

Linguaggi di Programmazione e Compilatori
A.A. 2009/2010
Homeworks – Lexical Analysis
Doc. Maria Rita Di Berardini

Questa raccolta di esercizi sostituisce una parte (quella relativa all'analisi lessicale) dello scritto (ma **non** dell'orale) dell'esame di Linguaggi di Programmazione e Compilatori. È in realtà la prima di una serie di "homeworks" previsti dal docente durante il periodo didattico (tre in totale).

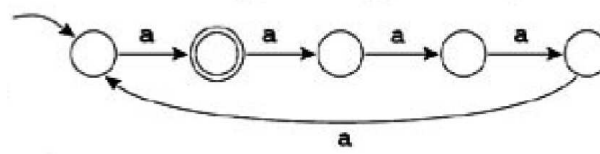
Regolamento

- per poter accedere all'orale è necessario *sostenere e superare* tutti gli homeworks assegnati.
- La soluzione degli esercizi proposti deve essere consegnata tassativamente entro le ore *12:00* di *Venerdì 09 aprile 2010*. La consegna può essere effettuata sia tramite posta elettronica (inviare una mail con oggetto *Lexical Analysis* all'indirizzo `mariarita.diberardini@unicam.it`) sia di persona (presso il mio ufficio, ultimo piano del polo didattico).
- Tutte le soluzioni consegnate oltre tempo massimo (ore 12:00 di Venerdì 09 aprile 2010) *non verranno prese in considerazione*.

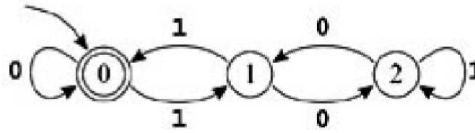
Esercizi

1. Fornire un'espressione regolare per ognuno dei seguenti linguaggi.
 - (1.1) le stringhe sull'alfabeto $\Sigma = \{a, b, c\}$ in cui la prima a precede la prima b ;
 - (1.2) le stringhe binarie che rappresentano numeri multipli di 4;
 - (1.3) le stringhe binarie che rappresentano numeri maggiori di 101001;
 - (1.4) le stringhe sull'alfabeto $\Sigma = \{a, b, c\}$ che non contengono la sottostringa *baa*.
2. Dire se esiste un'espressione regolare in grado di denotare ciascuno dei seguenti linguaggi.
 - (2.1) le stringhe sull'alfabeto $\Sigma = \{a, b\}$ che contengono un numero di a maggiore del numero di b ;
 - (2.2) le stringhe sull'alfabeto $\Sigma = \{a, b\}$ che sono palindrome.
3. Dimostrare che le espressioni regolari $(a|b)^*$ e $(a^*|(b|\epsilon))^*$ sono equivalenti

4. Quale è il linguaggio riconosciuto dai seguenti automi?



(a)



(b)

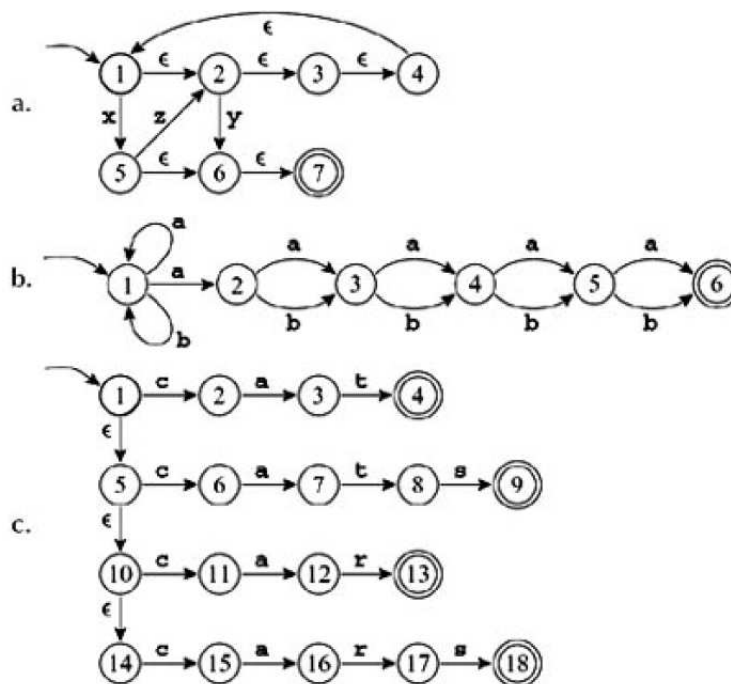
5. Convertire le seguenti espressioni regolari in automi a stati finiti non deterministici.

(4.1) **if | then | else**

(4.2) $a((b|a^*c)x)^*jx^+a$.

6. Convertire il DFA (b) dell'Esercizio 3 in un'espressione regolare.

7. Convertire i seguenti NFA in automi a stati finiti deterministici.



8. Utilizzare l'algoritmo di Thompson per costruire un NFA N in grado di riconoscere il linguaggio denotato dall'espressione regolare $0^*(01|10)0^*$.

9. Fornire un DFA minimo per i linguaggi denotati da ciascuna delle seguenti espressioni regolari.

(9.1) $(ab|ac)^*$;

(9.2) $a(a|b)^*a$.

10. Si consideri il seguente insieme di patterns

$$\begin{array}{ll} (aba)^+ & \{\} \\ ab^*a & \{\} \\ a \mid b & \{\} \end{array}$$

e si simuli il comportamento dell'algoritmo di pattern-matching in corrispondenza dell'input
abaabbaaba