STRUMENTI FORMALI PER LA BIOINFORMATICA

Automi finiti -
Parte 2. Automi e Grammatiche regolari

Vogliamo dimostrare che gli automi finiti deterministici sono la controparte, in termini di automi, delle grammatiche del livello più basso della gerarchia di Chomsky: le grammatiche di tipo 3 o regolari.

Vogliamo dimostrare che gli automi finiti deterministici sono la controparte, in termini di automi, delle grammatiche del livello più basso della gerarchia di Chomsky: le grammatiche di tipo 3 o regolari.

Faremo riferimento alla definizione estesa di grammatica di tipo 3.

Vogliamo dimostrare che gli automi finiti deterministici sono la controparte, in termini di automi, delle grammatiche del livello più basso della gerarchia di Chomsky: le grammatiche di tipo 3 o regolari.

Faremo riferimento alla definizione estesa di grammatica di tipo 3.

Quindi il potere computazionale delle grammatiche di tipo 3 è lo stesso di quello degli automi.

Vogliamo dimostrare che gli automi finiti deterministici sono la controparte, in termini di automi, delle grammatiche del livello più basso della gerarchia di Chomsky: le grammatiche di tipo 3 o regolari.

Faremo riferimento alla definizione estesa di grammatica di tipo 3.

Quindi il potere computazionale delle grammatiche di tipo 3 è lo stesso di quello degli automi.

Otteniamo una nuova caratterizzazione dei linguaggi regolari.

Definizione (Grammatica regolare)

Una grammatica $G = (V, T, P, S)$ è una grammatica di tipo 3 o regolare se ogni produzione è della forma $A \rightarrow aB$ o $A \rightarrow a$, dove A, B sono variabili e a è un terminale.

Definizione (Grammatica regolare)

Una grammatica $G = (V, T, P, S)$ è una grammatica di tipo 3 o regolare se ogni produzione è della forma $A \to aB$ o $A \to a$, dove A, B sono variabili e a è un terminale.

Inoltre, se $S \to \epsilon$ è una produzione in P allora S non compare alla destra di nessuna produzione in P.

Definizione (Automa a stati finiti deterministico)

Un automa a stati finiti deterministico (in breve DFA) è una quintupla

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

dove:

- Q *è l'insieme* finito *degli stati;*
- Σ *è l'alfabeto (finito);*
- $\delta : Q \times \Sigma \to Q$ *è la funzione di transizione;*
- $q_0 \in Q$ *è lo stato iniziale;*
- $F \subseteq Q$ *è l'insieme degli stati finali.*

Definizione (Automa a stati finiti non deterministico)

Un automa a stati finiti non deterministico (in breve NFA) è una quintupla

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

dove:

- Q *è l'insieme* finito *degli stati;*
- Σ *è l'alfabeto (finito);*
- $\delta \ : \ Q \times \Sigma_\epsilon \to \mathcal{P}(Q)$ *è la funzione di transizione,* *con* $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$ *e dove* $\mathcal{P}(Q)$ *è l'insieme potenza di* Q*;*
- $q_0 \in Q$ *è lo stato iniziale;*
- $F \subseteq Q$ *è l'insieme degli stati finali.*

Proveremo le affermazioni seguenti.

Proveremo le affermazioni seguenti.

1. Per ogni automa a stati finiti deterministico $\mathcal{A}$ esiste una grammatica regolare G tale che $L(\mathcal{A}) = L(G)$.

Proveremo le affermazioni seguenti.

1. Per ogni automa a stati finiti deterministico $\mathcal{A}$ esiste una grammatica regolare G tale che $L(\mathcal{A}) = L(G)$.

2. Per ogni grammatica regolare G esiste un automa a stati finiti non deterministico $\mathcal{A}$ tale che $L(\mathcal{A}) = L(G)$.

Nota:

Nota:

- Le prove sono costruttive.

Nota:

- Le prove sono costruttive.

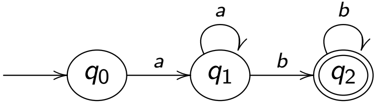
- A partire da un automa a stati finiti deterministico $\mathcal{A}$, costruiremo una grammatica regolare G. La grammatica regolare G sarà tale che $L(\mathcal{A}) = L(G)$.

Nota:

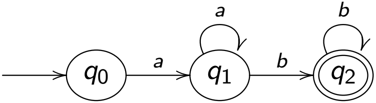
- Le prove sono costruttive.
- A partire da un automa a stati finiti deterministico $\mathcal{A}$, costruiremo una grammatica regolare G. La grammatica regolare G sarà tale che $L(\mathcal{A}) = L(G)$.
- A partire da una grammatica regolare G costruiremo un automa a stati finiti non deterministico $\mathcal{A}$. L'automa a stati finiti $\mathcal{A}$ sarà tale che $L(\mathcal{A}) = L(G)$.

Consideriamo il DFA in figura in cui abbiamo omesso lo stato pozzo q_3.

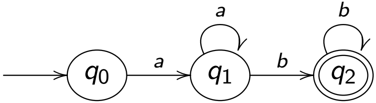
Consideriamo il DFA in figura in cui abbiamo omesso lo stato pozzo q_3.



Dall'automa alla grammatica

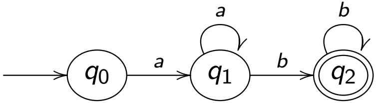


Dall'automa alla grammatica



La computazione dell'automa su *aabbb*

$$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{b} q_2 \xrightarrow{b} q_2$$

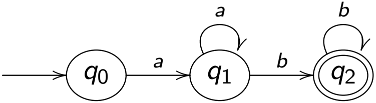


La computazione dell'automa su *aabbb*

$$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{b} q_2 \xrightarrow{b} q_2$$

Come potremmo simularla con le derivazioni in una grammatica?

Dall’automa alla grammatica

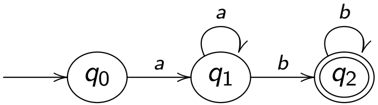


La computazione dell’automa su *aabbb*

$$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{b} q_2 \xrightarrow{b} q_2$$

Come potremmo simularla con le derivazioni in una grammatica?

$$q_0 \Rightarrow aq_1 \Rightarrow aaq_1 \Rightarrow aabq_2 \Rightarrow aabbq_2 \Rightarrow aabbb$$



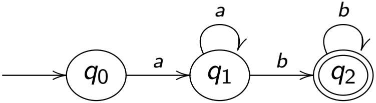
La computazione dell'automa su *aabbb*

$$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{b} q_2 \xrightarrow{b} q_2$$

Come potremmo simularla con le derivazioni in una grammatica?

$$q_0 \Rightarrow aq_1 \Rightarrow aaq_1 \Rightarrow aabq_2 \Rightarrow aabbq_2 \Rightarrow aabbb$$

E quindi quali produzioni?



La computazione dell'automa su *aabbb*

$$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{b} q_2 \xrightarrow{b} q_2$$

Come potremmo simularla con le derivazioni in una grammatica?

$$q_0 \Rightarrow aq_1 \Rightarrow aaq_1 \Rightarrow aabq_2 \Rightarrow aabbq_2 \Rightarrow aabbb$$

E quindi quali produzioni?

$$q_0 \rightarrow a\,q_1, \quad q_1 \rightarrow a\,q_1 \mid b\,q_2 \mid b, \quad q_2 \rightarrow b\,q_2 \mid b$$

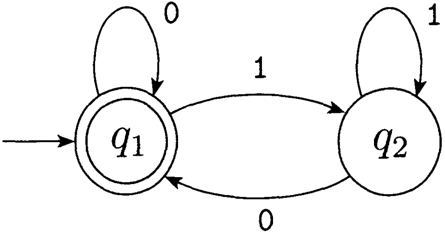


Figura: Un DFA che accetta $\{0,1\}^* \setminus \{w1 \mid w \in \{0,1\}^*\}$

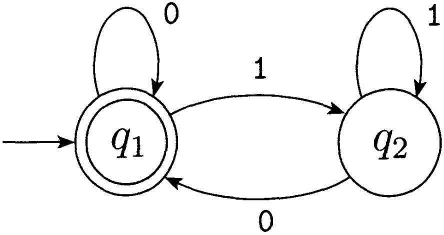


Figura: Un DFA che accetta $\{0,1\}^* \setminus \{w1 \mid w \in \{0,1\}^*\}$

$$q_1 \to 0q_1 \mid 0 \mid 1q_2, \quad q_2 \to 1q_2 \mid 0q_1 \mid 0$$

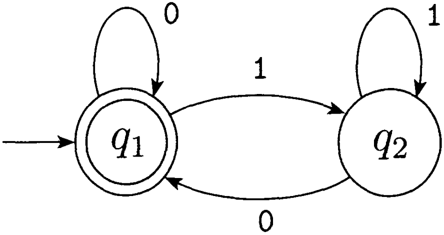


Figura: Un DFA che accetta $\{0,1\}^* \setminus \{w1 \mid w \in \{0,1\}^*\}$

$$q_1 \to 0q_1 \mid 0 \mid 1q_2, \quad q_2 \to 1q_2 \mid 0q_1 \mid 0$$

$$S \to 0q_1 \mid 0 \mid 1q_2 \mid \epsilon$$

Teorema

Per ogni automa finito deterministico $\mathcal{A}$ esiste una grammatica regolare G tale che $L(\mathcal{A}) = L(G)$.

Teorema

Per ogni automa finito deterministico $\mathcal{A}$ esiste una grammatica regolare G tale che $L(\mathcal{A}) = L(G)$.

- La grammatica che simula l'automa avrà come insieme di variabili (un insieme in corrispondenza biunivoca con) l'insieme degli stati di $\mathcal{A}$. Una variabile ausiliaria S è necessaria se $L(\mathcal{A})$ contiene la parola vuota, cioè se $q_0 \in F$.

Dall'automa alla grammatica

Teorema

Per ogni automa finito deterministico $\mathcal{A}$ esiste una grammatica regolare G tale che $L(\mathcal{A}) = L(G)$.

- La grammatica che simula l'automa avrà come insieme di variabili (un insieme in corrispondenza biunivoca con) l'insieme degli stati di $\mathcal{A}$. Una variabile ausiliaria S è necessaria se $L(\mathcal{A})$ contiene la parola vuota, cioè se $q_0 \in F$.
- A ogni transizione da uno stato a un altro etichettata da un simbolo è associata una produzione se lo stato di arrivo non è finale, due se lo stato è finale.

Teorema

Per ogni automa finito deterministico $\mathcal{A}$ esiste una grammatica regolare G tale che $L(\mathcal{A}) = L(G)$.

- La grammatica che simula l'automa avrà come insieme di variabili (un insieme in corrispondenza biunivoca con) l'insieme degli stati di $\mathcal{A}$. Una variabile ausiliaria S è necessaria se $L(\mathcal{A})$ contiene la parola vuota, cioè se $q_0 \in F$.
- A ogni transizione da uno stato a un altro etichettata da un simbolo è associata una produzione se lo stato di arrivo non è finale, due se lo stato è finale.
- Il simbolo iniziale sarà lo stato iniziale di $\mathcal{A}$ se $L(\mathcal{A})$ non contiene la parola vuota, sarà S se $L(\mathcal{A})$ contiene la parola vuota. In questo secondo caso, occorre aggiungere la produzione $S \to \epsilon$ e ulteriori altre produzioni.

Prova del teorema. Sia $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un automa finito deterministico.

Prova del teorema. Sia $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un automa finito deterministico.

Consideriamo una grammatica G definita come segue:

$$G = \begin{cases} (Q, \Sigma, P, q_0) & \text{se } q_0 \notin F \\ (Q \cup \{S\}, \Sigma, P', S) & \text{se } q_0 \in F \end{cases}$$

Prova del teorema. Sia $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un automa finito deterministico.

Consideriamo una grammatica G definita come segue:

$$G = \begin{cases} (Q, \Sigma, P, q_0) & \text{se } q_0 \notin F \\ (Q \cup \{S\}, \Sigma, P', S) & \text{se } q_0 \in F \end{cases}$$

con $S \notin Q$ e P, P' definiti come segue

Prova del teorema. Sia $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un automa finito deterministico.

Consideriamo una grammatica G definita come segue:

$$G = \begin{cases} (Q, \Sigma, P, q_0) & \text{se } q_0 \notin F \\ (Q \cup \{S\}, \Sigma, P', S) & \text{se } q_0 \in F \end{cases}$$

con $S \notin Q$ e P, P' definiti come segue

1. $B \to aC$ è in P se $\delta(B, a) = C$

Prova del teorema. Sia $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un automa finito deterministico.

Consideriamo una grammatica G definita come segue:

$$G = \begin{cases} (Q, \Sigma, P, q_0) & \text{se } q_0 \notin F \\ (Q \cup \{S\}, \Sigma, P', S) & \text{se } q_0 \in F \end{cases}$$

con $S \notin Q$ e P, P' definiti come segue

1. $B \to aC$ è in P se $\delta(B, a) = C$
2. $B \to a$ è in P se $\delta(B, a) = C$ e $C \in F$

Prova del teorema. Sia $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un automa finito deterministico.

Consideriamo una grammatica G definita come segue:

$$G = \begin{cases} (Q, \Sigma, P, q_0) & \text{se } q_0 \notin F \\ (Q \cup \{S\}, \Sigma, P', S) & \text{se } q_0 \in F \end{cases}$$

con $S \notin Q$ e P, P' definiti come segue

1. $B \to aC$ è in P se $\delta(B, a) = C$
2. $B \to a$ è in P se $\delta(B, a) = C$ e $C \in F$
3. $P' = P \cup \{S \to \epsilon\} \cup \{S \to \alpha \mid q_0 \to \alpha \in P\}$

Prova del teorema. Sia $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un automa finito deterministico.

Consideriamo una grammatica G definita come segue:

$$G = \begin{cases} (Q, \Sigma, P, q_0) & \text{se } q_0 \notin F \\ (Q \cup \{S\}, \Sigma, P', S) & \text{se } q_0 \in F \end{cases}$$

con $S \notin Q$ e P, P' definiti come segue

1. $B \to aC$ è in P se $\delta(B, a) = C$
2. $B \to a$ è in P se $\delta(B, a) = C$ e $C \in F$
3. $P' = P \cup \{S \to \epsilon\} \cup \{S \to \alpha \mid q_0 \to \alpha \in P\}$

Si può dimostrare che $L(\mathcal{A}) = L(G)$.

Dall'automa alla grammatica

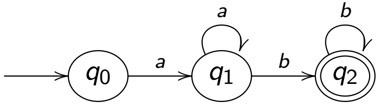
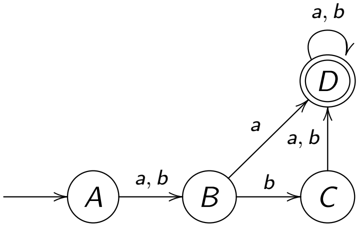


Figura: Un DFA in cui lo stato pozzo q_3 non è stato disegnato.

$$
\begin{aligned}
G &= (V, T, P, q_0) \\
V &= \{q_0, q_1, q_2, q_3\} \\
T &= \{a, b\} \\
P &= \{q_0 \to aq_1 \mid bq_3, \quad q_1 \to aq_1 \mid bq_2 \mid b, \quad q_2 \to bq_2 \mid b \mid aq_3, \\
&\quad\; q_3 \to aq_3 \mid bq_3\}
\end{aligned}
$$



$$G = (V, T, P, A), \quad V = \{A, B, C, D\}, \quad T = \{a, b\}$$

P contiene le produzioni

$$A \to aB \mid bB, \qquad B \to bC \mid aD \mid a,$$

$$C \to aD \mid a \mid bD \mid b, \qquad D \to aD \mid a \mid bD \mid b$$

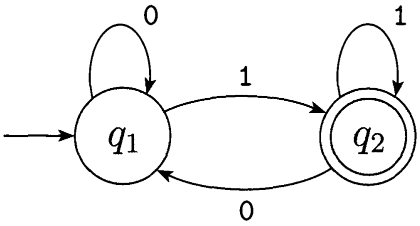


Figura: La grammatica equivalente ha simbolo iniziale q_1 ed è definita dalle produzioni $\{q_1 \rightarrow 0q_1 \mid 1q_2 \mid 1,\ q_2 \rightarrow 0q_1 \mid 1q_2 \mid 1\}$

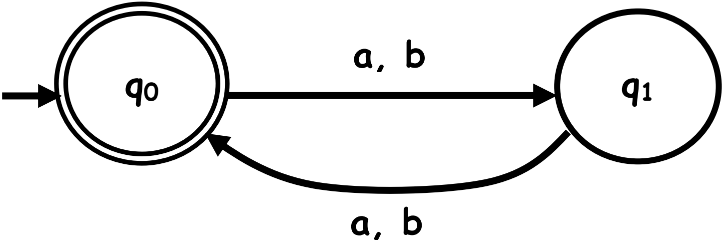


Figura: La grammatica equivalente ha simbolo iniziale S ed è definita dalle produzioni
$\{q_0 \rightarrow aq_1 \mid bq_1,\ q_1 \rightarrow aq_0 \mid bq_0 \mid a \mid b,\ S \rightarrow aq_1 \mid bq_1 \mid \epsilon\}$

L'algoritmo di costruzione dell'automa a partire da una grammatica regolare non consiste nella trasformazione inversa di quella che fornisce una grammatica regolare a partire da un DFA.

Data una grammatica regolare, la trasformazione inversa non è sempre possibile.

Esempio 1. Consideriamo la grammatica regolare $G_1 = (\{S, B, C\}, \{0, 1\}, P_1, S)$, dove le produzioni in P_1 sono:

$$S \to 0S \mid 1S \mid 1B, \quad B \to 0C \mid 1C, \quad C \to 0 \mid 1$$

Esempio 1. Consideriamo la grammatica regolare $G_1 = (\{S, B, C\}, \{0, 1\}, P_1, S)$, dove le produzioni in P_1 sono:

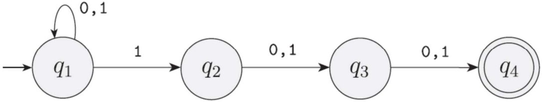
$$S \to 0S \mid 1S \mid 1B, \quad B \to 0C \mid 1C, \quad C \to 0 \mid 1$$

Non è possibile individuare un automa finito deterministico $\mathcal{A}$ a cui applicare l'algoritmo che abbiamo visto e che trasformi $\mathcal{A}$ in G_1. E questo non dipende solo dal fatto che partiamo da un DFA.

L'algoritmo che vedremo trasformerà la grammatica regolare del precedente esempio, definita dalle produzioni

$$S \to 0S \mid 1S \mid 1B, \quad B \to 0C \mid 1C, \quad C \to 0 \mid 1$$

nell'automa seguente.



Teorema

Per ogni grammatica regolare G esiste un automa finito non deterministico tale che $L(\mathcal{A}) = L(G)$.

Teorema

Per ogni grammatica regolare G esiste un automa finito non deterministico tale che $L(\mathcal{A}) = L(G)$.

- L'automa avrà come come insieme degli stati (un insieme in corrispondenza biunivoca con) l'insieme delle variabili di G più uno stato A che sarà uno stato finale.

Teorema
Per ogni grammatica regolare G esiste un automa finito non deterministico tale che $L(\mathcal{A}) = L(G)$.

- L'automa avrà come come insieme degli stati (un insieme in corrispondenza biunivoca con) l'insieme delle variabili di G più uno stato A che sarà uno stato finale.
- Se $S \to \epsilon$ è in P, dove S è il simbolo iniziale della grammatica, allora anche S sarà finale. Ricordiamo che in tal caso S non compare a destra di nessuna produzione in G.

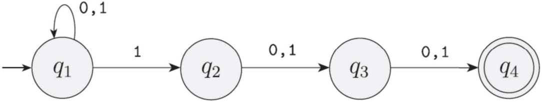
Teorema

Per ogni grammatica regolare G esiste un automa finito non deterministico tale che $L(\mathcal{A}) = L(G)$.

- L'automa avrà come come insieme degli stati (un insieme in corrispondenza biunivoca con) l'insieme delle variabili di G più uno stato A che sarà uno stato finale.
- Se $S \rightarrow \epsilon$ è in P, dove S è il simbolo iniziale della grammatica, allora anche S sarà finale. Ricordiamo che in tal caso S non compare a destra di nessuna produzione in G.
- A ogni produzione corrisponde una transizione dell'automa. Dallo stato A non parte nessuna transizione.

Automa per la grammatica definita da

$S \to 0S \mid 1S \mid 1B, \quad B \to 0C \mid 1C, \quad C \to 0 \mid 1$



Prova del teorema. Sia $G = (V, \Sigma, P, S)$ una grammatica regolare. Consideriamo un automa finito non deterministico $\mathcal{A} = (Q, \Sigma, \delta, S, F)$ definito come segue:

Prova del teorema. Sia $G = (V, \Sigma, P, S)$ una grammatica regolare. Consideriamo un automa finito non deterministico $\mathcal{A} = (Q, \Sigma, \delta, S, F)$ definito come segue:

$$Q = V \cup \{A\}, \quad A \notin V$$

$$F = \begin{cases} \{A\} & \text{se } S \to \epsilon \notin P \\ \{S, A\} & \text{se } S \to \epsilon \in P \end{cases}$$

Prova del teorema. Sia $G = (V, \Sigma, P, S)$ una grammatica regolare. Consideriamo un automa finito non deterministico $\mathcal{A} = (Q, \Sigma, \delta, S, F)$ definito come segue:

$$Q = V \cup \{A\}, \quad A \notin V$$

$$F = \begin{cases} \{A\} & \text{se } S \to \epsilon \notin P \\ \{S, A\} & \text{se } S \to \epsilon \in P \end{cases}$$

e con δ tale che

Prova del teorema. Sia $G = (V, \Sigma, P, S)$ una grammatica regolare. Consideriamo un automa finito non deterministico $\mathcal{A} = (Q, \Sigma, \delta, S, F)$ definito come segue:

$$Q = V \cup \{A\}, \quad A \notin V$$

$$F = \begin{cases} \{A\} & \text{se } S \to \epsilon \notin P \\ \{S, A\} & \text{se } S \to \epsilon \in P \end{cases}$$

e con δ tale che

1. $C \in \delta(B, a)$ se $B \to aC$ è in P,

Prova del teorema. Sia $G = (V, \Sigma, P, S)$ una grammatica regolare. Consideriamo un automa finito non deterministico $\mathcal{A} = (Q, \Sigma, \delta, S, F)$ definito come segue:

$$Q = V \cup \{A\}, \quad A \notin V$$

$$F = \begin{cases} \{A\} & \text{se } S \to \epsilon \notin P \\ \{S, A\} & \text{se } S \to \epsilon \in P \end{cases}$$

e con δ tale che

1. $C \in \delta(B, a)$ se $B \to aC$ è in P,
2. $A \in \delta(B, a)$ se $B \to a$ è in P

Prova del teorema. Sia $G = (V, \Sigma, P, S)$ una grammatica regolare. Consideriamo un automa finito non deterministico $\mathcal{A} = (Q, \Sigma, \delta, S, F)$ definito come segue:

$$Q = V \cup \{A\}, \quad A \notin V$$

$$F = \begin{cases} \{A\} & \text{se } S \to \epsilon \notin P \\ \{S, A\} & \text{se } S \to \epsilon \in P \end{cases}$$

e con δ tale che

1. $C \in \delta(B, a)$ se $B \to aC$ è in P,
2. $A \in \delta(B, a)$ se $B \to a$ è in P
3. $\delta(A, a) = \emptyset$, per ogni $a \in \Sigma$

Prova del teorema. Sia $G = (V, \Sigma, P, S)$ una grammatica regolare. Consideriamo un automa finito non deterministico $\mathcal{A} = (Q, \Sigma, \delta, S, F)$ definito come segue:

$$Q = V \cup \{A\}, \quad A \notin V$$

$$F = \begin{cases} \{A\} & \text{se } S \to \epsilon \notin P \\ \{S, A\} & \text{se } S \to \epsilon \in P \end{cases}$$

e con δ tale che

1. $C \in \delta(B, a)$ se $B \to aC$ è in P,
2. $A \in \delta(B, a)$ se $B \to a$ è in P
3. $\delta(A, a) = \emptyset$, per ogni $a \in \Sigma$

Si può dimostrare che $L(\mathcal{A}) = L(G)$.

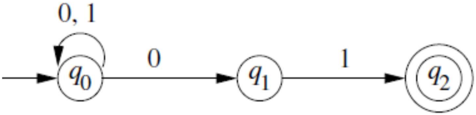
Esempio. Sia $G_1 = (\{S, B\}, \{0, 1\}, P_1, S)$, dove

$$P_1 = \{S \to 0S \mid 1S \mid 0B,\ \ B \to 1\}$$

Esempio. Sia $G_1 = (\{S, B\}, \{0, 1\}, P_1, S)$, dove

$$P_1 = \{S \to 0S \mid 1S \mid 0B,\ \ B \to 1\}$$

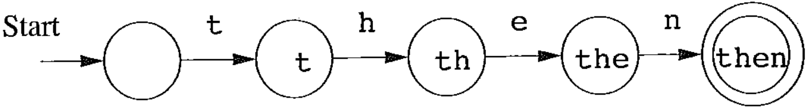
Un NFA $\mathcal{A}_1$ tale che $L(\mathcal{A}_1) = L(G_1)$



Esempio. Sia $G_2 = (\{S, B, C, D\}, \{t, h, e, n\}, P_2, S)$, dove $P_2 = \{S \rightarrow tB,\ B \rightarrow hC,\ C \rightarrow eD,\ D \rightarrow n\}$.

Esempio. Sia $G_2 = (\{S, B, C, D\}, \{t, h, e, n\}, P_2, S)$, dove $P_2 = \{S \to tB,\ B \to hC,\ C \to eD,\ D \to n\}$.

Un NFA $\mathcal{A}_2$ tale che $L(\mathcal{A}_2) = L(G_2)$



Esempio. Consideriamo la grammatica regolare
$G = (\{S, B\}, \{0, 1\}, P, S)$, dove le produzioni in P sono:

$$S \rightarrow 0B, \qquad B \rightarrow 0B \mid 1S \mid 0$$

Esempio. Consideriamo la grammatica regolare
$G = (\{S, B\}, \{0, 1\}, P, S)$, dove le produzioni in P sono:

$$S \to 0B, \qquad B \to 0B \mid 1S \mid 0$$

Un automa finito non deterministico $\mathcal{A}$ tale che $L(\mathcal{A}) = L(G)$
è

$$\mathcal{A} = (\{S, A, B\}, \{0, 1\}, \delta, S, \{A\})$$

dove δ è definita come segue.

Esempio. Consideriamo la grammatica regolare $G = (\{S, B\}, \{0, 1\}, P, S)$, dove le produzioni in P sono:

$$S \to 0B, \qquad B \to 0B \mid 1S \mid 0$$

Un automa finito non deterministico $\mathcal{A}$ tale che $L(\mathcal{A}) = L(G)$ è

$$\mathcal{A} = (\{S, A, B\}, \{0, 1\}, \delta, S, \{A\})$$

dove δ è definita come segue.
In base alla costruzione del teorema abbiamo

$$B \in \delta(S, 0), \quad B \in \delta(B, 0), \quad S \in \delta(B, 1), \quad A \in \delta(B, 0)$$

Quindi

$$\begin{aligned} \delta(S, 0) &= \{B\}, \quad \delta(S, 1) = \emptyset \\ \delta(B, 0) &= \{A, B\}, \quad \delta(B, 1) = \{S\} \\ \delta(A, 0) &= \delta(A, 1) = \emptyset \end{aligned}$$

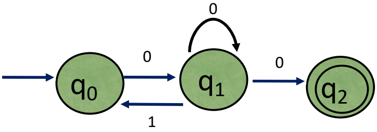
Sia $G = (\{S, B\}, \{0, 1\}, P, S)$, dove

$$P = \{S \to 0B,\ \ B \to 0B \mid 1S \mid 0\}$$

Sia $G = (\{S, B\}, \{0, 1\}, P, S)$, dove

$$P = \{S \to 0B,\ \ B \to 0B \mid 1S \mid 0\}$$

Un NFA $\mathcal{A}$ tale che $L(G) = L(\mathcal{A}) = L(0(10 + 0)^*0)$



Teorema

Un linguaggio L è regolare se e solo se esiste una grammatica regolare G che lo genera, cioè tale che $L = L(G)$.

Teorema

Un linguaggio L è regolare se e solo se esiste una grammatica regolare G che lo genera, cioè tale che $L = L(G)$.

Come conseguenza del teorema possiamo affermare che ogni linguaggio regolare è context-free.

Teorema

Un linguaggio L è regolare se e solo se esiste una grammatica regolare G che lo genera, cioè tale che $L = L(G)$.

Come conseguenza del teorema possiamo affermare che ogni linguaggio regolare è context-free.

La classe dei linguaggi regolari è strettamente inclusa nella classe dei linguaggi context-free.

Teorema

Un linguaggio L è regolare se e solo se esiste una grammatica regolare G che lo genera, cioè tale che $L = L(G)$.

Come conseguenza del teorema possiamo affermare che ogni linguaggio regolare è context-free.

La classe dei linguaggi regolari è strettamente inclusa nella classe dei linguaggi context-free.

Infatti $\{a^n b^n \mid n \geq 0\}$ è un linguaggio context-free che non è regolare.