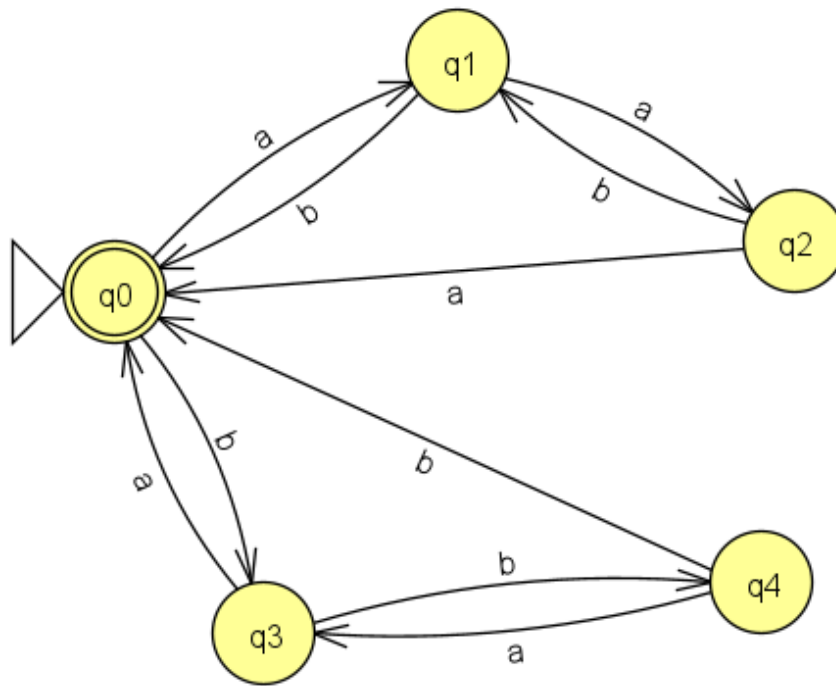
Esercizio 2. Definire (direttamente, senza fare trasformazioni) un DFA che rappresenti il linguaggio sull'alfabeto  $\{a,b\}$  delle stringhe in cui il numero  $n$  di "a" ed il numero  $m$  di "b" sono tali che:  $n$  ed  $m$  danno lo stesso resto quando divisi per 3, cioè  $n \bmod 3 = m \bmod 3$ . Ad esempio abaabb, bbabb, abab, aaa ed  $\epsilon$  appartengono al linguaggio mentre bb, a, aabbb e babaaa non vi appartengono.

Di questo esercizio sono presenti due versioni di DFA una più estesa e una più contratta.



Qui è possibile vedere la versione estesa, dove ogni possibile combinazione dei resti è stata modellata con uno stato. Le etichette degli stati ci mostrano a quanto ammonta il resto per il numero di "a" e per il numero di "b" (il primo numero dopo l'uguale si riferisce alle "a" e quello dopo "/" alle "b"). Leggendo ogni volta input "a" o "b" entriamo in una configurazione diversa.





In questo DFA vi sono 4 stati in meno e i 3 stati finali precedenti sono stati condensati in uno soltanto. Aumentando le "a" aumentano le "b" per "tornare indietro" e quindi avere un medesimo resto, viceversa aumentando le "b".

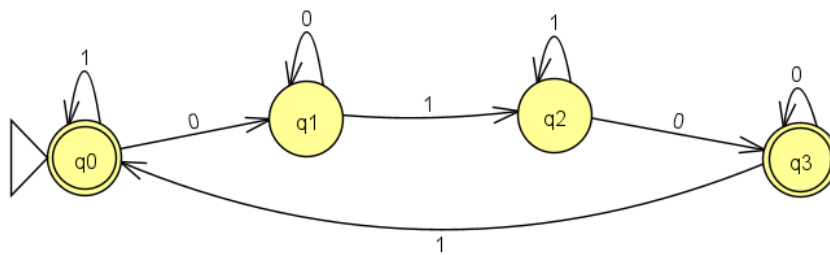
---

### **Esame-2017-Febbraio-22-dfa**

Esercizio 1. Definire (direttamente, senza fare trasformazioni) un'espressione regolare che rappresenti il linguaggio sull'alfabeto  $\{0,1\}$  delle stringhe in cui il numero di gruppi di 0 è pari. Ad esempio 011001, 1010110010, 111 ed  $\epsilon$  appartengono al linguaggio mentre 0110010, 1110 ed 1001 non vi appartengono.

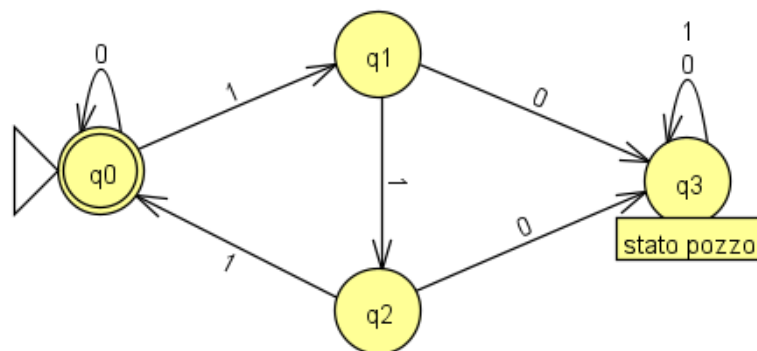
Esercizio 2. Realizzare (direttamente, senza fare trasformazioni) un DFA con 4 stati che riconosca il linguaggio dell'esercizio precedente. Dire, tramite l'algoritmo riempi-tabella visto a lezione/nel libro di testo (utilizzare una crocetta di tipo diverso ad ogni passata) se tale DFA è equivalente o meno al DFA...

*Sono riportati entrambi gli esercizi dato che le specifiche del linguaggio di quello relativo al DFA era nell'esercizio precedente*



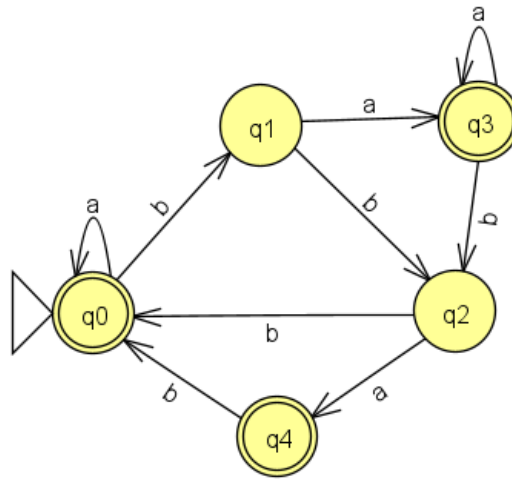
### Esame-2017-Gennaio-13

Esercizio 1. Realizzare un DFA che riconosca il linguaggio sull'alfabeto binario  $\{0,1\}$  che contiene le stringhe in cui ogni gruppo di 1 adiacenti è di lunghezza multipla di 3. Ad esempio 011100111111 appartiene al linguaggio mentre 1110010 non vi appartiene.



### Esame-2016-Giugno-14

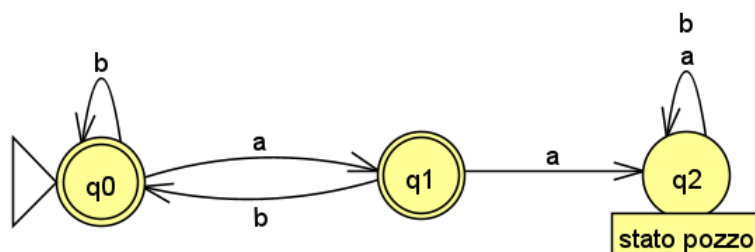
Esercizio 1. Dato l'alfabeto  $\{a,b\}$  realizzare un DFA che riconosce le stringhe che terminano con "a" o nelle quali il numero di "b" sia multiplo di 3.



Dal momento che 0 è multiplo di tutti i numero, anche la stringa vuota viene accettata. Il percorso  $q1 \rightarrow q2 \rightarrow q3$  si muove lungo il modulo per 3 del numero delle "b". Da q1 e q2 possiamo andare inoltre in uno stato di accettazione leggendo un carattere "a".

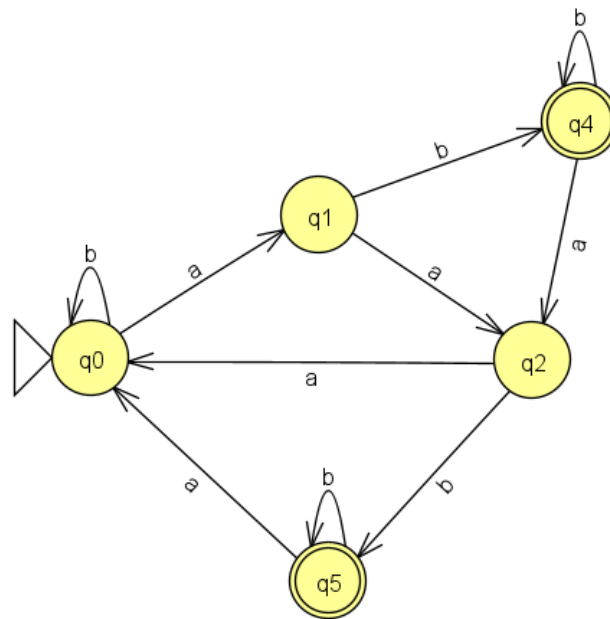
### Esame-2016-Febbraio-19

Esercizio 2. Costruire un DFA che riconosce le stringhe sull'alfabeto  $\{a,b\}$  che non contengono "aa" e trasformarlo in espressione regolare con il metodo di eliminazione degli stati visto a lezione (cioè quello definito nel libro di testo).



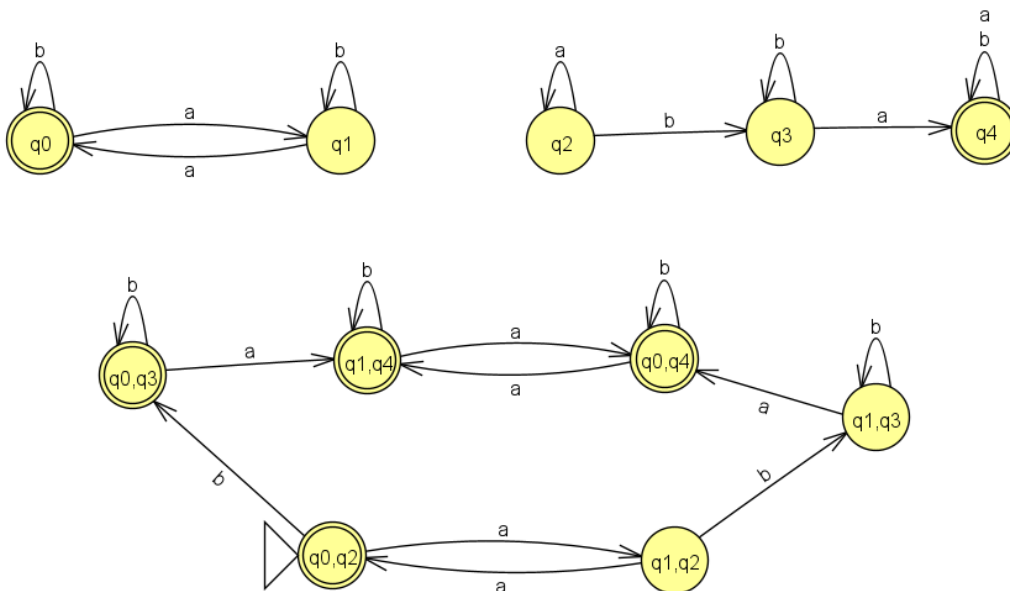
### Esame-2016-Febbraio-02

Esercizio 1: Dato l'alfabeto  $\{a,b\}$  realizzare un DFA che riconosce le stringhe che terminano con "b" e nelle quali il numero di "a" non sia multiplo di 3.



### Esame-2016-Gennaio-18

Esercizio 1. Dato l'alfabeto  $\{a,b\}$  realizzare un DFA che riconosce le stringhe che contengono "ba" oppure hanno un numero pari di "a".



Nota: gli stati "q1,q4" e "q0,q4" possono essere collasati in uno mettendoci un ulteriore autociclo per "a" oltre che per "b"

In questa risoluzione è mostrato come unire due automi date le specifiche dell'esercizio. **Gli stati iniziali degli automi parziali sono q0 e q2.** L'automata finale racchiude le coppie degli stati di quelli iniziali e ci si muove fra le coppie considerando le transizioni di entrambi per "a" e per "b". Gli stati finali sono quelli che possiedono nella coppia uno degli stati finali degli automi parziali, quindi q0 e q4.

# Esami