Automi e Linguaggi Formali

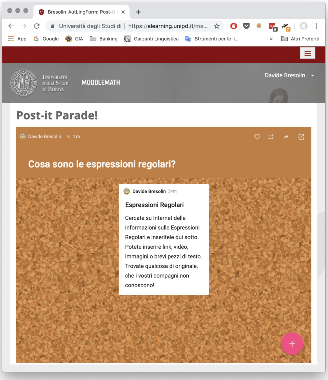
2. Automi con ε-transizioni ed Espressioni Regolari

Davide Bresolin
a.a. 2018/19

Università degli Studi di Padova

Post-it Parade!

- Accedete al Moodle del corso
- Andate alla settimana 4 Marzo – 10 Marzo
- Cliccate sul link Post-it Parade!
- Inserite un post sulla bacheca, seguendo le indicazioni.



Automi a Stati Finiti con epsilon-transizioni

NFA con epsilon-transizioni

Università degli Studi di Padova

Esercizio: costruiamo un NFA che accetta numeri decimali:

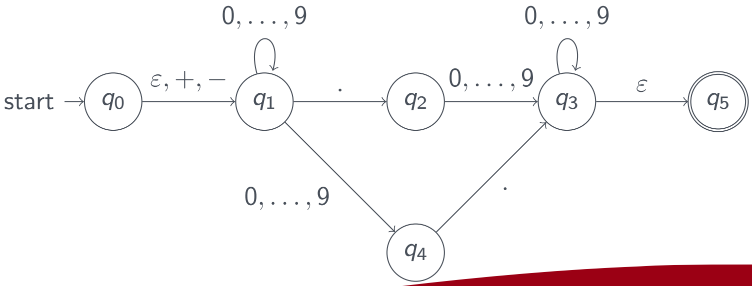
1. Un segno $+$ o $-$, opzionale
2. Una stringa di cifre decimali $\{0, \dots, 9\}$
3. un punto decimale .
4. un'altra stringa di cifre decimali

Una delle stringhe (2) e (4) può essere vuota, ma non entrambe

Esercizio: costruiamo un NFA che accetta numeri decimali:

1. Un segno $+$ o $-$, opzionale
2. Una stringa di cifre decimali $\{0, \ldots, 9\}$
3. un punto decimale .
4. un'altra stringa di cifre decimali

Una delle stringhe (2) e (4) può essere vuota, ma non entrambe



ε-NFA: definizione

Università degli Studi di Padova

Un Automa a Stati Finiti Non Deterministico con ε-transizioni (ε-NFA) è una quintupla

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

dove:

- Q, Σ, q_0, F sono definiti come al solito
- δ è una funzione di transizione che prende in input:
  - uno stato in Q
  - un simbolo nell'alfabeto $\Sigma \cup \{\varepsilon\}$

  e restituisce un sottoinsieme di Q

Esempio di ε-NFA

L'automa che riconosce le cifre decimali è definito come

$$A = (\{q_0, q_1, \dots, q_5\}, \{+, -, ., 0, \dots, 9\}, \delta, q_0, \{q_5\})$$

dove δ è definita dalla tabella di transizione

Esempio di ε-NFA

L'automa che riconosce le cifre decimali è definito come

$$A = (\{q_0, q_1, \ldots, q_5\}, \{+, -, ., 0, \ldots, 9\}, \delta, q_0, \{q_5\})$$

dove δ è definita dalla tabella di transizione

	ε	$+, -$	$.$	$0, \ldots, 9$
$\rightarrow q_0$	$\{q_1\}$	$\{q_1\}$	$\emptyset$	$\emptyset$
q_1	$\emptyset$	$\emptyset$	$\{q_2\}$	$\{q_1, q_4\}$
q_2	$\emptyset$	$\emptyset$	$\emptyset$	$\{q_3\}$
q_3	$\{q_5\}$	$\emptyset$	$\emptyset$	$\{q_3\}$
q_4	$\emptyset$	$\emptyset$	$\{q_3\}$	$\emptyset$
$*q_5$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Epsilon chiusura: definizione

L'eliminazione delle ε-transizioni procede per 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:

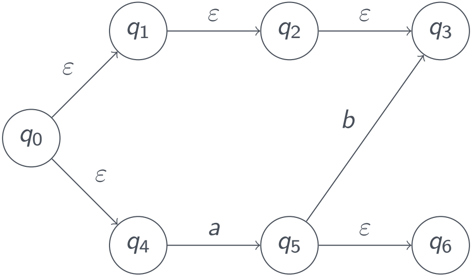
Caso base:

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

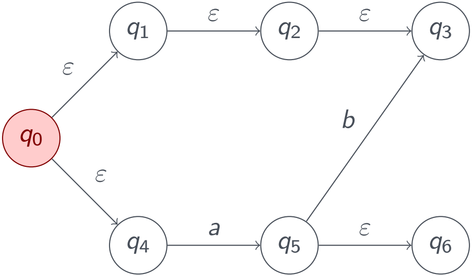
Caso induttivo:

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

Epsilon chiusura: esempio

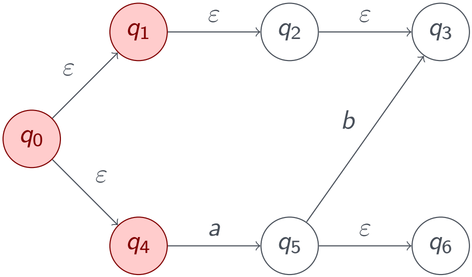


$$\mathrm{ECLOSE}(q_0) = \{$$



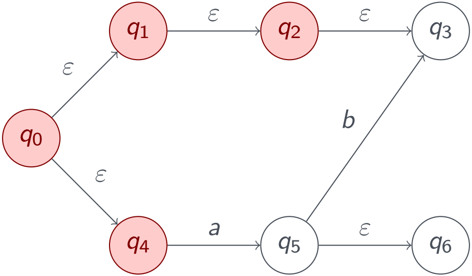
$$\mathrm{ECLOSE}(q_0) = \{q_0$$

Epsilon chiusura: esempio



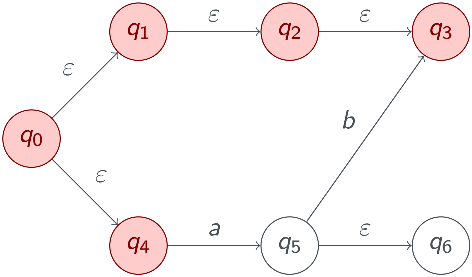
$$\mathrm{ECLOSE}(q_0) = \{q_0, q_1, q_4$$

Epsilon chiusura: esempio



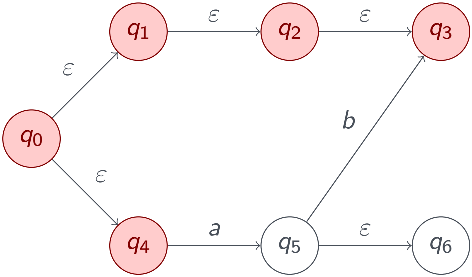
$$\mathrm{ECLOSE}(q_0) = \{q_0, q_1, q_4, q_2$$

Epsilon chiusura: esempio



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

Epsilon chiusura: esempio



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

Funzione di transizione estesa per ε-NFA

- La funzione di transizione estesa $\hat{\delta}$ per gli ε-NFA:

  Base:

$$\hat{\delta}(q, \varepsilon) = \mathrm{ECLOSE}(q)$$

  Induzione:

$$\hat{\delta}(q, w) = \mathrm{ECLOSE}\left(\bigcup_{p \in \hat{\delta}(q,x)} \delta(p, a) \right)$$

  con $w = xa$ (parola x seguita dal simbolo a)

- **Esempio:** calcoliamo $\hat{\delta}(q_0, 5.6)$ alla lavagna
- Formalmente, il linguaggio accettato da A è

$$L(A) = \{w : \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$$

Equivalenza di DFA e ε-NFA

- Anche in questo caso abbiamo definito una classe di automi che è equivalente ai DFA
- Per ogni ε-NFA E c'è un DFA D tale che $L(E) = L(D)$, e viceversa
- Lo si dimostra modificando la costruzione a sottoinsiemi:

Equivalenza di DFA e ε-NFA

- Anche in questo caso abbiamo definito una classe di automi che è equivalente ai DFA
- Per ogni ε-NFA E c'è un DFA D tale che $L(E) = L(D)$, e viceversa
- Lo si dimostra modificando la costruzione a sottoinsiemi: Dato un ε-NFA

$$E = (Q_E, \Sigma, q_0, \delta_E, F_E)$$

costruiremo un DFA

$$D = (Q_D, \Sigma, S_0, \delta_D, F_D)$$

tale che

$$L(D) = L(E)$$

La costruzione a sottoinsiemi modificata

- $Q_D = \{S \subseteq Q_E : S = \mathrm{ECLOSE}(S)\}$
  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
- $F_D = \{S \in Q_D : S \cap F_E \neq \emptyset\}$
  Uno stato del DFA è finale se c'è almeno uno stato finale di E
- Per ogni $S \in Q_D$ e per ogni $a \in \Sigma$:

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

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

La costruzione a sottoinsiemi modificata

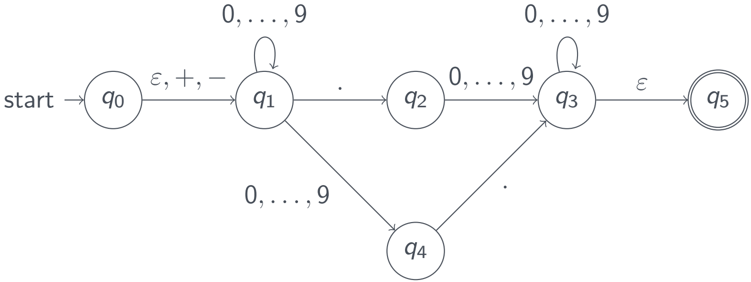
- $Q_D = \{S \subseteq Q_E : S = \mathrm{ECLOSE}(S)\}$
  Ogni stato è un insieme di stati chiuso per ε-transizioni
- $S_0 = \mathrm{ECLOSE}(q_0)$
  Lo stato iniziale è la ε-chiusura dello stato iniziale di E
- $F_D = \{S \in Q_D : S \cap F_E \neq \emptyset\}$
  Uno stato del DFA è finale se c'è almeno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

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

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

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

Costruiamo un DFA D equivalente all'ε-NFA E che riconosce i numeri decimali:



Esempio di costruzione a sottoinsiemi (2)

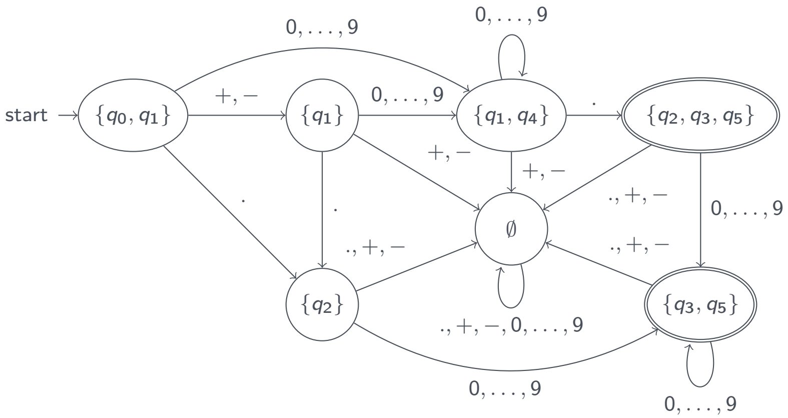
- Come prima cosa costruiamo la ε-chiusura di ogni stato:

$$
\begin{array}{ll}
\mathrm{ECLOSE}(q_0) = \{q_0, q_1\} & \mathrm{ECLOSE}(q_1) = \{q_1\} \\
\mathrm{ECLOSE}(q_2) = \{q_2\} & \mathrm{ECLOSE}(q_3) = \{q_3, q_5\} \\
\mathrm{ECLOSE}(q_4) = \{q_4\} & \mathrm{ECLOSE}(q_5) = \{q_5\}
\end{array}
$$

- Lo stato iniziale di D è $\{q_0, q_1\}$

Esempio di costruzione a sottoinsiemi (1)

- Applicando le regole otteniamo il diagramma di transizione:



Correttezza della costruzione a sottoinsiemi

Theorem

Sia $D = (Q_D, \Sigma, S_0, F_D)$ *il DFA ottenuto da un* ε*-NFA* E *con la costruzione a sottoinsiemi modificata. Allora* $L(D) = L(E)$.

Dimostrazione:

Correttezza della costruzione a sottoinsiemi

Theorem

Sia $D = (Q_D, \Sigma, S_0, F_D)$ il DFA ottenuto da un ε-NFA E con la costruzione a sottoinsiemi modificata. Allora $L(D) = L(E)$.

Dimostrazione: Prima mostriamo per induzione su $|w|$ che

$$\hat{\delta}_D(S_0, w) = \hat{\delta}_E(q_0, w)$$

Correttezza della costruzione a sottoinsiemi

Università degli Studi di Padova

Theorem

Sia $D = (Q_D, \Sigma, S_0, F_D)$ il DFA ottenuto da un ε-NFA E con la costruzione a sottoinsiemi modificata. Allora $L(D) = L(E)$.

Dimostrazione: Prima mostriamo per induzione su $|w|$ che

$$\hat{\delta}_D(S_0, w) = \hat{\delta}_E(q_0, w)$$

Base: $w = \varepsilon$. L'enunciato segue dalla definizione:

- Lo stato iniziale di D è $S_0 = \mathrm{ECLOSE}(q_0)$;
- $\hat{\delta}_D(S_0, \varepsilon) = S_0 = \mathrm{ECLOSE}(q_0)$;
- $\hat{\delta}_E(q_0, \varepsilon) = \mathrm{ECLOSE}(q_0)$.

Induzione:

- Sia $|w| = n+1$ e supponiamo vero l'enunciato per la lunghezza n. Scomponiamo w in $w = xa$ (con $|x| = n$ e a simbolo finale)
- Per ipotesi induttiva $\hat{\delta}_D(S_0, x) = \hat{\delta}_E(q_0, x) = \{p_1, \dots, p_k\}$
- Per la definizione di $\hat{\delta}$ per gli ε-NFA

$$\hat{\delta}_E(q_0, xa) = \mathrm{ECLOSE}\left(\bigcup_{i=1}^{k} \delta_E(p_i, a)\right)$$

- Per la costruzione a sottoinsiemi

$$\delta_D(\{p_1, \dots, p_k\}, a) = \mathrm{ECLOSE}\left(\bigcup_{i=1}^{k} \delta_E(p_i, a)\right)$$

Induzione (continua):

- Per la definizione di $\hat{\delta}$ per i DFA

$$\hat{\delta}_D(S_0, xa) = \delta_D(\{p_1, \ldots, p_k\}, a) = \mathrm{ECLOSE}\left(\bigcup_{i=1}^{k} \delta_E(p_i, a)\right)$$

- Quindi abbiamo mostrato che $\hat{\delta}_D(S_0, w) = \hat{\delta}_E(q_0, w)$

Poiché sia D che E accettano se solo se $\hat{\delta}_D(S_0, w)$ e $\hat{\delta}_E(q_0, w)$ contengono almeno un stato in F_E, allora abbiamo dimostrato che $L(D) = L(N)$.

Teorema di equivalenza tra DFA e NFA

Theorem

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

Dimostrazione:

- La parte "se" è il teorema precedente
- La parte "solo se" si dimostra osservando che ogni DFA può essere trasformato in un ε-NFA modificando δ_D in δ_E con la seguente regola:

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

1. Costruiamo un ε-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'ε-NFA in DFA

Espressioni Regolari

Espressioni Regolari

- Un FA (NFA o DFA) è un metodo per costruire una macchina che riconosce linguaggi regolari
- Una espressione regolare è un modo dichiarativo per descrivere un linguaggio regolare.
- Esempio: $\mathbf{01}^* + \mathbf{10}^*$
- Le espressioni regolari sono usate, ad esempio, in:
  - comandi UNIX (`grep`)
  - strumenti per l'analisi lessicale di UNIX (`lex` (Lexical analyzer generator) e `flex` (Fast Lex))
  - editor di testo

Operazioni sui linguaggi

- **Unione:**

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

- **Concatenazione:**

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

- **Potenze:**

$$L^0 = \{\varepsilon\} \qquad L^1 = L \qquad L^k = L.L^{k-1} = \underbrace{L.L.L.\dots.L}_{k \text{ volte}}$$

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

$$L^* = \bigcup_{i=0}^{\infty} L^i$$

Espressione regolari: sintassi

Le Espressioni Regolari sono costruite utilizzando

Espressione regolari: sintassi

Le Espressioni Regolari sono costruite utilizzando

- un insieme di costanti di base:
  - ε per la stringa vuota
  - $\emptyset$ per il linguaggio vuoto
  - $\mathbf{a}, \mathbf{b}, \ldots$ per i simboli $a, b, \cdots \in \Sigma$

Espressione regolari: sintassi

Le Espressioni Regolari sono costruite utilizzando

- un insieme di costanti di base:
  - ε per la stringa vuota
  - $\emptyset$ per il linguaggio vuoto
  - $\mathbf{a}, \mathbf{b}, \ldots$ per i simboli $a, b, \cdots \in \Sigma$
- collegati da operatori:
  - $+$ per l'unione
  - $\cdot$ per la concatenazione
  - $*$ per la chiusura di Kleene
- raggruppati usando le parentesi:
  - $(\quad)$

Espressioni regolari: semantica

Se **E** è un espressione regolare, allora $L(\mathbf{E})$ è il linguaggio denotato da **E**. La definizione di $L(\mathbf{E})$ è induttiva:

- **Caso Base:**
  - $L(\varepsilon) = \{\varepsilon\}$
  - $L(\emptyset) = \emptyset$
  - $L(\mathbf{a}) = \{a\}$

Se $\mathbf{E}$ è un espressione regolare, allora $L(\mathbf{E})$ è il linguaggio denotato da $\mathbf{E}$. La definizione di $L(\mathbf{E})$ è induttiva:

- **Caso Base:**
  - $L(\varepsilon) = \{\varepsilon\}$
  - $L(\emptyset) = \emptyset$
  - $L(\mathbf{a}) = \{a\}$
- **Caso induttivo:**
  - $L(\mathbf{E} + \mathbf{F}) = L(\mathbf{E}) \cup L(\mathbf{F})$
  - $L(\mathbf{EF}) = L(\mathbf{E}).L(\mathbf{F})$
  - $L(\mathbf{E}^*) = L(\mathbf{E})^*$
  - $L((\mathbf{E})) = L(\mathbf{E})$

- Scriviamo l'espressione regolare per

$$L = \{w \in \{0,1\}^* : 0 \text{ e } 1 \text{ alternati in } w\}$$

- Scriviamo l'espressione regolare per

$$L = \{w \in \{0,1\}^* : \text{0 e 1 alternati in } w\}$$

$$\mathbf{(01)}^* + \mathbf{(10)}^* + \mathbf{1(01)}^* + \mathbf{0(10)}^*$$

- Scriviamo l'espressione regolare per

$$L = \{w \in \{0,1\}^* : 0 \text{ e } 1 \text{ alternati in } w\}$$

$$(\mathbf{01})^* + (\mathbf{10})^* + \mathbf{1}(\mathbf{01})^* + \mathbf{0}(\mathbf{10})^*$$

oppure

$$(\varepsilon + \mathbf{1})(\mathbf{01})^*(\varepsilon + \mathbf{0})$$

Espressioni regolari: precedenza

Come per le espressioni aritmetiche, anche per le espressioni regolari ci sono delle regole di precedenza degli operatori:

1. Chiusura di Kleene
2. Concatenazione (punto)
3. Unione (+)

Esempio:

$$\mathbf{01}^* + \mathbf{1} \text{ è raggruppato in } (\mathbf{0}(\mathbf{1})^*) + \mathbf{1}$$

e denota un linguaggio diverso da

$$(\mathbf{01})^* + \mathbf{1}$$

Esercizi (1)

Per ognuno dei seguenti linguaggi, costruire una ER sull'alfabeto $\{a, b, c\}$ che li rappresenti:

1. Tutte le stringhe w che contengono un numero pari di a;
2. Tutte le stringhe w che contengono $4k + 1$ occorrenze di b, per ogni $k \geq 0$;
3. Tutte le stringhe la cui lunghezza è un multiplo di 3;

Per ognuno dei seguenti linguaggi, costruire una ER sull'alfabeto $\{0, 1\}$ che li rappresenti:

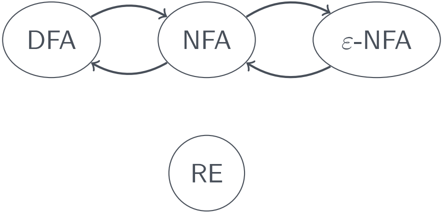
4. Tutte le stringhe w che contengono la sottostringa 101
5. Tutte le stringhe w che non contengono la sottostringa 101

Sfida!

Costruire una ER sull'alfabeto $\{0, 1\}$ per il linguaggio di tutti i numeri binari multipli di 3.

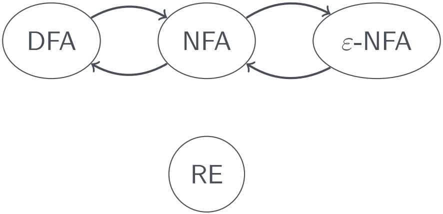
Equivalenza tra FA e RE

Sappiamo già che DFA, NFA, e ε-NFA sono tutti equivalenti.



Equivalenza tra FA e RE

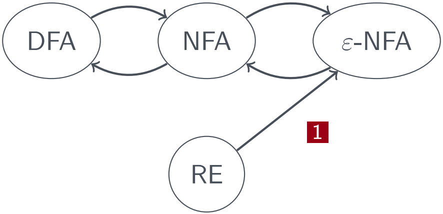
Sappiamo già che DFA, NFA, e ε-NFA sono tutti equivalenti.



Gli FA sono equivalenti alle espressioni regolari:

Equivalenza tra FA e RE

Sappiamo già che DFA, NFA, e ε-NFA sono tutti equivalenti.

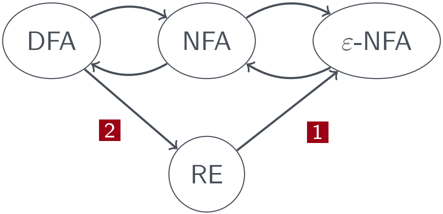


Gli FA sono equivalenti alle espressioni regolari:

1. Per ogni espressione regolare $\mathbf{R}$ esiste un ε-NFA A, tale che $L(A) = L(\mathbf{R})$

Equivalenza tra FA e RE

Sappiamo già che DFA, NFA, e ε-NFA sono tutti equivalenti.



Gli FA sono equivalenti alle espressioni regolari:

1. Per ogni espressione regolare $\mathbf{R}$ esiste un ε-NFA A, tale che $L(A) = L(\mathbf{R})$
2. Per ogni DFA A possiamo costruire un'espressione regolare $\mathbf{R}$, tale che $L(\mathbf{R}) = L(A)$

Da RE a ε-NFA

Theorem

Per ogni espressione regolare $\mathbf{R}$ *possiamo costruire un* ε*-NFA* A *tale che* $L(A) = L(\mathbf{R})$

Università degli Studi di Padova

Theorem

Per ogni espressione regolare $\mathbf{R}$ *possiamo costruire un* ε*-NFA* A *tale che* $L(A) = L(\mathbf{R})$

Dimostrazione:

Costruiremo un ε-NFA A con:

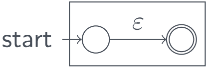
- un solo stato finale
- nessuna transizione entrante nello stato iniziale
- nessuna transizione uscente dallo stato finale

La dimostrazione è per induzione strutturale su $\mathbf{R}$

Caso Base:

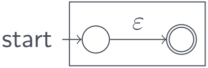
Caso Base:

- automa per ε

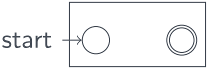


Caso Base:

- automa per ε



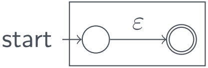
- automa per $\emptyset$



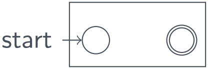
Da RE a ε-NFA

Caso Base:

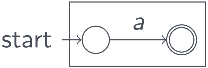
- automa per ε



- automa per $\emptyset$



- automa per **a**

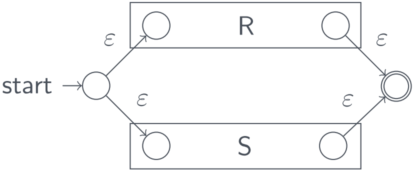


UNIVERSITÀ DEGLI STUDI DI PADOVA

Caso Induttivo:

Caso Induttivo:

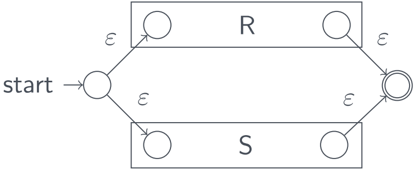
- automa per $\mathbf{R} + \mathbf{S}$



UNIVERSITÀ DEGLI STUDI DI PADOVA

Caso Induttivo:

- automa per $\mathbf{R} + \mathbf{S}$

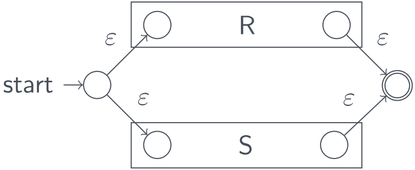


- automa per **RS**

Da RE a ε-NFA

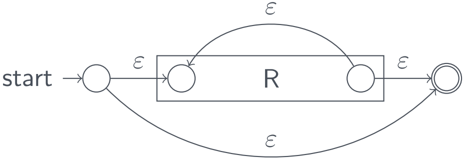
Caso Induttivo:

- automa per $\mathbf{R} + \mathbf{S}$



- automa per $\mathbf{RS}$

- automa per $\mathbf{R}^*$



Esercizio

Trasformiamo $(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})$ in ε-NFA