ELEMENTI DI INFORMATICA TEORICA

Parte 2: Linguaggi Formali e Automi

2.2 Automi a Stati Finiti

Giovanni Amendola

Corso di laurea triennale in Informatica
Università della Calabria

6 aprile 2022

Anno Accademico 2021/2022

Riconoscimento di Linguaggi

Dato un linguaggio $\mathcal{L}$, il problema del riconoscimento di $\mathcal{L}$ è il problema decisionale seguente

Problema del riconoscimento (o dell'appartenenza, *membership*)

Data una stringa s, stabilire se s appartiene a $\mathcal{L}$.

- I linguaggi di tipo 3 sono riconosciuti da dispositivi con memoria costante in tempo lineare (**automi a stati finiti**).
- I linguaggi di tipo 2 sono riconosciuti da dispositivi nondeterministici con pila in tempo lineare (**automi a pila non deterministici**).
- I linguaggi di tipo 1 sono riconosciuti da dispositivi nondeterministici con memoria che cresce linearmente con la lunghezza della stringa da esaminare (**automi non deterministici *linear bounded***).
- I linguaggi di tipo 0 non sono tutti decidibili, ma semidecidibili (se la stringa fa parte del linguaggio viene riconosciuta, altrimenti potrebbe non esserlo). Sono riconosciuti dalle **macchine di Turing**.

Automi

- **Automa** $\longrightarrow$ **auto-matos** $=$ “che si muove da sé”

Automi

- **Automa** $\longrightarrow$ **auto-matos** = “che si muove da sé”
- Altri termini correlati:
  - **Macchina** $\longrightarrow$ *mechanè* (*Deus ex machina*)

Automi

- **Automa** $\longrightarrow$ **auto-matos** = "che si muove da sé"
- Altri termini correlati:
  - **Macchina** $\longrightarrow$ *mechanè* (*Deus ex machina*)
  - **Calcolatore**, **Computer**
  - **Intelligenza Artificiale**, **Robot** (androidi)

Automi

- **Automa** $\longrightarrow$ **auto-matos** = "che si muove da sé"
- Altri termini correlati:
  - **Macchina** $\longrightarrow$ *mechanè* (*Deus ex machina*)
  - **Calcolatore**, **Computer**
  - **Intelligenza Artificiale**, **Robot** (androidi) $\longrightarrow$ **antropomorfismi**

Automi

- **Automa** $\longrightarrow$ **auto-matos** = "che si muove da sé"
- Altri termini correlati:
  - **Macchina** $\longrightarrow$ *mechanè* (*Deus ex machina*)
  - **Calcolatore**, **Computer**
  - **Intelligenza Artificiale**, **Robot** (androidi) $\longrightarrow$ **antropomorfismi**

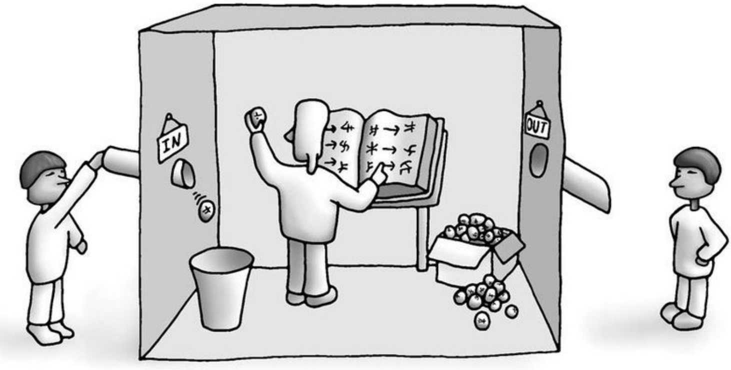
Automi

- **Automa** $\longrightarrow$ **auto-matos** = "che si muove da sé"
- Altri termini correlati:
  - **Macchina** $\longrightarrow$ *mechanè* (*Deus ex machina*)
  - **Calcolatore**, **Computer**
  - **Intelligenza Artificiale**, **Robot** (androidi) $\longrightarrow$ **antropomorfismi**

- **Solo sintassi, senza semantica (significati, comprensione)**

Automi

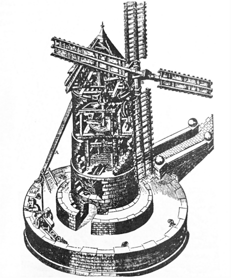
Esperimento mentale della Stanza Cinese (John Searle, 1980)



Solo sintassi: manipolazione meccanica di simboli senza significato

Automi

La macchina-mulino di Leibniz (*Monadologia*, 1714)



«Immaginiamo una **macchina** strutturata in modo tale che sia **capace di pensare, di sentire, di avere percezioni**; supponiamola ora ingrandita, con le stesse proporzioni, in modo che vi si possa entrare come un **mulino**. Fatto ciò, visitando la macchina al suo interno, troveremo sempre e soltanto pezzi che si spingono a vicenda, ma nulla che sia in grado di spiegare una percezione. Quindi **la percezione va cercata nella sostanza semplice**, non già nel Composto, cioè nella macchina».

Automi

- Un automa è una rappresentazione formale di un **modello di calcolo**, composto da:
  - un insieme di *stati*: *accettazione* o non accettazione;
  - un insieme di *regole di transizioni*: passare da uno stato ad un altro in base all'input.

Automi

- Un automa è una rappresentazione formale di un **modello di calcolo**, composto da:
  - un insieme di *stati*: *accettazione* o non accettazione;
  - un insieme di *regole di transizioni*: passare da uno stato ad un altro in base all'input.
- Un automa riceve un **input**, lo **elabora**, ed (eventualmente) restituisce un **output**.
  - La computazione parte da uno *stato iniziale* e dalla lettura di un input.
  - Se alla fine della lettura, l'automa è in uno *stato di accettazione*, si dice che l'automa ha *accettato* l'input, altrimenti lo ha rifiutato.

Automi

Esempio: **Porta automatica**

Rappresentiamo un automa che modella una porta automatica con ante scorrevoli che si aprono quando una persona si trova ad una certa distanza. Abbiamo 4 possibili input e 2 stati della porta:

- Input: $\{davanti, dietro, entrambi, nessuno\}$
- Stati: $\{aperta, chiusa\}$

Automi

Esempio: Porta automatica

Rappresentiamo un automa che modella una porta automatica con ante scorrevoli che si aprono quando una persona si trova ad una certa distanza. Abbiamo 4 possibili input e 2 stati della porta:

- Input: $\{davanti, dietro, entrambi, nessuno\}$
- Stati: $\{aperta, chiusa\}$

Le regole saranno le seguenti:

- Se la porta è *chiusa* e l'input è *davanti*, *dietro* o *entrambi*, si apre.
- Se la porta è *aperta* e l'input è *nessuno*, allora la porta si chiude.
- Se la porta è *chiusa* e l'input è *nessuno*, allora la porta resta chiusa.
- Se la porta è *aperta* e l'input è *davanti*, *dietro* o *entrambi*, resta aperta.

Automi

Esempio: Porta automatica

Rappresentiamo un automa che modella una porta automatica con ante scorrevoli che si aprono quando una persona si trova ad una certa distanza. Abbiamo 4 possibili input e 2 stati della porta:

- Input: $\{davanti, dietro, entrambi, nessuno\}$
- Stati: $\{aperta, chiusa\}$

Le regole saranno le seguenti:

- Se la porta è *chiusa* e l'input è *davanti*, *dietro* o *entrambi*, si apre.
- Se la porta è *aperta* e l'input è *nessuno*, allora la porta si chiude.
- Se la porta è *chiusa* e l'input è *nessuno*, allora la porta resta chiusa.
- Se la porta è *aperta* e l'input è *davanti*, *dietro* o *entrambi*, resta aperta.

Le regole possono rappresentarsi con una **matrice di transizione**:

Stati/Input	*davanti*	*dietro*	*entrambi*	*nessuno*
chiusa	*aperta*	*aperta*	*aperta*	*chiusa*
aperta	*aperta*	*aperta*	*aperta*	*chiusa*

Automa a stati finiti

Definizione: Automa a stati finiti

Un *automa a stati finiti* $\mathcal{A}$ è una quintupla $\langle Q, \Sigma, \delta, q_0, F \rangle$, dove

1. Q è un insieme finito non vuoto di *stati*.
2. Σ è un alfabeto finito non vuoto di input.
3. $\delta : Q \times \Sigma \to Q$ è la *funzione di transizione*.
4. $q_0 \in Q$ è lo *stato iniziale*.
5. $F \subseteq Q$ è l'insieme degli *stati di accettazione*.

Automa a stati finiti

Definizione: Automa a stati finiti

Un *automa a stati finiti* $\mathcal{A}$ è una quintupla $\langle Q, \Sigma, \delta, q_0, F \rangle$, dove

1. Q è un insieme finito non vuoto di *stati*.
2. Σ è un alfabeto finito non vuoto di input.
3. $\delta : Q \times \Sigma \to Q$ è la *funzione di transizione*.
4. $q_0 \in Q$ è lo *stato iniziale*.
5. $F \subseteq Q$ è l'insieme degli *stati di accettazione*.

- Un Automa a Stati Finiti sarà abbreviato con ASF.
- La funzione di transizione di un ASF possiamo rappresentarla con
  - Matrice di transizione
  - Diagramma degli stati

Automa a stati finiti

Esempio

Sia $\mathcal{A} = \langle \{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\} \rangle$, dove

$\delta =$

	0	1
q_0	q_0	q_1
q_0	q_2	q_1
q_2	q_1	q_1

Supponiamo di avere come input la stringa 1101:

Automa a stati finiti

Esempio

Sia $\mathcal{A} = \langle \{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\} \rangle$, dove

$\delta =$

	0	1
q_0	q_0	q_1
q_0	q_2	q_1
q_2	q_1	q_1

Supponiamo di avere come input la stringa 1101:

1. Inizia nello stato q_0.
2. Legge 1 ed esegue la transizione da q_0 a q_1.
3. Legge 1 ed esegue la transizione da q_1 a q_1.
4. Legge 0 ed esegue la transizione da q_1 a q_2.
5. Legge 1 ed esegue la transizione da q_2 a q_1.

Siccome q_1 è uno stato di accettazione, allora $\mathcal{A}$ accetta 1101.

Automa a stati finiti

Esempio

Sia $\mathcal{A} = \langle \{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_1\} \rangle$, dove

$\delta =$

	0	1
q_0	q_0	q_1
q_0	q_2	q_1
q_2	q_1	q_1

Automa a stati finiti

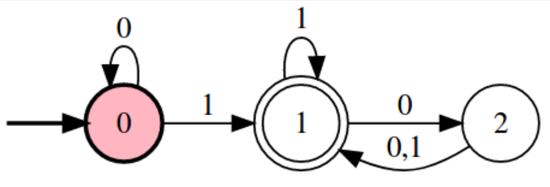
Esempio

Sia $\mathcal{A} = \langle \{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\} \rangle$, dove

$\delta =$

	0	1
q_0	q_0	q_1
q_0	q_2	q_1
q_2	q_1	q_1

Diagramma degli stati:



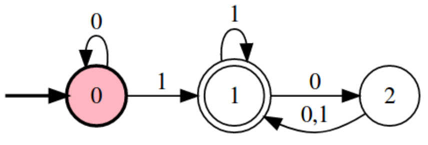
Automa a stati finiti

- Sia S l'insieme delle stringhe accettate da un automa $\mathcal{A}$. Diciamo che S è il **linguaggio** di $\mathcal{A}$ e che $\mathcal{A}$ *accetta* (o *riconosce*) S. Scriviamo $\mathcal{L}(\mathcal{A}) = S$.
- Se $\mathcal{A}$ non accetta nessuna stringa, allora $\mathcal{L}(\mathcal{A}) = \emptyset$.

Automa a stati finiti

- Sia S l'insieme delle stringhe accettate da un automa $\mathcal{A}$. Diciamo che S è il **linguaggio** di $\mathcal{A}$ e che $\mathcal{A}$ *accetta* (o *riconosce*) S. Scriviamo $\mathcal{L}(\mathcal{A}) = S$.
- Se $\mathcal{A}$ non accetta nessuna stringa, allora $\mathcal{L}(\mathcal{A}) = \emptyset$.

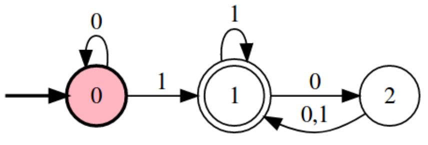
Quali stringhe accetta l'automa dell'esempio precendente?



Automa a stati finiti

- Sia S l'insieme delle stringhe accettate da un automa $\mathcal{A}$. Diciamo che S è il **linguaggio** di $\mathcal{A}$ e che $\mathcal{A}$ *accetta* (o *riconosce*) S. Scriviamo $\mathcal{L}(\mathcal{A}) = S$.
- Se $\mathcal{A}$ non accetta nessuna stringa, allora $\mathcal{L}(\mathcal{A}) = \emptyset$.

Quali stringhe accetta l'automa dell'esempio precendente?



- Tutte le stringhe che terminano con 1.
- Tutte le stringhe che terminano con un numero pari di 0 dopo l'ultimo 1.

$$\mathcal{L}(\mathcal{A}) = \{s \in \Sigma^* \mid s = w1(00)^n, n \geq 0, w \in \Sigma^*\}$$

Automa a stati finiti

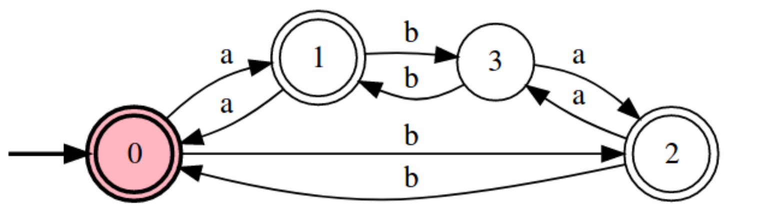
Esempio

Costruiamo un automa che riconosce il linguaggio delle stringhe su $\{a, b\}$ con un numero pari di a o un numero pari di b.

Automa a stati finiti

Esempio

Costruiamo un automa che riconosce il linguaggio delle stringhe su $\{a, b\}$ con un numero pari di a o un numero pari di b.



Automa a stati finiti

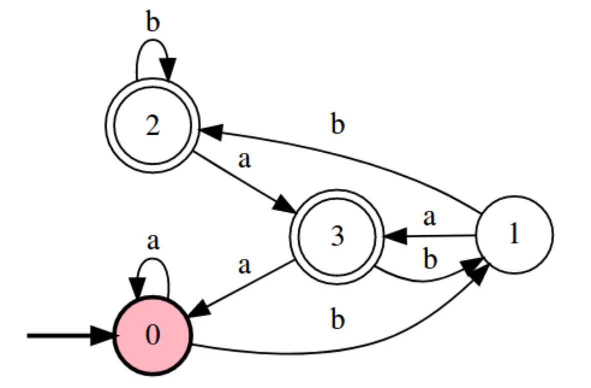
Esempio

Costruiamo un automa che riconosce il linguaggio delle stringhe su $\{a, b\}$ in cui il penultimo carattere è b.

Automa a stati finiti

Esempio

Costruiamo un automa che riconosce il linguaggio delle stringhe su $\{a, b\}$ in cui il penultimo carattere è b.



Automa a stati finiti

Definizione: Configurazione

Sia $\mathcal{A}$ un ASF, una *configurazione* di $\mathcal{A}$ è un elemento $uqv \in \Sigma \times Q \times \Sigma^*$, dove $q \in Q$, $u \in \Sigma$ e $v \in \Sigma^*$. Una configurazione può essere:

- **Iniziale** se $q = q_0$ e $u = \epsilon$;
- **Intermedia** se $u \neq \epsilon$ e $v \neq \epsilon$;
- **Finale** o **di accettazione** se $q \in F$ e $v = \epsilon$.

Automa a stati finiti

Definizione: Configurazione

Sia $\mathcal{A}$ un ASF, una *configurazione* di $\mathcal{A}$ è un elemento $uqv \in \Sigma \times Q \times \Sigma^*$, dove $q \in Q$, $u \in \Sigma$ e $v \in \Sigma^*$. Una configurazione può essere:

- **Iniziale** se $q = q_0$ e $u = \epsilon$;
- **Intermedia** se $u \neq \epsilon$ e $v \neq \epsilon$;
- **Finale** o **di accettazione** se $q \in F$ e $v = \epsilon$.

Definizione: Relazione di derivazione $\vdash_{\mathcal{A}}$

Siano $c_1 = (u_1 q_1 w_1)$ e $c_2 = (u_2 q_2 w_2)$ due configurazioni di $\mathcal{A}$. Diciamo che

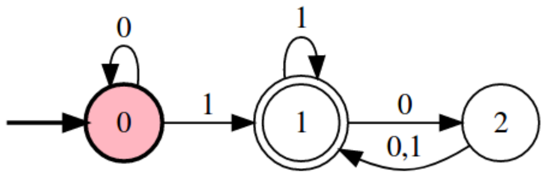
$$c_1 \vdash_{\mathcal{A}} c_2 \text{ sse } \delta(q_1, u_2) = q_2 \text{ e } w_1 = u_2 w_2.$$

Indichiamo con $\vdash_{\mathcal{A}}^*$ la chiusura riflessiva e transitiva della relazione $\vdash_{\mathcal{A}}$.

Automa a stati finiti

Esempio

Consideriamo l'ASF $\mathcal{A}$ del primo esempio.

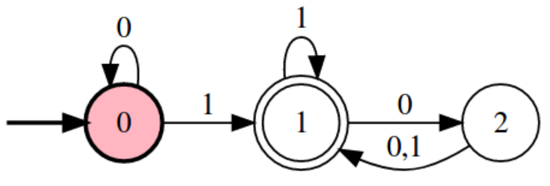


Supponiamo di avere in input la stringa 0101.

Automa a stati finiti

Esempio

Consideriamo l'ASF $\mathcal{A}$ del primo esempio.



Supponiamo di avere in input la stringa 0101. Allora

- $\epsilon q_0 0101$ è la configurazione iniziale;
- Applichiamo la relazione $\vdash_{\mathcal{A}}$:

$$\epsilon q_0 0101 \vdash_{\mathcal{A}} 0 q_0 101 \vdash_{\mathcal{A}} 1 q_1 01 \vdash_{\mathcal{A}} 0 q_2 1 \vdash_{\mathcal{A}} 1 q_1 \epsilon;$$

- $1 q_1 \epsilon$ è una configurazione finale (in quanto $q_1 \in F$).

Linguaggio accettato da un ASF

Definizione: Computazione e Linguaggio accettato

Sia $\mathcal{A}$ un ASF che accetta una stringa $w = w_1 \dots w_m \in \Sigma^*$.

- Un **passo di computazione** di w è una transizione fra due configurazioni c_i e c_j tale che $c_i \vdash_{\mathcal{A}} c_j$, $1 \leq i, j \leq m$.

Linguaggio accettato da un ASF

Definizione: **Computazione** e **Linguaggio accettato**

Sia $\mathcal{A}$ un ASF che accetta una stringa $w = w_1 \dots w_m \in \Sigma^*$.

- Un **passo di computazione** di w è una transizione fra due configurazioni c_i e c_j tale che $c_i \vdash_{\mathcal{A}} c_j$, $1 \leq i, j \leq m$.
- Una **computazione** di w è una sequenza di passi di computazione $c_1 \dots c_m$ di w tale che
    - esiste un $q \in F$ con $c_1 = (\epsilon q_0 w) \vdash^*_{\mathcal{A}} (w_m q \epsilon) = c_m$ e
    - per ogni j, $1 < j \leq m$, si ha che $c_{j-1} \vdash_{\mathcal{A}} c_j$.

Linguaggio accettato da un ASF

Definizione: **Computazione** e **Linguaggio accettato**

Sia $\mathcal{A}$ un ASF che accetta una stringa $w = w_1 \dots w_m \in \Sigma^*$.

- Un **passo di computazione** di w è una transizione fra due configurazioni c_i e c_j tale che $c_i \vdash_{\mathcal{A}} c_j$, $1 \leq i, j \leq m$.
- Una **computazione** di w è una sequenza di passi di computazione $c_1 \dots c_m$ di w tale che
  - esiste un $q \in F$ con $c_1 = (\epsilon q_0 w) \vdash^*_{\mathcal{A}} (w_m q \epsilon) = c_m$ e
  - per ogni j, $1 < j \leq m$, si ha che $c_{j-1} \vdash_{\mathcal{A}} c_j$.
- w è **accettata** da $\mathcal{A}$ se esiste una computazione di w in $\mathcal{A}$.

Linguaggio accettato da un ASF

Definizione: **Computazione** e **Linguaggio accettato**

Sia $\mathcal{A}$ un ASF che accetta una stringa $w = w_1 \dots w_m \in \Sigma^*$.

- Un **passo di computazione** di w è una transizione fra due configurazioni c_i e c_j tale che $c_i \vdash_{\mathcal{A}} c_j$, $1 \leq i, j \leq m$.
- Una **computazione** di w è una sequenza di passi di computazione $c_1 \dots c_m$ di w tale che
    - esiste un $q \in F$ con $c_1 = (\epsilon q_0 w) \vdash^*_{\mathcal{A}} (w_m q \epsilon) = c_m$ e
    - per ogni j, $1 < j \leq m$, si ha che $c_{j-1} \vdash_{\mathcal{A}} c_j$.
- w è **accettata** da $\mathcal{A}$ se esiste una computazione di w in $\mathcal{A}$.
- Il **linguaggio accettato** da $\mathcal{A}$ è l'insieme di stringhe accettate da $\mathcal{A}$:

$$\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^* \mid \exists q \in F \text{ tale che } \epsilon q_0 w \vdash^*_{\mathcal{A}} wq\epsilon\}$$