Trasparenze del corso di

Informatica Teorica

Parte Quarta

Proprietà dei Linguaggi Regolari

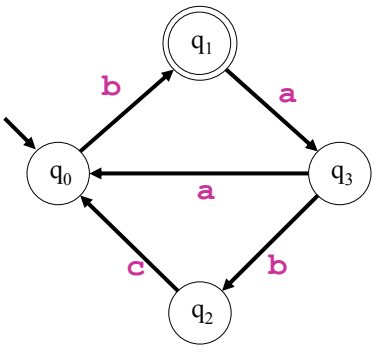
1

Proprietà dei linguaggi regolari

- pumping lemma
- chiusura rispetto ad operazioni insiemistiche
  - unione, complementazione, intersezione concatenazione, stella
- rapporti con espressioni regolari
- decidibilità e linguaggi regolari
- teorema di Myhill-Nerode

2

Considerazioni sul numero degli stati e sulle ripetizioni nelle stringhe del linguaggio



questo automa riconosce, tra le altre, le stringhe:

b,
b(aab),
b(abcb),
b(aab)(aab),
b(aab)(abcb),
b(abcb)(aab),
b(abcb)(abcb),
b(aab)(aab)(aab), …

Quando la stringa ha più caratteri del numero degli stati, contiene una parte che si può eliminare o ripetere a piacimento.

E' questa una condizione generale?

3

Pumping lemma per i linguaggi regolari

risultato fondamentale sui linguaggi regolari

lemma:

per ogni linguaggio regolare L esiste una costante n tale che se $z \in L$ e $|z| \geq n$ allora $z=uvw$ con $|uv| \leq n$, $|v| \geq 1$ e $uv^iw \in L$, per $i=0,1,\ldots$

dimostrazione:

basata sui cicli (eventuali) del diagramma degli stati

4

Dimostrazione (pumping lemma, regolari)

- consideriamo l'ASF $A=<\Sigma,K,\delta,q_0,F>$ che riconosce L con il minimo numero $n=|K|$ di stati
- supponiamo esista $z \in L$ con $|z| \geq n$ (altrimenti la dimostrazione è banale)
- abbiamo che $\underline{\delta}(q_0,z) \in F$
- sia $q_{i0},q_{i1},\ldots,q_{i|z|}$ la sequenza di stati attraversati da A durante il riconoscimento di z, $q_{i0}=q_0$ e $q_{i|z|} \in F$
- dato che $|z| \geq n$, allora esiste almeno uno stato attraversato 2 volte
- sia j il primo valore per cui q_{ij} è attraversato 2 volte, sia u il più breve prefisso di z tale che $\underline{\delta}(q_0,u)=q_{ij}$ e sia $z=ux$

5

Dimostrazione (continua)

- abbiamo che $|u|<n$ e che $\underline{\delta}(q_{ij},x)=q_{i|z|} \in F$
- sia v il più breve prefisso (non vuoto) di x tale che $\underline{\delta}(q_{ij},v)=q_{ij}$ e sia $x=vw$
- abbiamo che $|uv| \leq n$ e $\underline{\delta}(q_{ij},w)=q_{i|z|} \in F$
- inoltre $\underline{\delta}(q_0,uv)= \underline{\delta}(\underline{\delta}(q_0,u),v)= \underline{\delta}(q_{ij},v)=q_{ij}$
- $\underline{\delta}(q_0,uv^iw) = \underline{\delta}(\underline{\delta}(q_0,u),v^iw)$
  $= \underline{\delta}(q_{ij},v^iw)$
  $= \underline{\delta}(\underline{\delta}(q_{ij},v),v^{i-1}w)$
  $= \underline{\delta}(q_{ij},v^{i-1}w)$
  $= \ldots$
  $= \underline{\delta}(q_{ij},w)$
  $= q_{i|z|} \in F$

6

Considerazioni sul pumping lemma

- il pumping lemma mostra che gli ASF non hanno la capacità di memorizzare un numero infinito di situazioni, "non sanno contare"
- il pumping lemma offre l'opportunità di una verifica ragionevolmente rapida di non regolarità per un linguaggio dato: se non vale il pumping lemma $\Rightarrow$ il linguaggio non è regolare
- se vale il pumping lemma nulla si può affermare sulla regolarità del linguaggio

7

Esempio: il linguaggio a^nb^n

- il linguaggio $L=\{a^nb^n|n\geq 0\}$ non è regolare
- infatti, supponiamo che valga il pumping lemma
- dovrebbe esistere un indice m tale che in tutte le stringhe z del linguaggio di lunghezza maggiore di m è possibile individuare una sottostringa v, con $z=uvw$ e $|uv|\leq m$, che è possibile eclissare o ripetere indefinitamente generando stringhe sempre appartenenti al linguaggio
- se però prendiamo la stringa $a^{2m}b^{2m}$ vediamo che questo non è vero

8

Proprietà di chiusura dei linguaggi regolari

dimostreremo nel seguito che i linguaggi regolari sono chiusi rispetto alle operazioni di:

- unione,
- complementazione,
- intersezione,
- concatenazione e
- stella

9

Unione di linguaggi regolari

teorema:

l'unione di due linguaggi regolari è regolare

dimostrazione:

dati due automi non deterministici

$A_1=<\Sigma_1,K_1,\delta_{N1},q_{01},F_1>$

$A_2=<\Sigma_2,K_2,\delta_{N2},q_{02},F_2>$

che riconoscono i linguaggi L_1 ed L_2

costruiamo un automa $A=<\Sigma,K,\delta_N,q_0,F>$ che riconosca il linguaggio $L_1 \cup L_2$

10

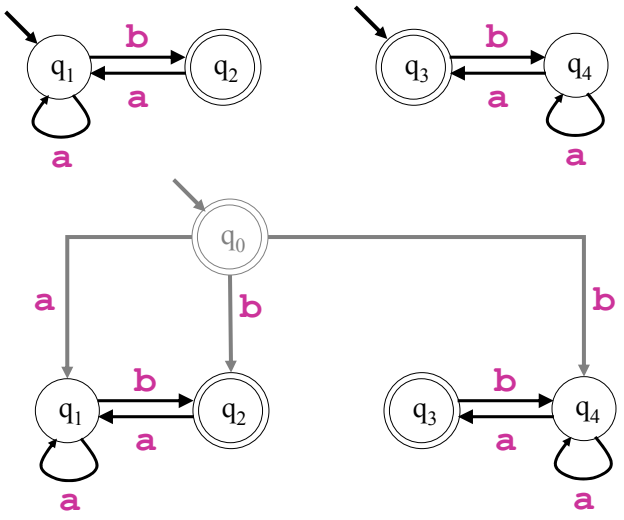
Dimostrazione (unione)

- $\Sigma=\Sigma_1\cup\Sigma_2$
- $K=K_1\cup K_2\cup\{q_0\}$
- q_0 nuovo stato iniziale
- se $\varepsilon\in L_1$ oppure $\varepsilon\in L_2 \Rightarrow F=F_1\cup F_2\cup\{q_0\}$
  altrimenti $F=F_1\cup F_2$
- $\delta_N(q,\mathtt{a}) = \delta_{N1}(q,\mathtt{a})$ se $q\in K_1$ $\mathtt{a}\in\Sigma_1$
  $\delta_N(q,\mathtt{a}) = \delta_{N2}(q,\mathtt{a})$ se $q\in K_2$ $\mathtt{a}\in\Sigma_2$
  $\delta_N(q_0,\mathtt{a}) = \delta_{N1}(q_{01},\mathtt{a}) \cup \delta_{N2}(q_{02},\mathtt{a})$ $\mathtt{a}\in\Sigma_1\cup\Sigma_2$

osservazione: l'automa risultante potrebbe non essere deterministico anche se lo sono A_1 e A_2

11

Esempio (unione)



12

Complementazione di un linguaggio regolare

teorema:

il complementare di un linguaggio regolare è regolare

dimostrazione:

a partire da un ASF

$A=<\Sigma,K,\delta,q_0,F>$

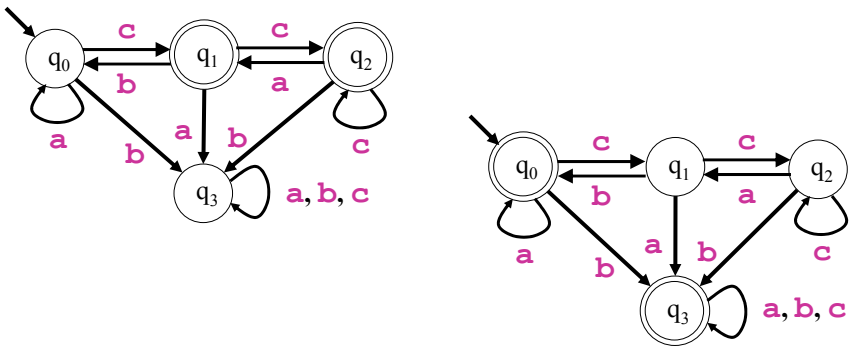
che riconosce L, costruiamo l'ASF A′ che riconosce il linguaggio complementare

13

Dimostrazione (complemento)

$A'=<\Sigma,K,\delta,q_0,F'=K-F>$

Ogni stringa che porta A in uno stato finale porta A′ in uno stato non finale, per cui $L(A')=\Sigma^*-L(A)$



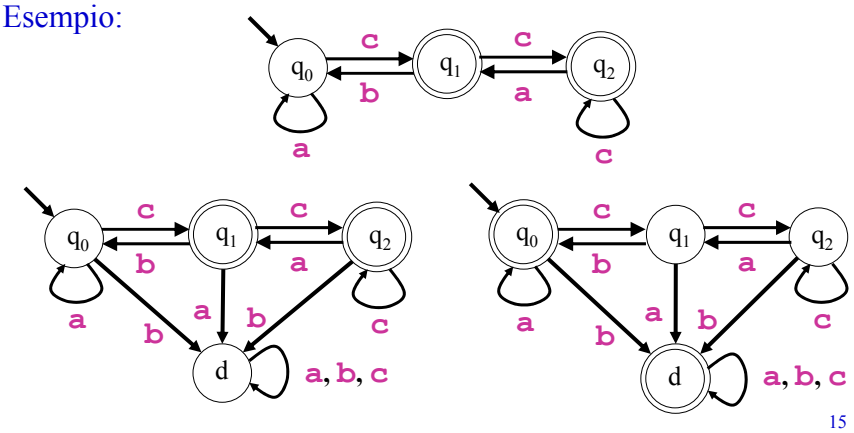
14

Osservazioni (complemento)

Prima di eseguire la trasformazione dell'automa A in A′, occorre verificare che la funzione δ sia totale

Eventualmente, se lo stato pozzo è taciuto dal A, occorre esplicitarlo

Esempio:



15

Intersezione di linguaggi regolari

teorema:

l'intersezione di due linguaggi regolari è regolare

dimostrazione:

banale usando le leggi di De Morgan

16

Concatenazione di linguaggi regolari

teorema:

la concatenazione di due linguaggi regolari è regolare

dimostrazione:

a partire dagli automi

$A_1=<\Sigma_1,K_1,\delta_1,q_{01}, F_1>$

$A_2=<\Sigma_2,K_2,\delta_2,q_{02}, F_2>$

che riconoscono i linguaggi $L(A_1)$ e $L(A_2)$, costruiamo un ASFND che riconosce $L(A_1) \circ L(A_2)$

17

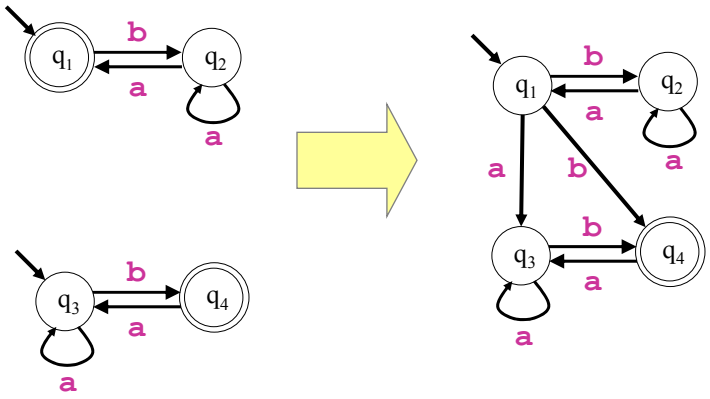
Dimostrazione (concatenazione)

l'ASFND $A=<\Sigma,K,\delta,q_0,F>$ è definito come segue

- $\Sigma=\Sigma_1\cup\Sigma_2$
- $K=K_1\cup K_2$
- se $\varepsilon\in L(A_2)$ allora $F=F_1\cup F_2$ altrimenti $F=F_2$
- $q_0=q_{01}$
- $\delta_N(q,a) = \delta_1(q,a)$ se $q\in K_1-F_1$ e $a\in\Sigma_1$
  $\delta_N(q,a) = \delta_1(q,a)\cup \delta_2(q_{02},a)$ se $q\in F_1$ e $a\in\Sigma$
  (gli stati finali di A_1 hanno anche le transizioni dello stato iniziale di A_2)
  $\delta_N(q,a) = \delta_2(q,a)$ se $q\in K_2$ e $a\in\Sigma_2$

18

Esempio (concatenazione)



19

Chiusura transitiva di linguaggi regolari

teorema:

se L è un linguaggio regolare, L^* è regolare

dimostrazione:

a partire dall'automa

$A=<\Sigma,K,\delta,q_0,F>$

che riconosce il linguaggio L(A), costruiamo l'automa

$A'=<\Sigma',K',\delta',q'_0,F'>$

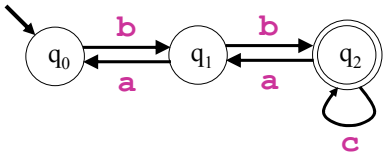
che riconosce il linguaggio $L^*=(L(A))^*$

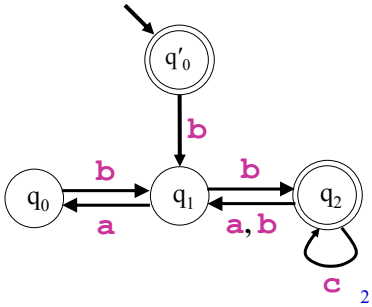
20

Dimostrazione (iterazione)

L'ASF $A'=<\Sigma',K',\delta',q'_0,F'>$ è definito come segue:

- $\Sigma'=\Sigma$
- $K'=K\cup\{q'_0\}$
- $\delta'(q,a) = \delta(q,a)$ se $q\in K-F$
  $\delta'(q,a) = \delta(q,a)\cup\delta(q_0,a)$ se $q\in F$
  $\delta'(q'_0,a) = \delta(q_0,a)$
- $F'=F\cup\{q'_0\}$





21

Proprietà di chiusura dei linguaggi regolari

teorema:

la classe dei linguaggi regolari su Σ è la più piccola contenente Λ e $\{\sigma_i\}$, con $\sigma_i\in\Sigma$, che sia chiusa rispetto alle operazioni di unione, concatenazione e stella

22

Espressioni e grammatiche regolari

teorema:

un linguaggio definito da un'espressione regolare è regolare

dimostrazione:

dimostriamo che ogni espressione regolare definisce un linguaggio regolare:

- ciò è facile per il linguaggi Λ e $\{\sigma_i\}$, con $\sigma_i \in \Sigma$
- per i linguaggi definiti utilizzando gli operatori $+$, $\circ$, e *, possiamo applicare le proprietà di chiusura dei linguaggi regolari rispetto a unione, concatenazione e stella ripetuti un numero finito di volte

23

Espressioni e grammatiche regolari

teorema:

un linguaggio regolare è definibile con una espressione regolare

dimostrazione: 

dato un linguaggio regolare L costruiamo l'espressione regolare corrispondente risovendo un sistema di equazioni

supponiamo che L non contenga ε

altrimenti si aggiunge "$+\varepsilon$" all'espressione regolare ottenuta con lo stesso procedimento sul linguaggio $L-\{\varepsilon\}$

estendiamo il linguaggio delle espressioni regolari ammettendo la presenza di variabili

esempio: la presenza della variabile A in una espressione regolare indica che A è una espressione regolare non nota

24

Dimostrazione (impostazione del sistema)

raggruppiamo le produzioni che hanno lo stesso non terminale a sinistra

ad ogni produzione

$A \rightarrow a_1B_1|a_2B_2|\ldots|a_nB_n|b_1|b_2|\ldots|b_m$

associamo l'equazione

$A = a_1B_1+a_2B_2+\ldots+a_nB_n+b_1+b_2+\ldots+b_m$

il sistema ottenuto è composto da equazioni lineari destre, con monomi dalla struttura semplice

esempio: alla grammatica

$A \rightarrow aA|bB$

$B \rightarrow bB|c$

corrisponde il sistema

$A = aA+bB$

$B = bB+c$

25

Dimostrazione (risoluzione del sistema)

tecnica di sostituzione

esempio da

$A = aA+bB$

$B = bB+c$

otteniamo

$A = aA+b(bB+c)$

$B = bB+c$

26

Dimostrazione (risoluzione del sistema)

tecnica di eliminazione della ricorsione

esempio da

$B = bB+c$

otteniamo

$B = b^*c$

infatti sostituendo $B=b^*c$ in $B=bB+c$ otteniamo:

$b^*c = b(b^*c)+c=b^+c+c=(b^++\varepsilon)c = b^*c$

più in generale

$A = \alpha_1A+\alpha_2A+\ldots\alpha_nA+ \beta_1+\beta_2+\ldots+\beta_m$

si risolve con

$A = (\alpha_1+\alpha_2+\ldots\alpha_n)^*(\beta_1+\beta_2+\ldots+\beta_m)$

27

Dimostrazione (conclusioni)

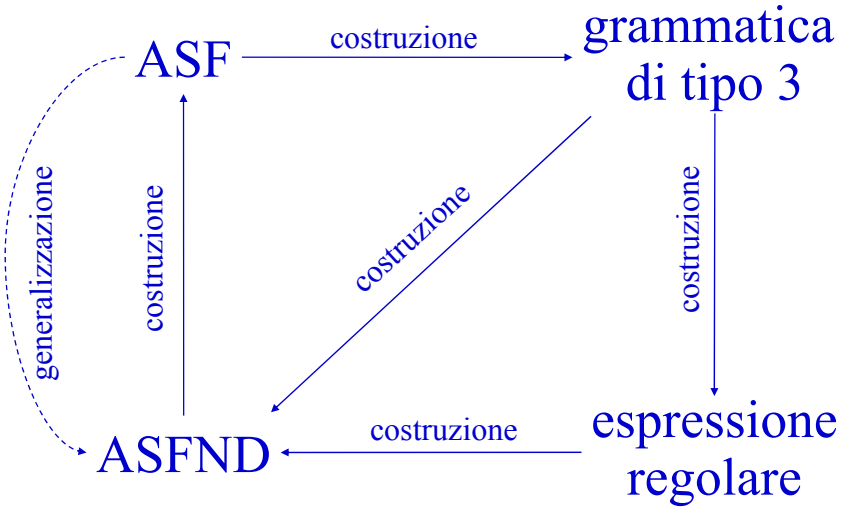
applicando le regole di sostituzione ed eliminazione della ricorsione possiamo risolvere ogni sistema di equazioni lineari destre

quindi possiamo determinare il linguaggio associato alla variabile corrispondente all'assioma

quindi possiamo determinare l'espressione regolare che descrive il linguaggio

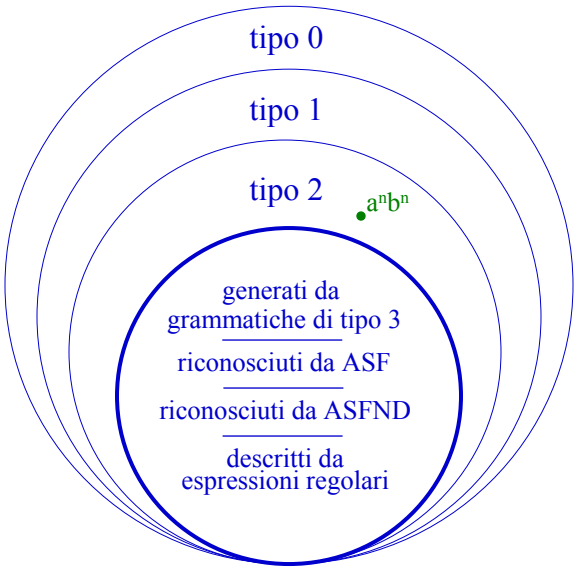
28

Quadro riassuntivo sui linguaggi regolari



29

Quadro riassuntivo sui linguaggi regolari



30

Esempio

trovare l'espressione regolare corrispondente alla grammatica:

$S \rightarrow aS|bA|\varepsilon$
$A \rightarrow aA|bS|\varepsilon$

rimuoviamo le ε-produzioni che non sono sull'assioma:

$S \rightarrow aS|bA|\varepsilon|b$
$A \rightarrow aA|bS|a$

cerchiamo l'espressione regolare corrispondente alla grammatica:

$S \rightarrow aS|bA|b$
$A \rightarrow aA|bS|a$

31

Esempio (continua)

espressione regolare corrispondente:

$A = aA+bS+a = aA+(bS+a)$
$A = a^*(bS+a)$
$S = aS+bA+b = aS+b\,a^*(bS+a)\,+b = aS+ba^*bS+ba^*a+b$
$S = (a+ba^*b)S+(ba^*a+b)$
$S = (a+ba^*b)^*(ba^*a+b)$
$S = (a+ba^*b)^*b(a^*a+\varepsilon) = (a+ba^*b)^*ba^*$

dunque l'espressione regolare corrispondente alla grammatica di partenza è:

$(a+ba^*b)^*ba^*+\varepsilon$

32

Esempio

ricavare l'espressione regolare corrispondente alla grammatica:

S → aA|bB|cA|a|b|c

A → aA|cA|a|c

B → bB|b

impostiamo il sistema:

S = aA + bB + cA + a + b + c

A = aA + cA + a + c

B = bB + b

33

Esempio (soluzione del sistema)

S = aA + bB + cA + a + b + c
A = aA + cA + a + c
B = bB + b = b^*b *(eliminazione della ricorsione)*

S = aA + bB + cA + a + b + c
A = aA + cA + a + c = $(a+c)^*(a+c)$ *(elim della ricorsione)*
B = b^*b

S = $a(a+c)^*(a+c) + bb^*b + c(a+c)^*(a+c) + a + b + c$ *(sost.)*

S = $(a+c)(a+c)^*(a+c) + bb^*b + a + b + c$
S = $(a+c)^*(a+c)(a+c) + (a+c) + bbb^* + b$
S = $(a+c)^*(a+c) + bb^*$

34

Proprietà decidibili dei linguaggi regolari

teorema:

è possibile decidere se il linguaggio regolare L(A) riconosciuto da un ASF A è vuoto, finito o infinito

dimostrazione:

- L(A) è vuoto

  L(A) è vuoto se e solo se non accetta nessuna stringa z con $|z|<n$; infatti, se A riconosce parole con almeno n simboli, per il pumping lemma riconosce anche parole con meno di n simboli

  quindi per verificare se L(A) è vuoto basta verificare che A non riconosca nessuna parola tra tutte quelle di lunghezza < n definibili con Σ

35

Proprietà decidibili dei linguaggi regolari

- L(A) è finito o infinito

  L(A) è infinito se e solo se accetta una stringa z con $n \leq |z| < 2n$; infatti, se A accetta z con $|z| \geq n$ per il pumping lemma accetta infinite stringhe

  inoltre, se A accetta z con $|z| \geq 2n$, per il pumping lemma accetta una stringa x con $n \leq |x| < 2n$

- algoritmo

  input: automa A che riconosce il linguaggio L(A)

  output: determina se L(A) è vuoto, finito o infinito

  - proponi ad A tutte le stringhe di Σ^* di lunghezza minore di 2n
  - se nessuna stringa è accettata allora L è vuoto
  - se tutte le strighe accettate hanno lunghezza < n allora n è finito
  - se tra le strighe accettate ve ne sono alcune con lunghezza ≥ n allora L è infinito

36

Proprietà decidibili dei linguaggi regolari

teorema:

il problema dell'equivalenza di due linguaggi regolari è decidibile

dimostrazione:

basta dimostrare che la loro intersezione coincide con la loro unione

basta quindi dimostrare che l'unione dell'intersezione del primo con il complementare del secondo e dell'intersezione del secondo con il complementare del primo è il linguaggio vuoto

osservazione: la decidibilità dell'equivalenza implica la decidibilità della minimizzazione del numero degli stati

algoritmo: dato un automa con n stati che riconosce L, costruisco tutti gli automi con m<n stati (sull'alfabeto di L) e verifico l'equivalenza con L

37

Teorema di Myhill-Nerode

un linguaggio L definisce implicitamente la seguente relazione binaria di equivalenza su Σ^*

$$x\ R_L\ y \Leftrightarrow (\forall z \in \Sigma^* \quad xz \in L \Leftrightarrow yz \in L)$$

cioè x e y sono nella stessa classe di equivalenza se completate con lo stesso suffisso entrambe appartengono o non appartengono ad L

teorema:

L è regolare se e solo se R_L ha indice finito

dimostrazione:

- se L è regolare dimostriamo che R_L ha indice finito costruendo una relazione di equivalenza più fine di R_L e dimostrando che la nuova relazione ha indice finito
- se R_L ha indice finito dimostriamo che L è regolare costruendo un automa che riconosce L

38

Dimostrazione (L regolare $\Rightarrow$ R_L indice finito)

costruiamo l'ASF $A=<\Sigma,K,\delta,q_0,F>$ che riconosce L e definiamo una nuova relazione binaria R_M su Σ^*

$$x\ R_M\ y \Leftrightarrow \underline{\delta}(q_0,x)=\underline{\delta}(q_0,y)$$

due stringhe sono equivalenti se portano nello stesso stato di K a partire da q_0

- R_M ha indice finito, pari, al più, a $|K|$
- se $x\ R_M\ y$ allora $\forall z\in\Sigma^*$ $xz\ R_M\ yz$, infatti se $\underline{\delta}(q_0,x)=\underline{\delta}(q_0,y)$ allora anche $\underline{\delta}(q_0,xz)=\underline{\delta}(q_0,yz)$
- se $x\ R_M\ y$ allora $x\ R_L\ y$, infatti se $\underline{\delta}(q_0,x)=\underline{\delta}(q_0,y)$ allora anche $\underline{\delta}(q_0,xz)=\underline{\delta}(q_0,yz)$ e quindi $xz\in L\Leftrightarrow yz\in L$
- quindi l'indice di R_L è minore o uguale dell'indice di R_M a sua volta finito

39

Dimostrazione (R_L indice finito $\Rightarrow$ L regolare)

supponiamo che R_L abbia indice finito

definiamo un automa che riconosce L

ad ogni classe [x] associamo lo stato $q_{[x]}$

poniamo:

$q_0 = q_{[\varepsilon]}$

$F=\{q_{[x]} \mid x\in L\}$

$\delta(q_{[x]},\sigma) = q_{[x\sigma]}$

l'automa riconosce L che quindi è regolare

osservazione: l'automa costruito ha il numero minimo di stati, in quanto R_L ha indice $\leq$ di R_M

40