Devo definire M t.c. $L(M) = L_1 \cup L_2$.

Problema: dato x candidato non posso prima provare e vedere se $M_1(x)$ accetta.

Idea: M deve eseguire M_1, M_2 in parallelo e accettare sse uno dei due accetta.

DIM. Siano $M_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1)$

$M_2 = (Q_2, \Sigma, \delta_2, q_0^2, F_2)$

t.c. $L(M_1) = L_1$ e $L(M_2) = L_2$.

Devo costruire $M = (Q, \Sigma, \delta, q_0, F)$ t.c. $L(M) = L_1 \cup L_2$. Avremo:

- $Q = \{(r_1, r_2) : r_1 \in Q_1, r_2 \in Q_2\}$

$$= Q_1 \times Q_2.$$

- $\delta : Q \times \Sigma \to Q$

$$\delta((r_1, r_2), a) =$$

$$= (\delta_1(r_1, a), \delta_2(r_2, a))$$

$$- F = \{ (r_1, r_2) : r_1 \in F_1 \vee r_2 \in F_2 \}$$

$$= (F_1 \times Q_2) \cup (Q_1 \times F_2)$$

(Se fosse stato $L = L_1 \cap L_2$?

Avremmo dovuto usare $F_1 \times F_2 = F$).

Correttezza : Usare induzione per dimostrare la seguente proprietà

$$\delta^*(q_0, x) = \delta^*((q_0^1, q_0^2), x)$$

$$\forall x \in \Sigma^*$$

$$= \left(\delta_1^*(q_0^1, x), \delta_2^*(q_0^2, x)\right)$$

Esercizio: Provatelo per induzione.

La correttezza segue immediatamente:

$$\forall x \in \Sigma^* \quad x \in L(M) \iff x \in L_1 \cup L_2$$

$\Rightarrow$ Se $x \in L(M)$, allora

$$\delta^*(q_0, x) \in F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$$

$$S = (p, q)$$

$$p = \delta_1^*(q_0^1, x) \text{ e } q = \delta_2^*(q_0^2, x)$$

$\Rightarrow$ $p \in F_1$ oppure $q \in F_2$

$\Rightarrow$ $x \in L_1$ oppure $x \in L_2$ ✓

$\Leftarrow$ Se $x \in L_1$, $\delta^*(q_0^1, x) \in F_1$

$$\delta^*(q_0, x) = (\delta_1^*(q_0^1, x), \delta_2^*(q_0^2, x))$$

$\in F_1 \times Q_2 \subseteq F$

$\Longrightarrow M$ accetta x

Stessa cosa se $x \in L_2$. ■

Esercizio: Dimostrare che REG è chiuso per COMPLEMENTO.

Vogliamo che trattare la CONCATENAZIONE.

Dati $L_1, L_2 \in REG$ vorremmo dimostrare che $L_1 \circ L_2 \in REG$.

Dato x, devo "capire" come spezzare x in $x_1 \circ x_2 = x_1 x_2$ t.c. $x_1 \in L_1$, $x_2 \in L_2$. Sembra complicato. Per risolvere questo problema introduciamo:

NON DETERMINISMO

E' un po' una "magia". Finora la computazione è deterministica, ovvero se $D \in DFA$ e D legge $e \in \Sigma$ nello stato $q \in Q$, allora D può andare in un unico altro stato $p \in Q$.

Nel caso non deterministico:

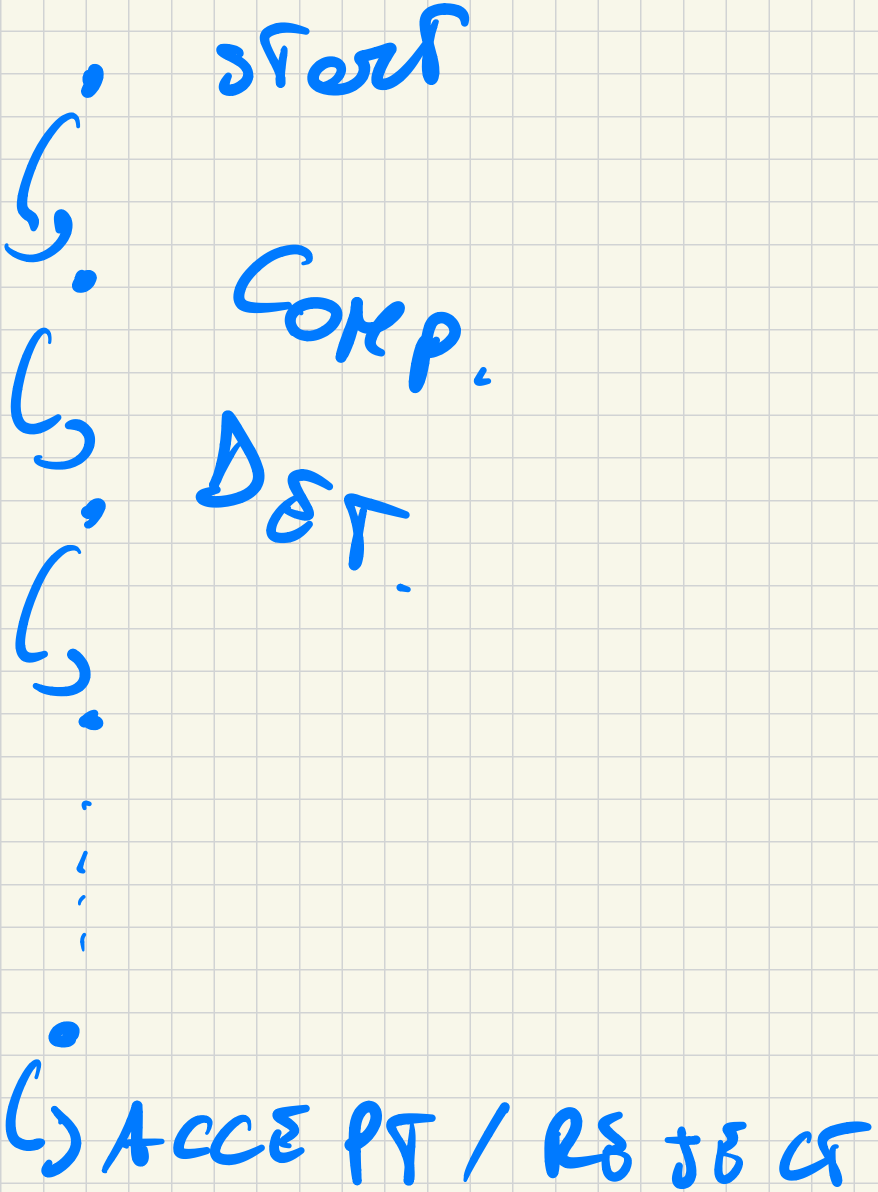
- Quando l'automa è in q e legge a può andare in diversi stati $p_1, p_2, \dots$

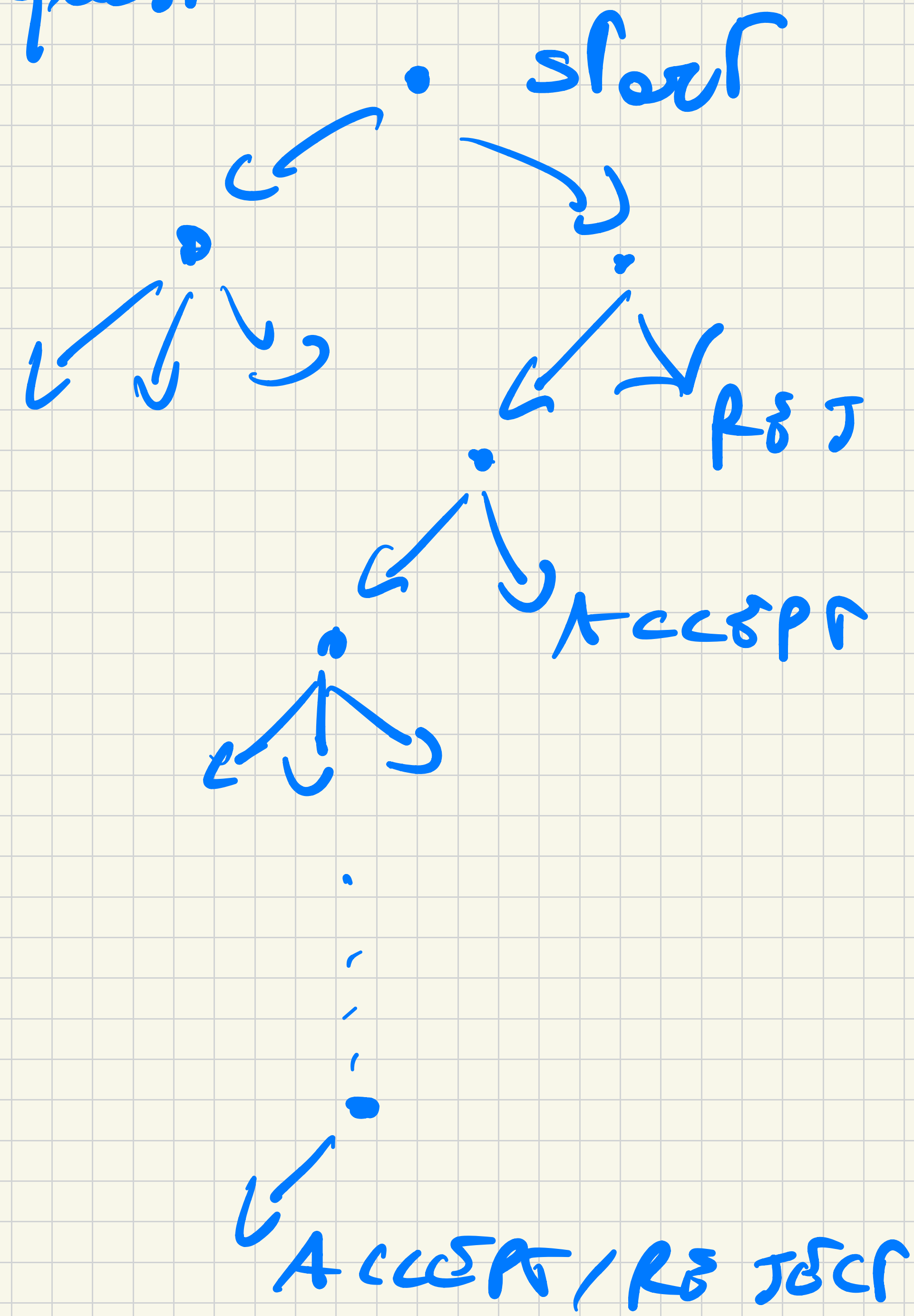
- Inoltre sono ammessi "ε-archi".

Consentono di aprire rami col comportamento parallelo senza leggere nulla.

- Accettazione: Accetto sse esiste ALMENO un ramo che accetta.

In pratica abbiamo questo:





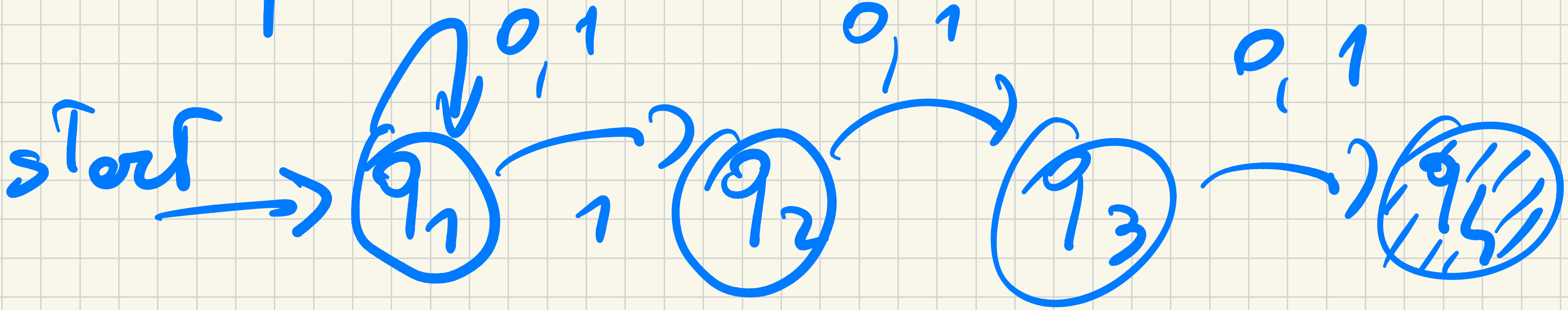
Def (NFA). Un NFA è $(Q, \Sigma, \delta, q_0, F)$ dove Q, Σ, q_0, F sono come nel DFA mentre invece:

$$\delta : Q \times \Sigma_\varepsilon \rightarrow \mathcal{P}(Q)$$

$$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$$

$\mathcal{P}(Q)$ l'insieme di tutti i possibili sottoinsiemi di Q

Esempio:



$L = \{ x \in \{0,1\}^* :$ x ha "1" in terz'ultima posizione $\}$

$x = 100$ ✓

$x = 1000$ ✗

$x = 10110 \in L$

(Per caso: 10010.)

- $q_1, 10110$
  - $q_1, 0110$
    - $q_1, 110$
      - $q_1, 10$
      - $q_2, 10$
  - $q_2, 0110$
    - $q_3, 110$
      - $q_4, 10$
        - × REJ

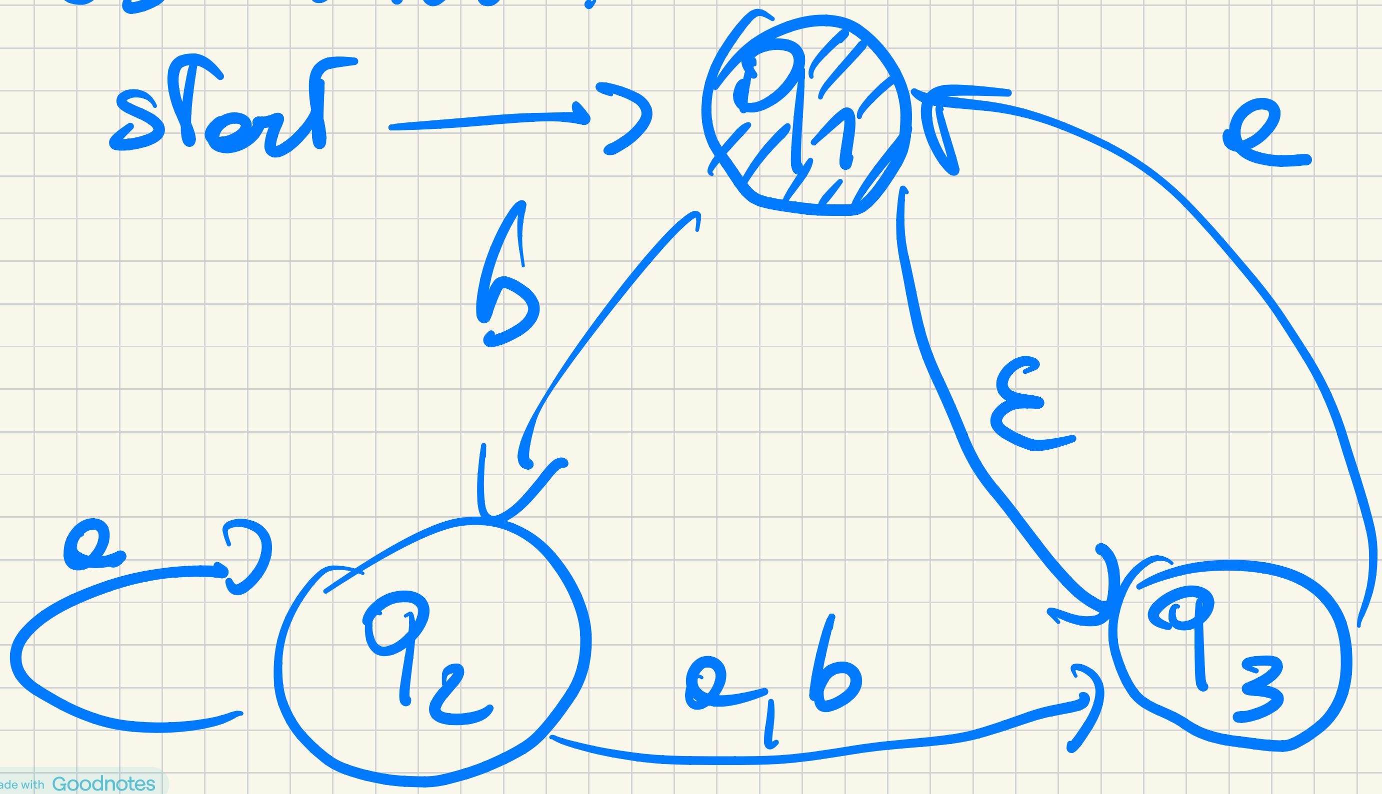
q_1, a q_2, a q_3, a

$|$ $|$ $|$

q_1, ε q_3, ε q_4, ε ✓ ACC

x REJ x REJ

Esercizio:



Controllore computazione per:

$w = \varepsilon$

$w = a$

$w = bb$

$w = babba$.

Possiamo estendere il concetto di CONFIGURAZIONE: per NFA N sarà una coppia $(q, x) \in Q \times \Sigma_\varepsilon^*$. Avremo:

$$(p, ex) \vdash_N (q, x)$$

$x \in \Sigma_\varepsilon^*$
$e \in \Sigma_\varepsilon$
$p, q \in Q$

sse $q \in \delta(p, e)$

$$\delta(p, e) \in \mathcal{P}(Q)$$

Quando è che N accetta $w \in \Sigma_\varepsilon^*$? Se e solo se $\exists\, q \in F$ t.c.

$$(q_0, w) \vdash_N^* (q, \varepsilon)$$

dove $\vdash_N^*$ è la relazione estesa.

Adesso voglio confrontare due classi:

$$\mathcal{L}(DFA) = REG$$

$$\mathcal{L}(NFA) = \{ L : \exists \text{ NFA } N \text{ t.c. } L(N) = L \}$$

TEO. $REG = \mathcal{L}(NFA)$.

DIM. Devo mostrare $\mathcal{L}(NFA) \subseteq \mathcal{L}(DFA)$ e $\mathcal{L}(DFA) \subseteq \mathcal{L}(NFA)$.

OVVIO: Se $L \in \mathcal{L}(DFA)$ $\exists$ DFA D t.c. $L(D) = L$, ma allora $N = D$ è NFA t.c. $L(N) = L$.

L'altra direzione: Se $L \in \mathcal{L}(NFA)$ $\exists$ NFA $N = (Q_N, \Sigma, \delta_N, q_0^N, F_N)$ t.c. $L(N) = L$.

Devo costruire DFA $D = (Q_D, \Sigma, \delta_D, q_0^D, F_D)$

t.c. $L(D) = L$.

(Stessa idea della prova di chiusura per unione.)

Caso semplice: N_0 ε-ordine. Allora avremo:

- $Q_D = \mathcal{P}(Q_N)$

- $q_0^D = \{q_0^N\}$

R contiene uno stato finale di Q_N.

- $F_D = \{R \in Q_D : R \cap F_N \neq \emptyset\}$

- Sia $R \in Q_D$, $e \in \Sigma$:

$$\delta_D(R, e) = \bigcup_{r \in R} \delta_N(r, e)$$

$$\left(= \{ q \in Q_N : q \in \delta_N(r, e) \text{ per qualche } r \in R \} \right)$$

Di fatto D sta emulando deterministicamente tutti i rami di computazione non-det. di N.

Caso generale: ci sono ε-archi. Devo tenere traccia degli stati raggiungibili usando ε-archi.

Sia $R \in Q_D$. Definisco:

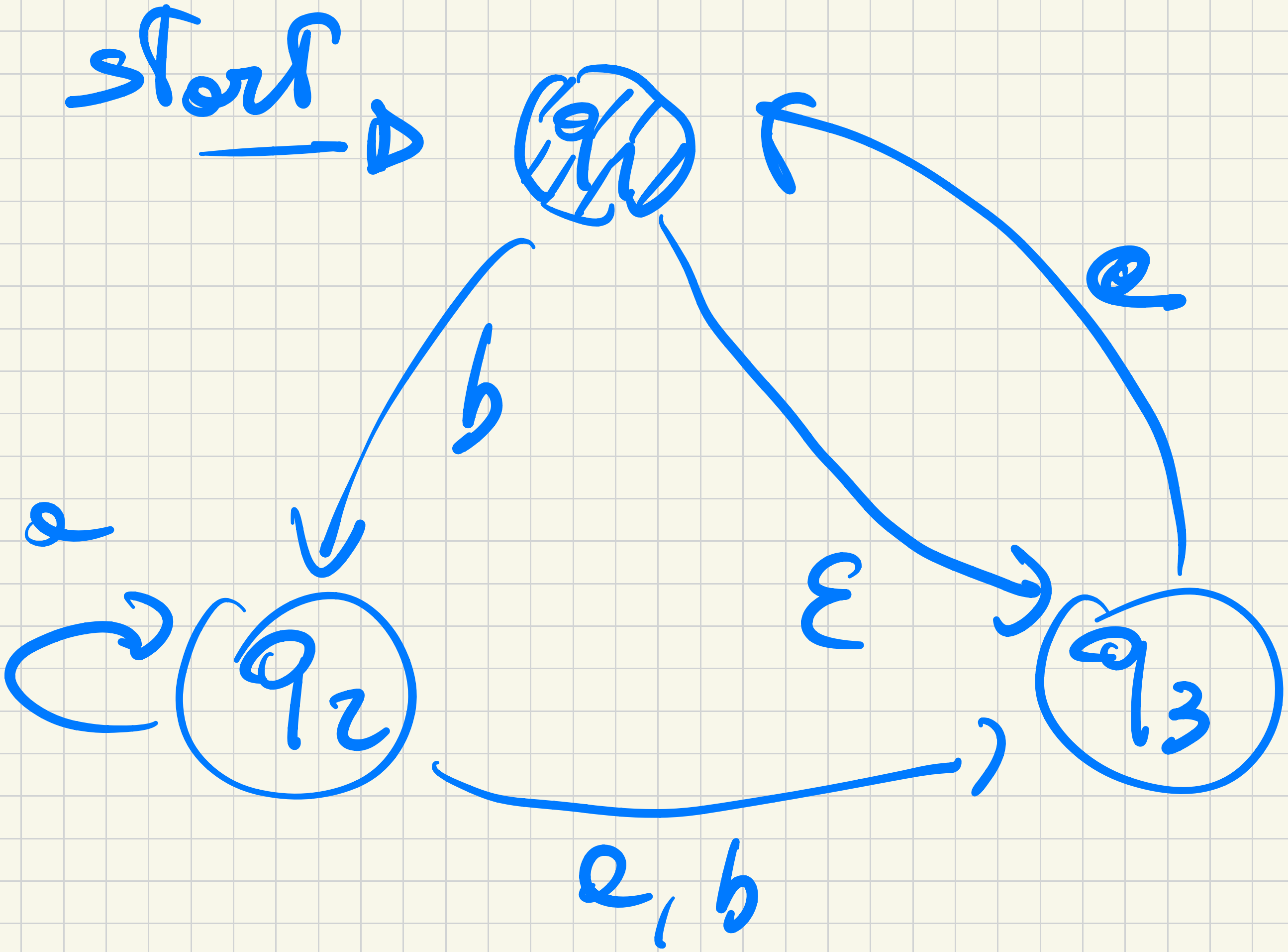
$$E(R) = \{ q \in Q_N : q \text{ può essere raggiunto da stato in } R \text{ attraverso } \geq 0 \ \varepsilon\text{-archi} \}.$$

A questo punto:

$$\delta_D(R, a) = \bigcup_{r \in R} E(\delta_N(r, a))$$

Correttezza: come sopra $\blacksquare$

Esempio: Dato il seguente NFA



Definire DFA D t.c.

$L(D) = L(N)$.