FILTRO ~~DEI~~ DEGLI SBALZI

- chiarimento sul funzionamento del filtro

$\ldots 000010\boxed{10}1000100 \ldots$ $\Downarrow$ $\ldots ?00000000000000 \ldots$

$\ldots 1111010\boxed{10}1101011 \ldots$ $\Downarrow$ $\ldots ?11111111111111$

- problema con la definizione di stato data ieri:

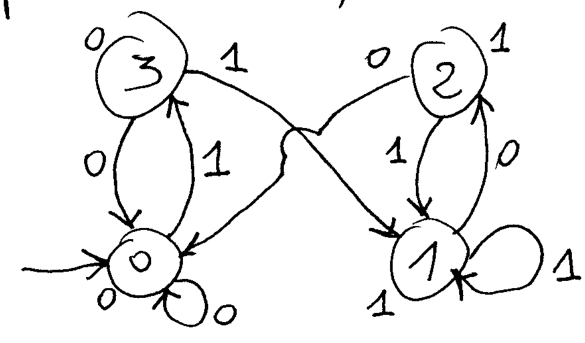
STATO 2: l'ultimo bit è stato 0 e il penultimo è stato 1

⇓

manca l'informazione sul tipo di sottosequenza "attesa" dal filtro

DEFINIZIONE CORRETTA

- sottosequenza che comincia con due 0 consecutivi e non contiene due 1 consecutivi
  * l'ultimo bit ricevuto è 0 => nessun problema => STATO 0
  * l'ultimo bit ricevuto è 1 => ATTENZIONE: se il prossimo bit è 1, si deve passare a una sottosequenza attesa di 1. => STATO 3
- sottosequenza che comincia con due 1 consecutivi e non contiene due 0 consecutivi
  * l'ultimo bit ricevuto è 1 => nessun problema => STATO 1
  * l'ultimo bit ricevuto è 0 => ATTENZIONE: se il prossimo bit è 0, si deve passare a una sottosequenza attesa di 0 => STATO 2



FINO QUI

SED

- $\mathcal{E}$ eventi
- $\mathcal{X}$ stati
- $f(\cdot,\cdot)$ funzione di transizione

$$\{e_1, e_2, e_3, \dots\} \longrightarrow \boxed{SED} \longrightarrow \{x_0, x_1, x_2, x_3, \dots\}$$

- ci sono sequenze di eventi che non sono ammissibili per il ~~sistema~~ SED dal punto di vista fisico

  esempio del carrello

  oppure dal punto di vista logico

  esempio della macchina che esegue operazioni

- possiamo interpretare il SED come un "accettatore" di sequenze di eventi.
- una sequenza di eventi accettata dal SED corrisponde a un'evoluzione del sistema
- l'insieme di tutte le evoluzioni del SED ne costituisce il comportamento (DINAMICA)

Due problemi:

- esistono dei formalismi che ci permettono di rappresentare in maniera compatta il comportamento di un SED? ⟹ LINGUAGGI
- esistono dei modelli di "accettatori di sequenze" che possiamo utilizzare come modelli del SED?
  ⟹ AUTOMI A STATI FINITI

LINGUAGGI

alfabeto: insieme non vuoto di simboli, $\mathcal{E}$

stringhe (parole): sequenze di simboli appartenenti ad $\mathcal{E}$

linguaggio: insieme di stringhe costruite con i simboli appartenenti all'alfabeto $\mathcal{E}$.

In un SED:

- l'insieme degli eventi rappresenta l'alfabeto
- ciascuna evoluzione del SED è una stringa del linguaggio
- il comportamento del SED è il linguaggio

comportamenti dei SED ⟺ linguaggi

Operazioni sui linguaggi

notazione: la parola vuota (sequenza di lunghezza zero) viene indicata con ε.

~~un alfabeto Σ e due ling~~

~~Dato due alfabeti~~ A e B

Dato un alfabeto Σ e due linguaggi A e B definiti su Σ:

1. concatenazione

$$AB = \{ w = uv \; : \; u \in A, \; v \in B \}$$

2. unione e intersezione

i) $A \cup B = \{ w : w \in A \text{ oppure } w \in B \}$

ii) $A \cap B = \{ w : w \in A \text{ e } w \in B \}$

3. chiusura di Kleene

unione di tutte le possibili concatenazioni dello stesso linguaggio, inclusa la stringa vuota.

$$A^* = \{\varepsilon\} \cup A \cup A^2 \cup A^3 \cup \ldots = \bigcup_{i=0}^{\infty} A^i$$

dove $A^i = \begin{cases} \{\varepsilon\} & \text{se } i = 0 \\ A A^{i-1} & \text{se } i = 1, 2, 3, \ldots \end{cases}$

<u>esempio</u> - $A = \{a, b, bba\}$

$$A^* = \{\varepsilon, \underbrace{a, b, bba}_{A}, \underbrace{aa, ab, abba, ba, bb, bbba, bbaa, bbab, bbabba}_{A^2}, \dots\}$$

<u>osservazione</u>: In questo esempio, il linguaggio A ha cardinalità finita, mentre A^* ha cardinalità infinita.

$\Downarrow$

La chiusura di Kleene ci permette di rappresentare in maniera compatta un linguaggio di cardinalità infinita.

LINGUAGGI REGOLARI

Un'<u>espressione regolare</u> (ER) ~~su~~ su un alfabeto Σ si costruisce a partire dalle seguenti regole:

<u>espressioni regolari atomiche</u>

- $\emptyset$ è un' ~~insieme~~ ER. (insieme vuoto)
- ε è un'ER
- se $e \in \Sigma$, e ~~un~~ è un'ER.

(un simbolo è un'ER)

operatori su espressioni regolari

Se u e v sono ER:

- $u+v$ è un'ER (unione)
- uv è un'ER (concatenazione)
- u^* è un'ER (chiusura di Kleene).

⇒ Le ER ci permettono di rappresentare in maniera compatta linguaggi (⟺ comportamenti di SED) composti da infinite parole/stringhe.

Regole di precedenza tra gli operatori

- la chiusura di Kleene è l'operatore con maggiore priorità, seguito dalla concatenazione, e infine dall'unione.

Per cambiare la precedenza tra gli operatori, si utilizzano le parentesi.

esempio

$$\underbrace{ab^*}+c \neq (ab)^*+c$$

$\{c, a, ab, abb, abbb, \dots\}$

$a\{\varepsilon, b, bb, bbb, bbbb, \dots\}$

$a, ab, abb \dots$

$\{c, \varepsilon, ab, abab, ababab, \dots\}$

Def: Un LINGUAGGIO REGOLARE è un linguaggio descritto da un'espressione regolare.

OSSERVAZIONE: Un linguaggio costituito da un numero finito di parole è un linguaggio regolare.
$\Rightarrow$ il linguaggio può essere descritto dall'ER costruita facendo la "somma" (unione) delle singole parole.

esempio - $\mathcal{L} = \{a, b, aba, bbb\}$

$ER = a+b+aba+bbb$

PROBLEMA: DATO UN SED, come determinare un'ER (se esiste) che ne descrive il comportamento.

AUTOMI A STATI FINITI

Def.: Un automa (accettatore) 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
  $f: \mathcal{X} \times \mathcal{E} \longrightarrow \mathcal{X}$

- $x_0 \in \mathcal{X}$ stato iniziale
- $\mathcal{F} \subseteq \mathcal{X}$ è l'insieme degli stati finali

PROBLEMA: come determinare le sequenze di simboli appartenenti ad $\mathcal{E}$ accettate dall'automa?

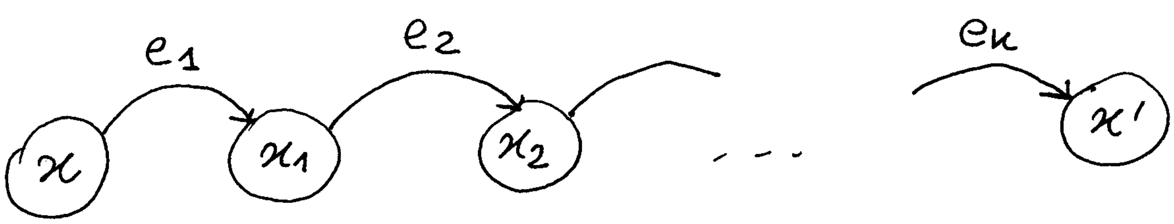
STRINGHE E LINGUAGGI RICONOSCIBILI

Consideriamo una funzione f^* che estende la funzione di transizione f dello stato a sequenze di simboli/eventi:

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

$\Longleftrightarrow \quad f(x_i, e_{i+1}) = x_{i+1}, \quad i = 0, 1, \dots, k-1$

con $x_0 = x$ e $x_k = x'$.

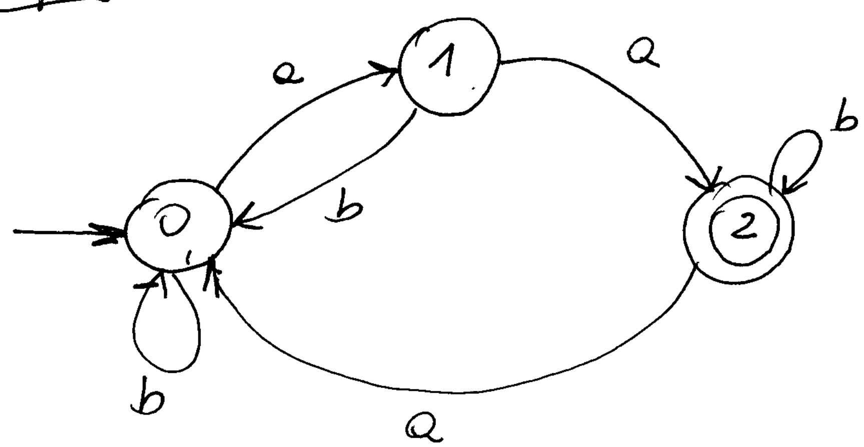


⇓ cammino sul grafo

Def: Una stringa $u \in \mathcal{E}^*$ [$\mathcal{E}^*$ rappresenta l'insieme di tutte le parole costruibili ~~stati~~ con l'alfabeto $\mathcal{E}$] si dice riconoscibile/accettata dall'automa a stati finiti se

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

Esempio - $\mathcal{E} = \{a, b\}$



aabba $\notin$ stringhe accettate dall'automa

~~abaab~~

$$f^*(0, aabba) = 0 \notin \mathcal{F}$$

abaab $\in$ stringhe accettate dall'automa

$$f^*(0, abaab) = 2 \in \mathcal{F}.$$