Automi e Linguaggi Formali

Parte 2 – Automi a Stati Finiti Non Deterministici

Università degli Studi di Padova

Davide Bresolin

Ultimo aggiornamento: 5 marzo 2024

Sommario

- **Intersezione:**

$$L \cap M = \{w : w \in L \text{ e } w \in M\}$$

- **Unione:**

$$L \cup M = \{w : w \in L \text{ oppure } w \in M\}$$

- **Complemento:**

$$\overline{L} = \{w : w \notin L\}$$

- **Concatenazione:**

$$L.M = \{uv : u \in L \text{ e } v \in M\}$$

- **Chiusura (o Star) di Kleene:**

$$L^* = \{w_1 w_2 \dots w_k : k \geq 0 \text{ e ogni } w_i \in L\}$$

Proprietà di chiusura

Se L e M sono linguaggi regolari, allora ...

- Intersezione: $L \cap M$ è un linguaggio regolare?
- Unione: $L \cup M$ è un linguaggio regolare?
- Complemento: $\overline{L}$ è un linguaggio regolare?
- Concatenazione: $L.M$ è un linguaggio regolare?
- Chiusura di Kleene: L^* è un linguaggio regolare?

Chiusura per intersezione

Theorem

Se L e M sono regolari, allora anche $L \cap M$ è un linguaggio regolare.

Chiusura per intersezione

Theorem

Se L e M sono regolari, allora anche $L \cap M$ è un linguaggio regolare.

Dimostrazione. Sia L il linguaggio di

$$A_L = (Q_L, \Sigma, \delta_L, q_L, F_L)$$

e M il linguaggio di

$$A_M = (Q_M, \Sigma, \delta_M, q_M, F_M)$$

Possiamo assumere che entrambi gli automi siano deterministici
Costruiremo un automa che simula A_L e A_M in parallelo, e accetta se e solo se sia A_L che A_M accettano.

Dimostrazione (continua).

Se A_L va dallo stato p allo stato s leggendo a, e A_M va dallo stato q allo stato t leggendo a, allora $A_{L \cap M}$ andrà dallo stato (p, q) allo stato (s, t) leggendo a.

Dimostrazione (continua).

Se A_L va dallo stato p allo stato s leggendo a, e A_M va dallo stato q allo stato t leggendo a, allora $A_{L \cap M}$ andrà dallo stato (p, q) allo stato (s, t) leggendo a.

$A_{L \cap M}$ accetta una parola solo quando sia A_L che A_M accettano

$$\Leftrightarrow$$

$A_{L \cap M}$ accetta solo quando (p, q) è una coppia di stati finali

Dimostrazione (continua).

Se A_L va dallo stato p allo stato s leggendo a, e A_M va dallo stato q allo stato t leggendo a, allora $A_{L \cap M}$ andrà dallo stato (p, q) allo stato (s, t) leggendo a.

$A_{L \cap M}$ accetta una parola solo quando sia A_L che A_M accettano

$$\Leftrightarrow$$

$A_{L \cap M}$ accetta solo quando (p, q) è una coppia di stati finali

Formalmente

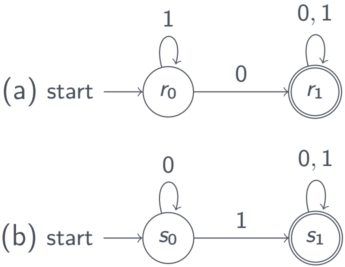
$$A_{L \cap M} = (Q_L \times Q_M, \Sigma, \delta_{L \cap M}, (q_L, q_M), F_L \times F_M),$$

dove

$$\delta_{L \cap M}((p, q), a) = (\delta_L(p, a), \delta_M(q, a))$$

Esempio

Costruiamo l'automa che rappresenta l'intersezione di (a) e (b)



E le altre operazioni ?

Dati i DFA per L e M:

E le altre operazioni ?

Dati i DFA per L e M:

- possiamo costruire un DFA per $L \cup M$? Se si, come?

E le altre operazioni ?

Dati i DFA per L e M:

- possiamo costruire un DFA per $L \cup M$? Se si, come?
- possiamo costruire un DFA per $\overline{L}$? Se si, come?

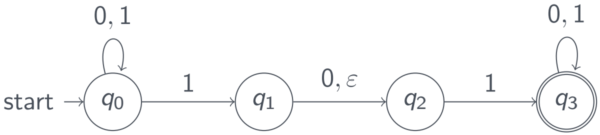
E le altre operazioni ?

Dati i DFA per L e M:

- possiamo costruire un DFA per $L \cup M$? Se si, come?
- possiamo costruire un DFA per $\overline{L}$? Se si, come?
- possiamo costruire un DFA per $L.M$? Se si, come?
- possiamo costruire un DFA per L^*? Se si, come?

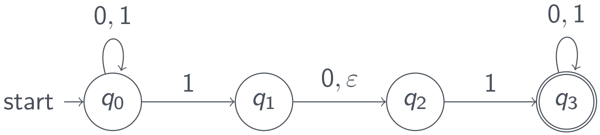
Sommario

- Cosa fa questo automa?



Automi a stati finiti non deterministici (NFA)

- Cosa fa questo automa?



- È un esempio di automa a stati finiti non deterministico:
  - ci possono essere più transizioni con lo stesso simbolo
  - o simboli senza transizioni uscenti
  - ed ε-transizioni che non consumano simboli
- Data una parola, esistono più percorsi possibili
- Si accetta se esiste almeno un percorso accettante

Un Automa a Stati Finiti Non Deterministico (NFA) è una quintupla

$$A = (Q, \Sigma, \delta, q_0, F)$$

- Q è un insieme finito di stati
- Σ è un alfabeto finito che non contiene ε. Definiamo $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$.
- $\delta : Q \times \Sigma_\varepsilon \mapsto 2^Q$ è una funzione di transizione che prende in input (q, a) e restituisce un sottoinsieme di Q
- $q_0 \in Q$ è lo stato iniziale
- $F \subseteq Q$ è un insieme di stati finali

Computazione di un NFA

- Data una parola $w = w_1 w_2 \dots w_m$, dove $w_i \in \Sigma_\varepsilon$
- una computazione di un NFA A con input w è una sequenza di stati $r_0 r_1 \dots r_m$ che rispetta due condizioni:
  1. $r_0 = q_0$ (inizia dallo stato iniziale)
  2. $r_{i+1} \in \delta(r_i, w_{i+1})$ per ogni $i = 0, \dots, m-1$ (rispetta la funzione di transizione)
- Diciamo che una computazione accetta la parola w se:
  3. la computazione legge tutti i simboli della stringa)
  4. la computazione termina in uno stato finale)
- A causa del nondeterminismo, ci può essere più di una computazione per ogni parola!

Linguaggio accettato da un NFA

- Un NFA A accetta la parola w se esiste una computazione che accetta w
- Un NFA A rifiuta la parola w se tutte le computazioni la rifiutano
- Formalmente, il linguaggio accettato da A è

$$L(A) = \{w \in \Sigma^* \mid A \text{ accetta } w\}$$

Esercizi

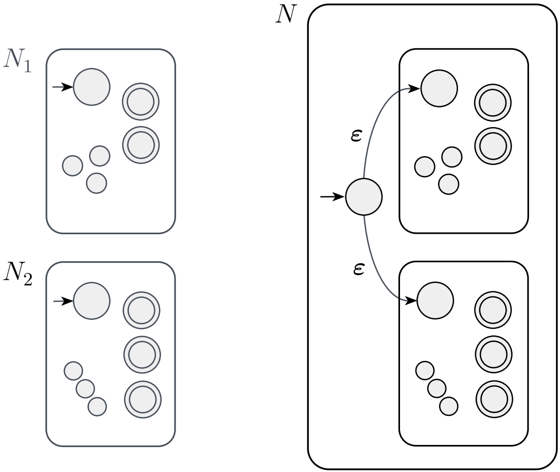
Definire degli automi a stati finiti non deterministici che accettino i seguenti linguaggi:

- L'insieme delle parole sull'alfabeto $\{0, 1, \ldots, 9\}$ tali che la cifra finale sia comparsa in precedenza
- L'insieme delle parole sull'alfabeto $\{0, 1, \ldots, 9\}$ tali che la cifra finale *non* sia comparsa in precedenza
- L'insieme delle parole di 0 e 1 tali che esistono due 0 separati da un numero di posizioni multiplo di 4 (0 è un multiplo di 4)

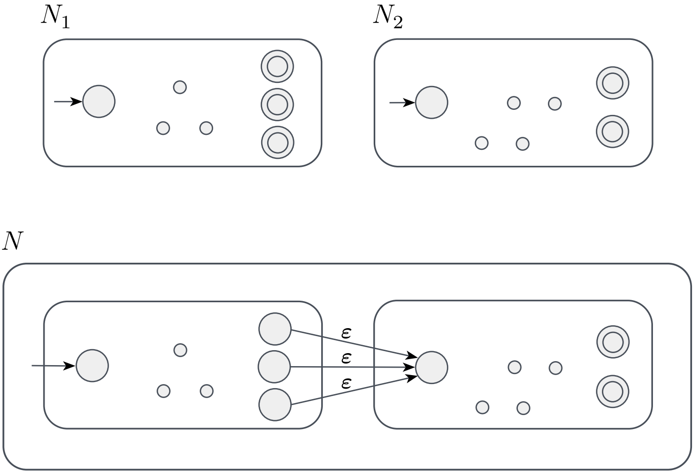
Consideriamo l'alfabeto $\Sigma = \{a, b, c, d\}$ e costruiamo un automa non deterministico che riconosce il linguaggio di tutte le parole tali che uno dei simboli dell'alfabeto non compare mai:

- tutte le parole che non contengono a
- $+$ tutte le parole che non contengono b
- $+$ tutte le parole che non contengono c
- $+$ tutte le parole che non contengono d

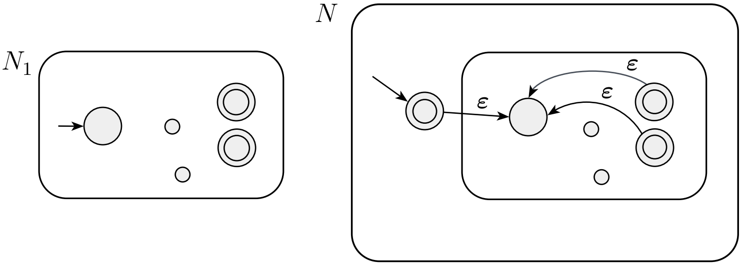
Unione



Concatenazione



Star di Kleene

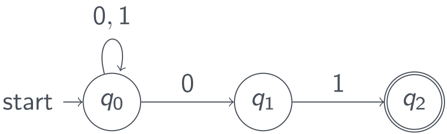


Sommario

Equivalenza di DFA e NFA

- Sorprendentemente, NFA e DFA sono in grado di riconoscere gli stessi linugaggi
- Per ogni NFA N c'è un DFA D tale che $L(D) = L(N)$, e viceversa

Esempio:



La costruzione a sottoinsiemi (1)

Università degli Studi di Padova

- L'equivalenza di dimostra mediante una costruzione a sottoinsiemi:

La costruzione a sottoinsiemi (1)

- L'equivalenza di dimostra mediante una costruzione a sottoinsiemi:
  Dato un NFA
  $$N = (Q_N, \Sigma, q_0, \delta_N, F_N)$$
  costruiremo un DFA
  $$D = (Q_D, \Sigma, S_0, \delta_D, F_D)$$
  tale che
  $$L(D) = L(N)$$

La costruzione a sottoinsiemi

Università degli Studi di Padova

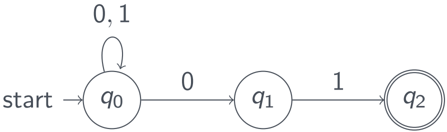
Iniziamo dal caso più semplice in cui N non ha ε-transizioni

- $Q_D = \{S : S \subseteq Q_N\}$
  Ogni stato del DFA corrisponde ad un insieme di stati dell'NFA
- $S_0 = \{q_0\}$
  Lo stato iniziale del DFA è l'insieme che contiene solo q_0
- $F_D = \{S \subseteq Q_N : S \cap F_N \neq \emptyset\}$
  Uno stato del DFA è finale se c'è almeno uno stato finale corrispondente nell'NFA
- Per ogni $S \subseteq Q_N$ e per ogni $a \in \Sigma$

$$\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a)$$

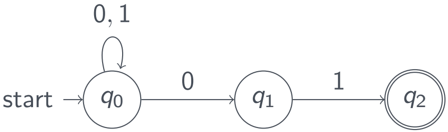
La funzione di transizione "percorre tutte le possibili strade"

Esempio di costruzione a sottoinsiemi



Costruiamo δ_D per l'NFA qui sopra:

Esempio di costruzione a sottoinsiemi

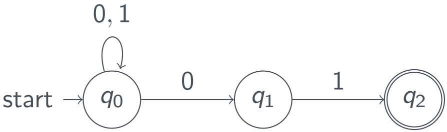


Costruiamo δ_D per l'NFA qui sopra:

	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_2\}$
$*\{q_2\}$	$\emptyset$	$\emptyset$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$*\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
$*\{q_1, q_2\}$	$\emptyset$	$\{q_2\}$
$*\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

Esempio di costruzione a sottoinsiemi

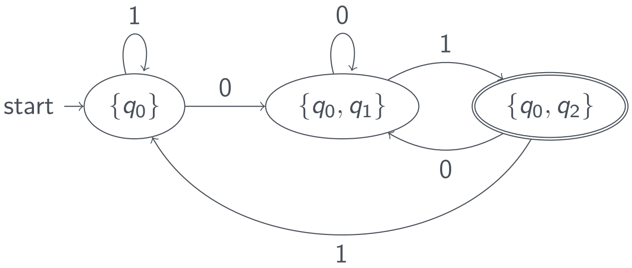
Università degli Studi di Padova



Costruiamo δ_D per l'NFA qui sopra:

	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_2\}$
$*\{q_2\}$	$\emptyset$	$\emptyset$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$*\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
$*\{q_1, q_2\}$	$\emptyset$	$\{q_2\}$
$*\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

La tabella di transizione per D ci permette di ottenere il diagramma di transizione



Nota: $|Q_D| = 2^{|Q_N|}$, anche se alcuni degli stati in Q_D possono essere "inutili", cioè non raggiungibili dallo stato iniziale. In questo caso solo tre stati sono raggiungibili, e gli altri possono essere omessi.

Epsilon chiusura: definizione

Per poter gestire le ε-transizioni introduciamo ε-chiusura degli stati:

- tutti gli stati raggiungibili da q con una sequenza $\varepsilon\varepsilon \dots \varepsilon$

La definizione di $\mathrm{ECLOSE}(q)$ è per induzione:

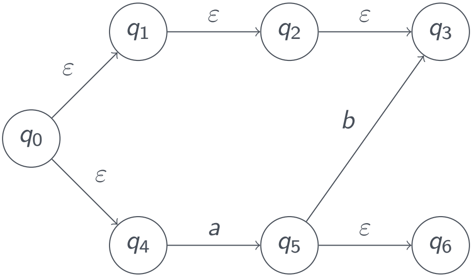
Caso base:

$$q \in \mathrm{ECLOSE}(q)$$

Caso induttivo:

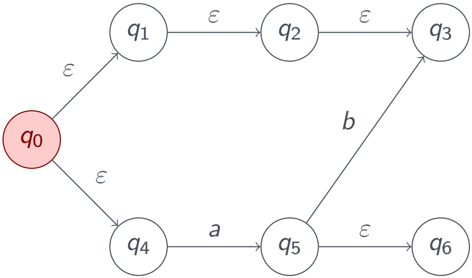
$$\text{se } p \in \mathrm{ECLOSE}(q) \text{ e } r \in \delta(p, \varepsilon) \text{ allora } r \in \mathrm{ECLOSE}(q)$$

Epsilon chiusura: esempio



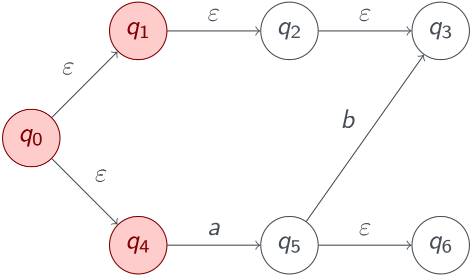
$$\mathrm{ECLOSE}(q_0) = \{$$

Epsilon chiusura: esempio



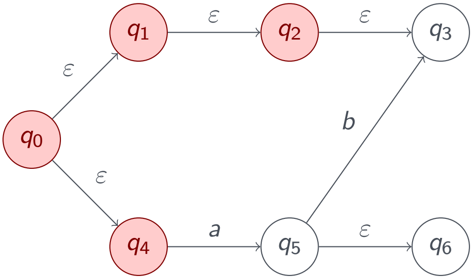
$$\mathrm{ECLOSE}(q_0) = \{q_0$$

Epsilon chiusura: esempio



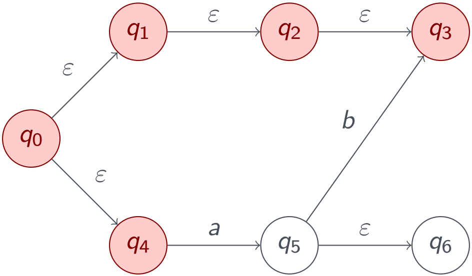
$$\mathrm{ECLOSE}(q_0) = \{q_0, q_1, q_4\}$$

Epsilon chiusura: esempio



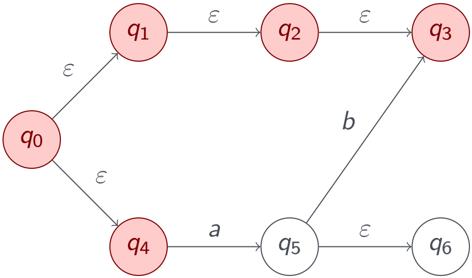
$$\mathrm{ECLOSE}(q_0) = \{q_0, q_1, q_4, q_2$$

Epsilon chiusura: esempio



$$\mathrm{ECLOSE}(q_0) = \{q_0, q_1, q_4, q_2, q_3$$

Epsilon chiusura: esempio



$$\mathrm{ECLOSE}(q_0) = \{q_0, q_1, q_4, q_2, q_3\}$$

La costruzione a sottoinsiemi completa

- $Q_D = \{S \subseteq Q_N : S = \mathrm{ECLOSE}(S)\}$
  Ogni stato è un insieme di stati chiuso per ε-transizioni
- $S_0 = \mathrm{ECLOSE}(q_0)$
  Lo stato iniziale è la ε-chiusura dello stato iniziale di E
- $F_D = \{S \in Q_D : S \cap F_N \neq \emptyset\}$
  Uno stato del DFA è finale se c'è almeno uno stato finale di N
- Per ogni $S \in Q_D$ e per ogni $a \in \Sigma$:

$$\delta_D(S, a) = \mathrm{ECLOSE}\Big(\bigcup_{p \in S} \delta_N(p, a)\Big)$$

La funzione di transizione "percorre tutte le possibili strade" (comprese quelle con ε-transizioni)

La costruzione a sottoinsiemi completa

- $Q_D = \{S \subseteq Q_N : S = \mathrm{ECLOSE}(S)\}$
  Ogni stato è un insieme di stati chiuso per ε-transizioni
- $S_0 = \mathrm{ECLOSE}(q_0)$
  Lo stato iniziale è la ε-chiusura dello stato iniziale di E
- $F_D = \{S \in Q_D : S \cap F_N \neq \emptyset\}$
  Uno stato del DFA è finale se c'è almeno uno stato finale di N
- Per ogni $S \in Q_D$ e per ogni $a \in \Sigma$:

$$\delta_D(S, a) = \mathrm{ECLOSE}\Big(\bigcup_{p \in S} \delta_N(p, a)\Big)$$

  La funzione di transizione "percorre tutte le possibili strade" (comprese quelle con ε-transizioni)

Nota: anche in questo caso $|Q_D| = 2^{|Q_N|}$

Theorem

Un linguaggio L è accettato da un DFA se e solo se è accettato da un NFA.

Dimostrazione:

- La parte "se" è data dalla costruzione per sottoinsiemi
- La parte "solo se" si dimostra osservando che ogni DFA può essere trasformato in un NFA modificando δ_D in δ_N con la seguente regola:

  Se $\delta_D(q, a) = p$ allora $\delta_N(q, a) = \{p\}$

1. Determinare il DFA equivalente all'NFA con la seguente tabella di transizione:

	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_1\}$	$\{q_0, q_2\}$
$*q_2$	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$

2. Qual è il linguaggio accettato dall'automa?

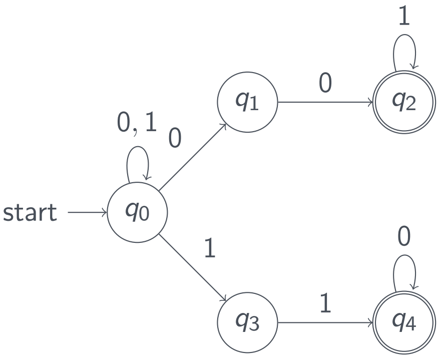
1. Determinare il DFA equivalente all'NFA con la seguente tabella di transizione:

	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_1\}$	$\{q_0, q_2\}$
$*q_2$	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$

2. Qual è il linguaggio accettato dall'automa?

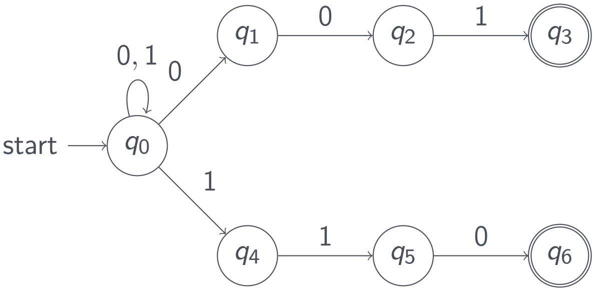
Esercizio 3

Trasformare il seguente NFA in DFA



Esercizio 4

Dato il seguente NFA



1. determinare il linguaggio riconosciuto dall'automa
2. costruire un DFA equivalente

Esercizio 5

Università degli Studi di Padova

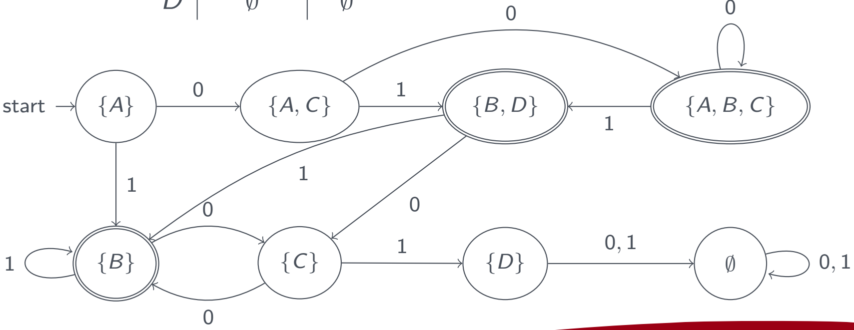
Convertire il seguente NFA in DFA:

	0	1
$\rightarrow A$	$\{A, C\}$	$\{B\}$
$*B$	$\{C\}$	$\{B\}$
C	$\{B\}$	$\{D\}$
D	$\emptyset$	$\emptyset$

Università degli Studi di Padova

Convertire il seguente NFA in DFA:

	0	1
$\rightarrow A$	$\{A, C\}$	$\{B\}$
$*B$	$\{C\}$	$\{B\}$
C	$\{B\}$	$\{D\}$
D	$\emptyset$	$\emptyset$



Esercizio 6

1. Costruiamo un NFA che riconosce le parole costituite da
   - zero o più a
   - seguite da zero o più b
   - seguite da zero o più c
2. Calcolare ECLOSE di ogni stato dell'automa
3. Convertire l'NFA in DFA

Esercizio

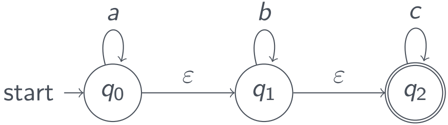
1. Costruiamo un NFA che riconosce le parole costituite da zero o più a, seguite da zero o più b, seguite da zero o più c

2. Calcolare la ε-chiusura di ogni stato

3. Convertire l'NFA in DFA

Esercizio

1. Costruiamo un NFA che riconosce le parole costituite da zero o più a, seguite da zero o più b, seguite da zero o più c

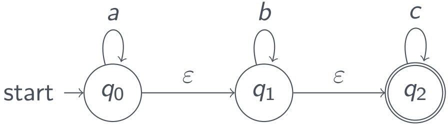


2. Calcolare la ε-chiusura di ogni stato

3. Convertire l'NFA in DFA

Esercizio

1. Costruiamo un NFA che riconosce le parole costituite da zero o più a, seguite da zero o più b, seguite da zero o più c



2. Calcolare la ε-chiusura di ogni stato

$$\mathrm{ECLOSE}(q_0) = \{q_0, q_1, q_2\}$$
$$\mathrm{ECLOSE}(q_1) = \{q_1, q_2\}$$
$$\mathrm{ECLOSE}(q_2) = \{q_2\}$$

3. Convertire l'NFA in DFA

1. Costruiamo un NFA che riconosce le parole costituite da zero o più a, seguite da zero o più b, seguite da zero o più c
2. Calcolare la ε-chiusura di ogni stato
3. Convertire l'NFA in DFA