AUTOMI

Un automa è un dispositivo in grado di generare un linguaggio secondo regole predefinite.

DEFINIZIONE - Un automa a stati finiti è una quintupla $(\mathcal{E}, \mathcal{X}, f, x_0, \mathcal{F})$ dove:

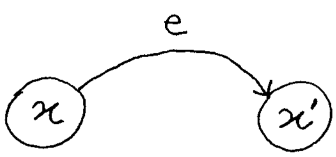
- $\mathcal{E}$ è un alfabeto finito
- $\mathcal{X}$ è l'insieme degli stati (finito)
- f è la funzione di transizione dello stato: $f: \mathcal{X} \times \mathcal{E} \longrightarrow \mathcal{X}$
- $x_0 \in \mathcal{X}$ è lo stato iniziale
- $\mathcal{F} \subseteq \mathcal{X}$ è l'insieme degli stati finali

rappresentazione grafica di automi a stati finiti

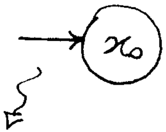
Un automa a stati finiti può essere rappresentato da un grafo orientato dove:

- I vertici rappresentano gli stati
- Gli archi rappresentano le transizioni

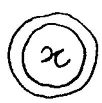
$$f(x, e) = x' \iff$$



NOTA - Gli archi sono etichettati con gli eventi che determinano le transizioni.

- Lo stato iniziale è indicato dal simbolo 

arco senza etichetta e senza nodo di partenza

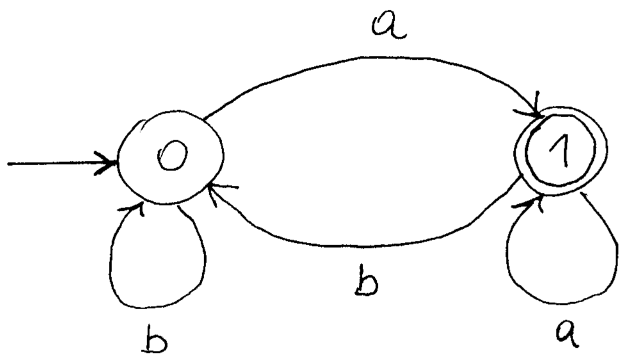
- Gli stati finali sono indicati dal simbolo 

Esempio

$(\mathcal{E}, \mathcal{X}, f, x_0, \mathcal{F})$ con

- $\mathcal{E} = \{a, b\}$ • $f(0,a) = 1$ $f(1,a) = 1$
- $\mathcal{X} = \{0, 1\}$ $f(0,b) = 0$ $f(1,b) = 0$
- $x_0 = 0$ • $\mathcal{F} = \{1\}$

grafo corrispondente:



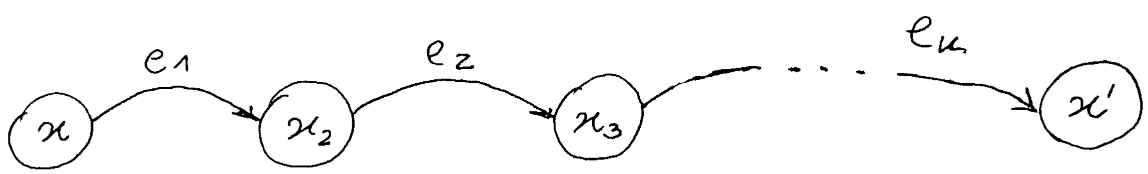
PROBLEMA: definire le parole accettate da un dato automa a stati finiti.

STRINGHE E LINGUAGGI RICONOSCIBILI

Estendiamo la definizione di f a sequenze di eventi:

$f^*(x, u) = x'$ con $u = e_1 e_2 \dots e_k$ sequenza di eventi in $\mathcal{E}$

$\Longleftrightarrow$ $f(x_i, e_i) = x_{i+1}$, $i = 1, \dots, k$ con $x_1 = x$ e $x_{k+1} = x'$.



NOTA - $f^*(x, e_1 e_2) = f(f(x, e_1), e_2)$, ecc.

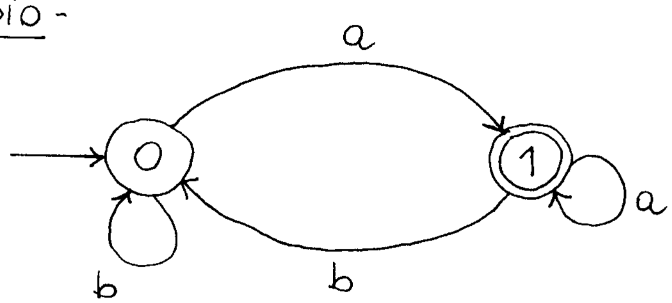
<u>Definizione</u> - Una stringa $u \in \mathcal{E}^*$ si dice <u>riconoscibile</u> dall'automa a stati finiti se

$$f^*(x_0, u) \in \mathcal{F}$$

<u>Definizione</u> - Il <u>linguaggio riconoscibile</u> dall'automa a stati finiti è l'insieme delle stringhe riconoscibili, cioè

$$\mathcal{L} = \{ u \in \mathcal{E}^* : f^*(x_0, u) \in \mathcal{F} \}$$

<u>esempio</u> -



L'automa riconosce tutte le sequenze di simboli dall'alfabeto $\mathcal{E} = \{a, b\}$ che terminano con "a".

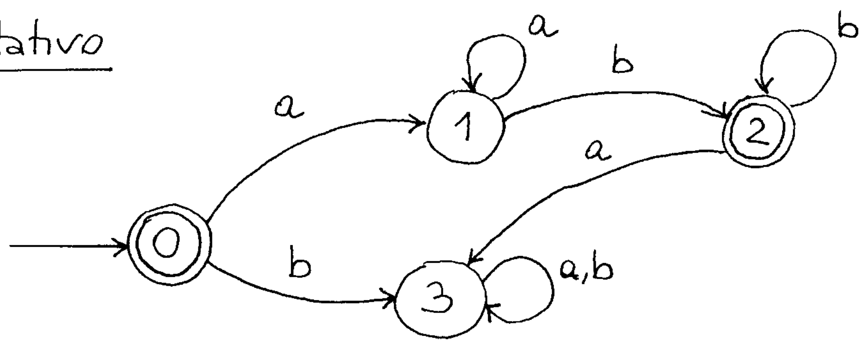
Il linguaggio riconoscibile dall'automa è dunque descritto dall'espressione regolare $(a+b)^* a$ $\Rightarrow$ è un <u>linguaggio regolare</u>.

<u>esempio</u> - Consideriamo il linguaggio

$$\mathcal{L} = \{ u : u = a^n b^n, \ n = 0, 1, 2, \ldots \} = \{ \varepsilon, ab, aabb, aaabbb, \ldots \}$$

Cerchiamo un automa a stati finiti che riconosca $\mathcal{L}$ ...

<u>1° tentativo</u>



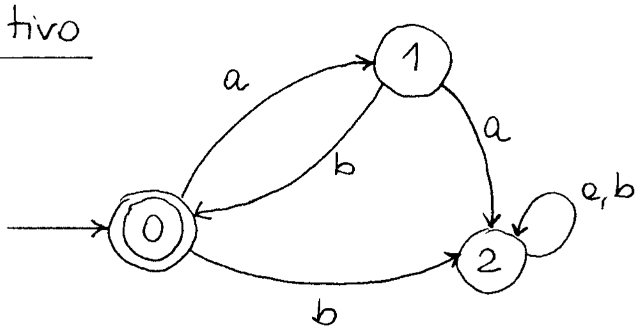
Non va bene! L'automa che abbiamo costruito riconosce il linguaggio $\mathcal{L}'$ descritto da $\varepsilon + aa^*bb^* = \{\varepsilon, ab, aab, aabb, \dots\}$

parola non appartenente a $\mathcal{L}$!

Si osservi che $\mathcal{L}' \supset \mathcal{L}$.

inclusione stretta.

2° tentativo



Non va bene! L'automa che abbiamo costruito riconosce il linguaggio $\mathcal{L}''$ descritto da $(ab)^* = \{\varepsilon, ab, abab, ababab, \dots\}$

parole non appartenenti a $\mathcal{L}$!

Si osservi che $\mathcal{L} \cap \mathcal{L}'' = \{\varepsilon, ab\}$.

Il motivo dei nostri tentativi falliti è che, con un automa a stati finiti, non possiamo <u>contare</u> un numero <u>arbitrario</u> di occorrenze del simbolo "a" per poi ripetere lo stesso numero di occorrenze per il simbolo "b".
Inoltre, si osservi che il linguaggio $\mathcal{L}$ non è esprimibile attraverso una espressione regolare.

Vale il seguente fondamentale teorema, che mette in relazione biunivoca i linguaggi regolari e gli automi a stati finiti:

Teorema di Kleene

Se un linguaggio è regolare, allora può essere generato da un automa a stati finiti; viceversa, il linguaggio riconoscibile da un automa a stati finiti è un linguaggio regolare.

linguaggi regolari $\longleftrightarrow$ automi a stati finiti

---

- In un SED, una sequenza di eventi corrisponde a una specifica evoluzione del sistema.

- Siamo interessati al progetto di SED che implementino i comportamenti (insiemi di evoluzioni) desiderati.

Si pensi a un comportamento anche come a una logica di funzionamento.

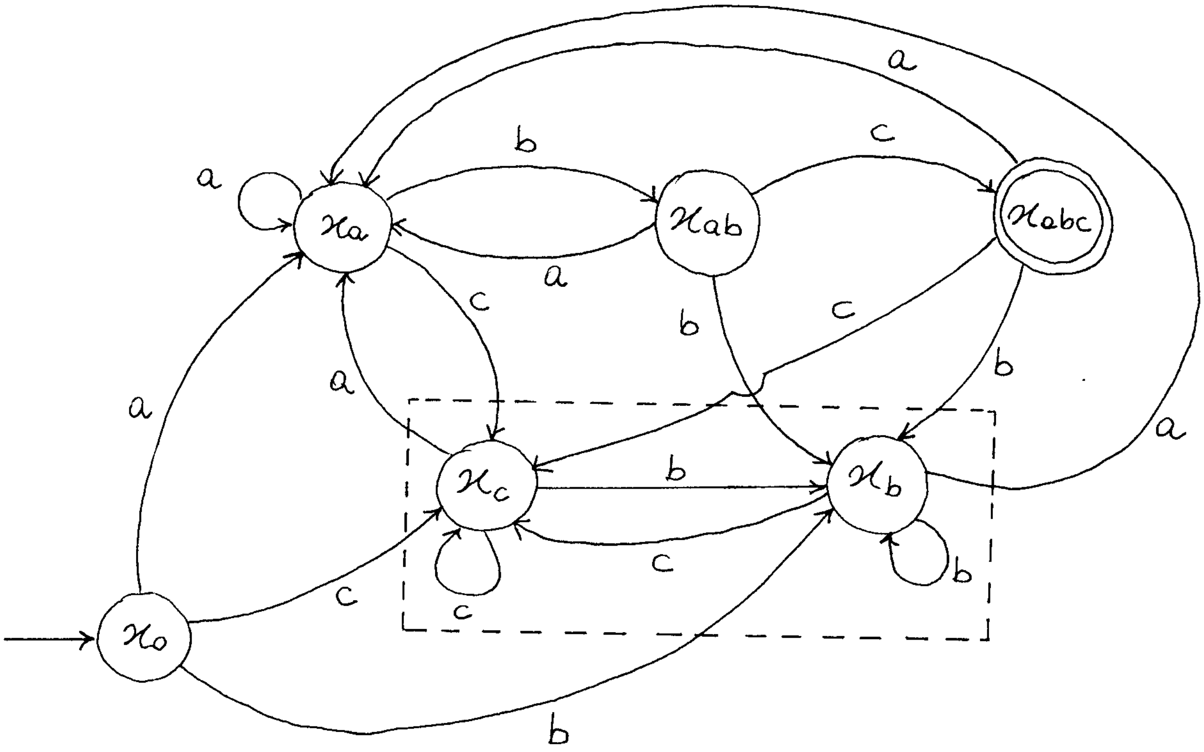
LINGUAGGI $\longleftrightarrow$ COMPORTAMENTI DI SED NON TEMPORIZZATI

esempio – Consideriamo l'esempio della macchina che abbiamo visto nella lezione del 17 ottobre 2007. I comportamenti di tale SED sono descritti dall'espressione regolare $\left(\alpha\left(\beta+\gamma\delta\right)\right)^*$.

- Nel costruire un automa in base a definizioni informali di SED, tipicamente si introduce un numero eccessivo di stati $\Rightarrow$ alcuni stati sono ridondanti.

Esempio - Progettare un automa che, dato l'alfabeto $\{a, b, c\}$, riconosca tutte le sequenze terminanti con "abc".

Un possibile automa che svolge il compito richiesto è il seguente:



Notare l'interpretazione logica degli stati:

- $x_a \equiv$ l'ultimo simbolo osservato è "a"
- $x_{ab} \equiv$ gli ultimi due simboli osservati sono "ab"
- $x_{abc} \equiv$ gli ultimi tre simboli osservati sono "abc" (stato finale)
- $x_b \equiv$ l'ultimo simbolo osservato è "b" non preceduto da "a".
- $x_c \equiv$ l'ultimo simbolo osservato è "c" non preceduto da "ab"

PROBLEMA - Costruire un automa con numero minimo di stati.

↓

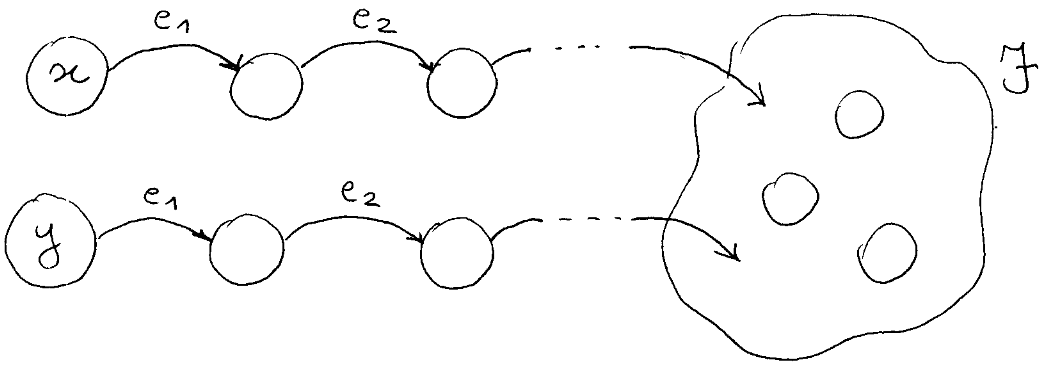
semplicità del sistema progettato.

Aggregazione degli stati (stati equivalenti)

Sia $(\mathcal{E}, \mathcal{X}, f, x_0, \mathcal{F})$ un automa a stati finiti, e sia $R \subseteq \mathcal{X}$.

L'insieme R si dice costituito da <u>stati equivalenti rispetto a $\mathcal{F}$</u> se, per ogni coppia $\{x, y\}$ in R, ed ogni stringa $u \in \mathcal{E}^*$, vale la relazione

$$f^*(x,u) \in \mathcal{F} \iff f^*(y,u) \in \mathcal{F}$$



OSSERVAZIONI

- Due stati x, y con $x \in \mathcal{F}$ e $y \notin \mathcal{F}$ <u>non sono</u> mai equivalenti rispetto a $\mathcal{F}$.

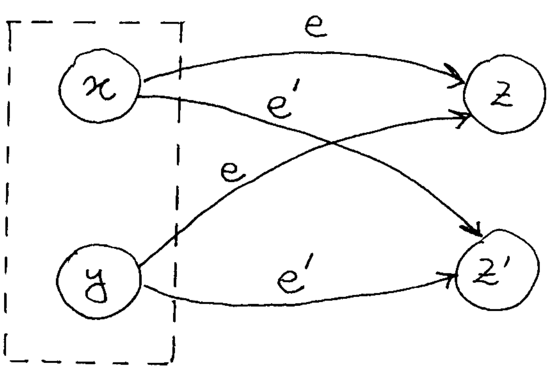
  Infatti, basta applicare $u = \varepsilon$:

  $$f^*(x,\varepsilon) = x \in \mathcal{F}$$

  $$f^*(y,\varepsilon) = y \notin \mathcal{F}$$

- Due stati x, y con $x, y \in \mathcal{F}$, oppure $x, y \notin \mathcal{F}$, sono equivalenti rispetto a $\mathcal{F}$ <u>se</u> (CONDIZIONE SUFFICIENTE, MA <u>NON</u> NECESSARIA)

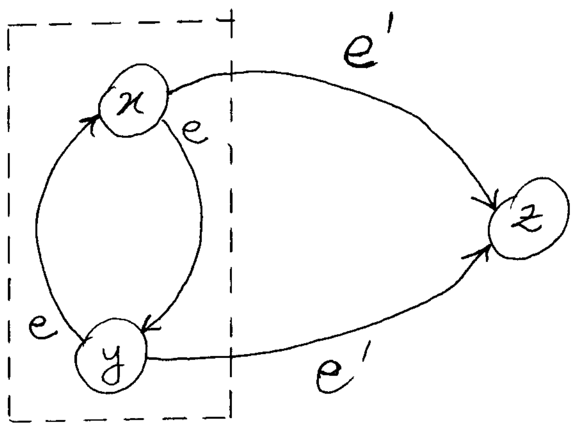
  $$f(x,e) = f(y,e) \quad \forall e \in \mathcal{E}.$$



- Due stati x, y con $x, y \in \mathcal{F}$, oppure $x, y \notin \mathcal{F}$, sono equivalenti rispetto a $\mathcal{F}$ <u>se</u> (CONDIZIONE SUFFICIENTE, MA <u>NON</u> NECESSARIA)

i) $f(x,e)=y$ e $f(y,e)=x$ per qualche $e \in \mathcal{E}$

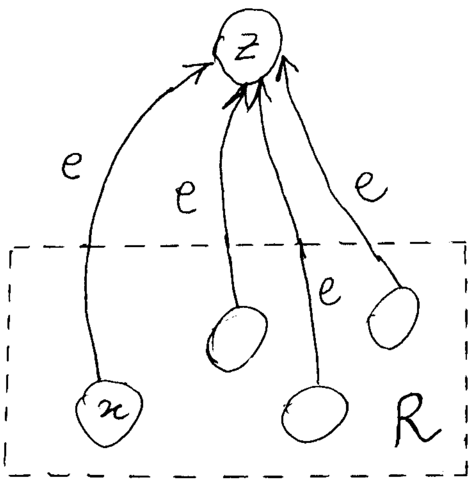
ii) $f(x,e')=f(y,e')$ per tutti gli altri $e' \in \mathcal{E}$.



- Un insieme R tale che $R \subseteq \mathcal{F}$, oppure $R \cap \mathcal{F} = \emptyset$, è composto da stati equivalenti rispetto a $\mathcal{F}$ <u>se</u>

(CONDIZIONE SUFFICIENTE, MA <u>NON</u> NECESSARIA)

$$f(x,e) = z \notin R \overset{\text{Implica}}{\Longrightarrow} f(y,e) = z \quad \forall y \in R$$



esempio - Ritorniamo all'esempio precedente dell'automa che riconosce tutte le sequenze terminanti con "abc".

Consideriamo $R = \{x_b, x_c\}$ (insieme di stati racchiuso nel rettangolo tratteggiato). Osserviamo che:

$f(x_b, a) = x_a = f(x_c, a)$ con $x_a \notin R$

$f(x_b, b) = x_b = f(x_c, b)$ con $x_b \in R$

$f(x_b, c) = x_c = f(x_c, c)$ con $x_c \in R$

Quindi, R è costituito da stati equivalenti rispetto a $\mathcal{F}$, e gli stati in R possono essere fusi in un unico stato.

automa ridotto