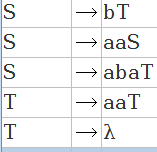
ESERCITAZIONE 10\_17

1. Definire delle grammatiche lineari destre sull'alfabeto {a,b} per i seguenti linguaggi:

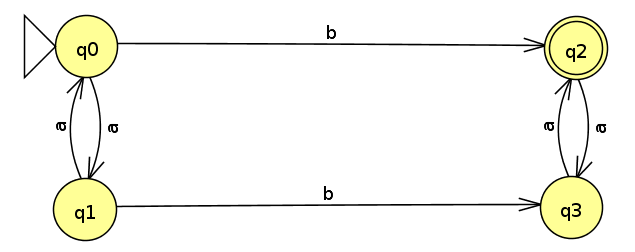
**a- linguaggio delle stringhe con un numero pari di a e con esattamente una b**



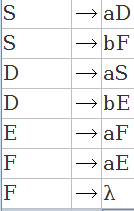
Quando arriva una b cambia stato, può leggere solo a

Quindi lo stato accettante è solo T, deve arrivare una b

Se si fa prima il DFA:



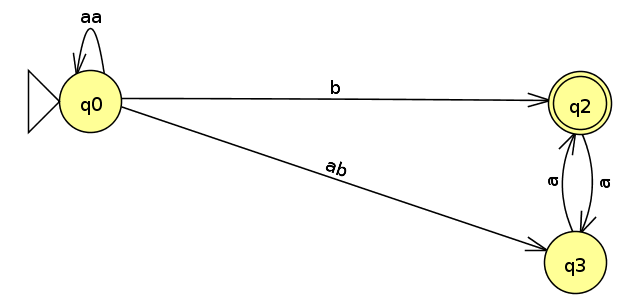
Si ottiene la seguente grammatica, prendendo come variabile uno stato direttamente:

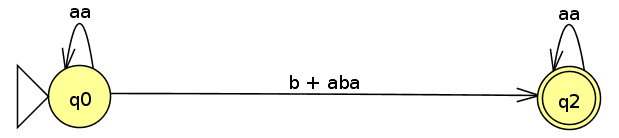


**NB**

Nello stato finale del DFA, bisogna aggiungere la produzione epsilon.

Se si vuole convertire il DFA in espressione regolare si fanno i seguenti passaggi:





L'espressione regolare si ricava dalla formula:

(R + S U\* T)\* S U\*

dove

R = aa,

S = b + aba

U = aa

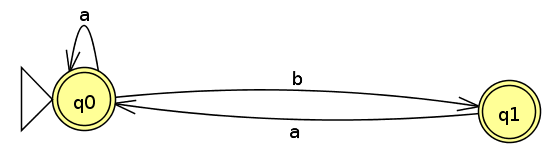
T = {} //insieme vuoto

Quindi sostituendo, si trova l'espressione regolare:

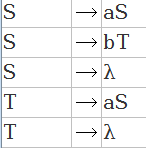
(aa)\* (b + aba) (aa\*)

**b- linguaggio delle stringhe in cui ogni b non puo' essere adiacente ad un'altra b**

Facciamo prima il DFA:



La grammatica quindi è:



In S può ricevere solo a

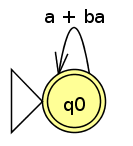
Quando arriva una b cambia stato, e deve ricevere una a, per poi tornare nello stato iniziale.

Gli stati accettanti sono sia S che T, in quanto la stringa può terminare sia con a che con b.

Convertiamo ora il DFA in espressione regolare:

Dato che ci sono 2 stati finali, di cui uno è anche iniziale, si fa l'unione di 2 casi: un caso contiene entrambi gli stati finali, un caso contiene solo lo stato finale che è anche lo stato iniziale.

Caso 1:

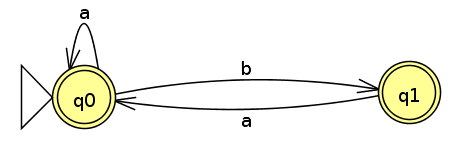


Quindi l'espressione regolare per il caso 1 è:

(a + ba)\*

perché si considera la formula R\*, dove R è “a + ba”

Caso 2:



R = a

S = b

U = {}

T = a

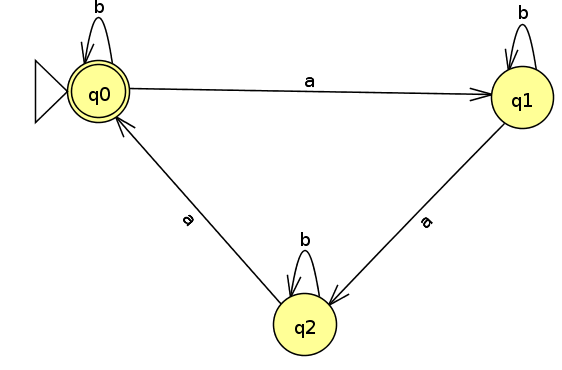
la REGEXP per il caso 2 quindi è: (a + ba)\* b

Quindi la REGEXP finale è l'unione delle 2, cioè:

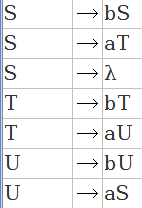
(a + ba)\* + (a + ba)\* b

c**- linguaggio delle stringhe in cui il numero delle a e' un multiplo di 3**

Il DFA corrispondente è:

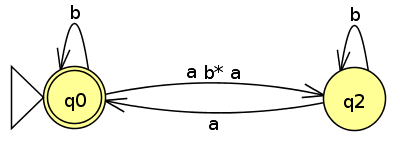


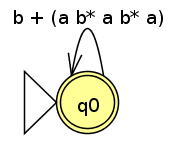
S = q0, T = q1, U = q2



Lo stato accettante è lo stato iniziale, lo stato che rappresenta il numero di a come multiplo di 3.

Conversione del DFA in REG EXP:





Quindi l'espressione regolare si ricava dalla formula R\*, quindi:

(b + (a b\* a b\* a) )\*

---------------------

Praticamente, le grammatiche lineari a destra rappresentano i linguaggi regolari, dove ogni variabile indica uno stato di un automa a stati finiti, il cambio di variabile indica un cambio di stato.

Per risolvere gli esercizi, quindi bisogna capire prima quali sono tutti gli stati necessari al linguaggio, e poi provare a scrivere la grammatica, simulandola poi con il relativo E\_NFA o DFA.

Lo stato accettante è lo stato di ogni variabile che produce epsilon o solo una sequenza di terminali

-------------------------------

**2. E' possibile trasformare una grammatica lineare destra in un automa a stati finiti? Come?**

Ogni variabile indica lo stato di un E\_NFA

Ogni produzione di ogni variabile indica i terminali in input e la variabile a destra indica lo stato di destinazione della funzione di transizione.

Per ogni stato/variabile che produce solo terminali o epsilon, lo stato di destinazione è un nuovo stato, che ha il ruolo di stato finale dell'E\_NFA.

**3. E' possibile trasformare un DFA in una grammatica lineare destra? Come?**

Ogni stato del DFA è una variabile della grammatica, che ha una produzione per ogni arco dello stato, in cui i terminali della produzione dsono tutti i terminali dell'arco, e la variabile a Destra è lo stato di destinazione.

Per ogni stato finale, la variabile corrispondente produce anche epsilon.