Cognome .... Nome ....Matricola ....

Informatica Teorica I – Informatica Teorica primo modulo
Esame del 22 novembre 2004

Tempo a disposizione: 100 minuti A & B

Regole del gioco: Libri e quaderni chiusi, vietato scambiare informazioni con altri; indicare con chiarezza cognome, nome e numero di matricola; *consegnare solo i fogli con le domande (questi)*.

Esercizio 1 (20%) Determina le espressioni regolari che descrivono i seguenti linguaggi su Σ={**a**,**b**}.
NOTA: una buona strategia di soluzione (forse un po' prolissa) per trovare le espressioni regolari dei complementi dei linguaggi qui sotto è quella di determinare un ASF deterministico che riconosca il linguaggio, complementare gli stati e poi trovare le espressioni regolari di tutte le stringhe che conducono ad uno stato finale.

1.1 Il complemento del linguaggio (**a**+**b**)*.

il linguaggio qui sopra coincide con Σ*. Il suo complemento è il linguaggio nullo Λ. L'espressione regolare è ∅. NOTA: è errato identificare ∅ con ε.

1.2 Il complemento del linguaggio **a***+**b***.

Il linguaggio qui sopra è costituito da stringhe uniformi di sole "a" o di sole "b" (più la stringa nulla ε). Qualsiasi stringa contenga una "a" e una "b" appartiene al complemento. Alcuni esempi di soluzioni corrette:
(a+b)*ab(a+b)* + (a+b)*ba(a+b)*
aa*b(a+b)* + bb*a(a+b)*
(a+b)*(ab+ba)(a+b)*
(a+b)*a(a+b)*b(a+b)* + (a+b)*b(a+b)*a(a+b)*

1.3 Il complemento del linguaggio (**aa**)*.

Il complemento del linguaggio qui sopra è costituito dalle stringhe con un numero dispari di "a" o che contengono almeno una "b". Alcuni esempi di soluzioni corrette:
a(aa)* + a*b(a+b)*
a(aa)* + (a+b)*b(a+b)*

1.4 Il complemento del linguaggio (**ab**)*.

Il complemento del linguaggio qui sopra è composto da: (1) stringhe alternate di "a" e "b" comincianti per "a" ma di lunghezza dispari; (2) stringhe alternate di "a" e "b" comincianti per "b"; (3) stringhe non alternate (cioè con "aa" o con "bb").
Esempi di soluzioni corrette:
a(ba)* + (ab)*b(a+b)* + a(ba)*a(a+b)*
(ab)*a + (ab)*b(a+b)* + (ab)*aa(a+b)*
(ab)*(a + b(a+b)* + aa(a+b)*

Cognome .... Nome ....Matricola ....

Esercizio 2 compito A (20%) Dimostra tramite induzione matematica che per ogni n=1,2,3,... $\sum_{k=1}^{n}\frac{1}{k(k+1)}=\frac{n}{(n+1)}$

passo base (per n=1)

$$\sum_{k=1}^{1}\frac{1}{k(k+1)}=\frac{1}{1(1+1)}=\frac{1}{2}=\frac{1}{(1+1)}$$

passo induttivo:

ipotesi induttiva

$$\sum_{k=1}^{n_0}\frac{1}{k(k+1)}=\frac{n_0}{(n_0+1)}$$

lo devo dimostrare per $n=n_0+1$

$$\sum_{k=1}^{n}\frac{1}{k(k+1)}=\sum_{k=1}^{n_0+1}\frac{1}{k(k+1)}=\frac{1}{(n_0+1)(n_0+2)}+\frac{n_0}{(n_0+1)}=$$

$$=\frac{1+n_0(n_0+2)}{(n_0+1)(n_0+2)}=\frac{n_0^2+2n_0+1}{(n_0+1)(n_0+2)}=\frac{(n_0+1)^2}{(n_0+1)(n_0+2)}=$$

$$=\frac{(n_0+1)}{(n_0+2)}=\frac{n}{(n+1)}$$

Esercizio 2 compito B (20%) Dimostra tramite induzione matematica che per ogni n=1,2,3,... $\sum_{k=1}^{n}k^3=\frac{n^2(n+1)^2}{4}$

passo base (per n=1)

$$\sum_{k=1}^{1}k^3=1^3=1=\frac{1^2(1+1)^2}{4}=\frac{4}{4}=1$$

passo induttivo:

ipotesi induttiva

$$\sum_{k=1}^{n_0}k^3=\frac{n_0^2(n_0+1)^2}{4}$$

lo devo dimostrare per $n=n_0+1$

$$\sum_{k=1}^{n}k^3=\sum_{k=1}^{n_0+1}k^3=(n_0+1)^3+\sum_{k=1}^{n_0}k^3=(n_0+1)^3+\frac{n_0^2(n_0+1)^2}{4}=$$

$$=\frac{4(n_0+1)^3+n_0^2(n_0+1)^2}{4}=\frac{(n_0+1)^2\left[4(n_0+1)+n_0^2\right]}{4}=\frac{(n_0+1)^2\left[n_0^2+4n_0+4\right]}{4}=$$

$$=\frac{(n_0+1)^2(n_0+2)^2}{4}=\frac{n^2(n+1)^2}{4}$$

Cognome .... Nome ....Matricola ....

Esercizio 3 (20%) Ricava un ASF deterministico per il linguaggio unione dei linguaggi su $\Sigma=\{\mathbf{0},\mathbf{1}\}$ descritti dalle espressioni regolari $\mathbf{1}(\mathbf{01})^*(\mathbf{0}+\mathbf{1})$ e $(\mathbf{10})^*$. Mostra anche la procedura adottata.

soluzione pedissequa:

1) trovo gli ASFND per i due linguaggi
2) ne faccio l'unione (non occorre trasformarli in ASF deterministici!)
3) trasformo l'ASFND ottenuto in un ASF

soluzioni più furbe:

1) Noto che il primo linguaggio è l'unione di due linguaggi: $\mathbf{1}(\mathbf{01})^*\mathbf{0}$ e $\mathbf{1}(\mathbf{01})^*\mathbf{1}$, di cui il primo $\mathbf{1}(\mathbf{01})^*\mathbf{0}$ può essere riscritto $(\mathbf{10})^*\mathbf{10}$ ed è contenuto nel terzo linguaggio $(\mathbf{10})^*$. Dunque è sufficiente trovare un ASF per $\mathbf{1}(\mathbf{01})^*\mathbf{1}$ unione $(\mathbf{10})^*$, cioè per $(\mathbf{10})^*\mathbf{11}+(\mathbf{10})^*$ cioè per $(\mathbf{10})^*$ $(\mathbf{11}+\varepsilon)$.

2) Molto simile al precedente: noto che il primo linguaggio può essere riscritto come $(\mathbf{10})^*(\mathbf{10}+\mathbf{11})$, mentre il terzo linguaggio $(\mathbf{10})^*$ può essere riscritto come $(\mathbf{10})^*\mathbf{10}+\varepsilon$. Dunque è sufficiente aggiungere ε alle stringhe riconosciute dall'automa che riconosce il primo linguaggio (cioè trasformare lo stato iniziale in stato finale) e verificare che l'automa sia deterministico.

Esercizio 4 (20%) Come si può dimostrare che se L è un linguaggio regolare, allora la relazione binaria di equivalenza R_L su Σ^* ha indice finito? Ricorda che la definizione formale di R_L è la seguente: $x\ R_L\ y \Leftrightarrow (\forall z\in\Sigma^*,\ xz\in L \Leftrightarrow yz\in L)$

Vedi appunti sulla dimostrazione del teorema di Myhill-Nerode

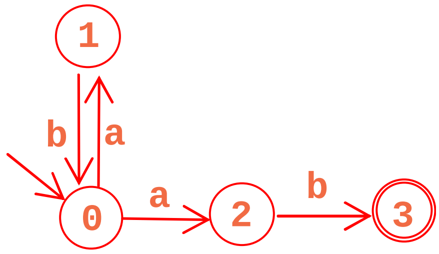
NOTE:
1) Non è legittimo utilizzare il teorema di Myhill-Nerode per dimostrare il teorema di Myhill-Nerode (cioè fare dei discorsi del tipo: "siccome accorpando le classi di R_M trovo le classi di R_L, allora R_M è più fina di R_L").
2) Non è richiesto dimostrare il vice-versa

Esercizio 5 (20%) Mostra le classi di equivalenza di Myhill-Nerode per il linguaggio $(\mathbf{ab})^*\mathbf{ab}$.

1) trovo un AFS (o ASFND) che riconosca il linguaggio
2) se ho trovato un AFSND lo trasformo in un ASF
3) trovo le classi di equivalenza di R_M
4) trovo le classi di equivalenza di R_L

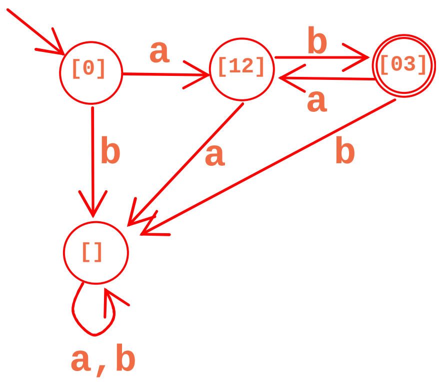
Cognome .... Nome ....Matricola ....

1) trovo un AFS (o ASFND) che riconosca il linguaggio

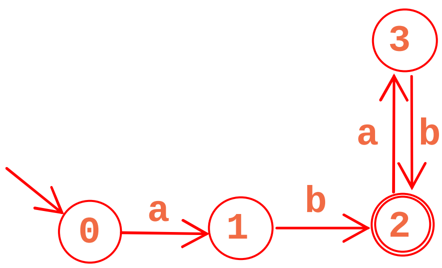


2) se ho trovato un AFSND lo trasformo in un ASF

	a	b
[0]	[1,2]	[]
[1,2]	[]	[0,3]
[]	[]	[]
[0,3]	[1,2]	[]



NOTA: visto che $(\mathbf{ab})^*\mathbf{ab} = \mathbf{ab}(\mathbf{ab})^*$ si poteva trovare direttamente un AFS, per esempio il seguente (in cui lo stato pozzo è taciuto):



3) trovo le classi di equivalenza di R_M

Nel primo caso ho:

$C_{[0]} = \{\varepsilon\}$ ⇢ $\mathbf{ab}(\mathbf{ab})^*$

$C_{[1,2]} = \{\mathbf{a}(\mathbf{ba})^*\}$ ⇢ $\mathbf{b}(\mathbf{ab})^*$

Cognome .... Nome ....Matricola ....

$C_{[0,3]}$ = { **ab(ab)***} ⇢ **(ab)***

$C_{[]}$ = $\{x\in\Sigma^* \mid \forall z\in\Sigma^*, xz\notin\Sigma^*\}$ ⇢ $\varnothing$

Nel secondo caso ho:

C_0 = {ε} ⇢ **ab(ab)***

C_1 = {**a**} ⇢ **b(ab)***

C_2 = { **ab(ab)***} ⇢ **(ab)***

C_3 = { **aba(ba)***} ⇢ **b(ab)***

C_{pozzo} = $\{x\in\Sigma^* \mid \forall z\in\Sigma^*, xz\notin\Sigma^*\}$ ⇢ $\varnothing$

4) trovo le classi di equivalenza di R_L

Nel primo caso le classi di R_L coincidono con quelle di R_M

Nel secondo caso le due classi C_1 e C_3 possono essere fuse (ammettono gli stessi completamenti) dando luogo a:

$C_{1,3} = C_1 \cup C_3$ = {**a**+**aba(ba)***} = {**a(**ε+**ba(ba)*****)**} = {**a(ba)***}
che coincide con $C_{[1,2]}$ del primo caso.

Sono considerati errori gravi:
- Cercare classi di equivalenza in un ASFND
- Produrre classi di R_M intersecanti
- Produrre classi di R_L intersecanti
- Fondere impropriamente classi di R_L