

Corso di Laboratorio di linguaggi di programmazione e compilatori

Compito scritto di esame

7 luglio 2004

Docente: Prof. Simone Brunozi
email: labcompilatori@wedoit.us
www: www.wedoit.us/labcompilatori/

file di riferimento:
esame_scritto__2004-07-07_v1.0

URI di riferimento:
non disponibile

Questo documento è rilasciato sotto licenza Creative Commons (www.creativecommons.org),
il testo della licenza è reperibile agli URI
<http://creativecommons.org/licenses/by-sa/1.0/>
<http://creativecommons.org/licenses/by-sa/1.0/legalcode>

This work is licensed under the Creative Commons Attribution-ShareAlike License.
To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/>
or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Attribution



Share-Alike



E' possibile reperire materiale licenziato con Creative Commons presso l'URI
<http://commoncontent.org/>

Durata del compito scritto: **120 minuti**

Punteggi delle domande:

domanda	punteggio	domanda	punteggio
01	2,0	06	8,0
02	3,0	07	3,0
03	3,5	08	6,0
04	3,5	09	7,0
05	4,0		

Totale: 40,0 punti

COGNOME e nome: _____/_____

scrivere in stampatello con una calligrafia ben comprensibile

Domanda 07 [punti 3,0]

Associare (con delle frecce, ad esempio) i vari tipi di automi con la corretta definizione.

Tipo di automa		definizione
Pushdown		Macchina astratta che ha accesso a un ammontare di memoria potenzialmente illimitato, nella forma di due stack.
		Macchina astratta che ha accesso a un ammontare di memoria potenzialmente illimitato, nella forma di due pile LILO.
A stati finiti		Macchina astratta che ha solamente un ammontare di memoria finita e costante, nella forma di una pila LILO.
		Macchina astratta che ha solamente un ammontare di memoria finita e costante, nella forma di una pila LILO e uno stack singolo.
Turing machine		Macchina astratta che ha solamente un ammontare di memoria finita e costante.
		Macchina astratta che ha accesso a un ammontare di memoria potenzialmente illimitato, nella forma di un singolo stack.
		Macchina astratta che ha solamente un ammontare di memoria finita e costante, nella forma di due stack.

Domanda 08 [punti 6,0]

Siamo in ambiente di lavoro Linux Debian 3.0, e usiamo una shell Bash. Dopo aver lanciato il comando: `vim prova.txt` ci ritroviamo nell'ambiente di lavoro dell'editor vim. Questa è la sequenza di tasti che da questo momento vengono pigiati (i simboli `<esc>` `<return>` indicano i rispettivi tasti, e si presume che le lettere maiuscole siano ottenute pigiando il tasto corrispondente insieme al tasto `<shift>`; **NON** considerate un ulteriore `<return>` quello tra la prima e la seconda riga qui sotto):

**`iciao<esc><esc>ddPiP<esc>P:<esc>ry:w<return>i<return>
:w<return><esc>ii<esc>:q!<return>`**

A questo punto ci ritroviamo di nuovo al prompt della nostra shell.

Indicare l'output dei seguenti comandi (usando una casella per ogni carattere, come nell'esempio):

<pre>sim@deb:~\$ cat prova.txt</pre>	<pre>sim@deb:~\$ cat prova.txt tail -2</pre>
--------------------------------------	--

<pre>sim@deb:~\$ grep -n "ciao" prova.txt</pre>	<pre>sim@deb:~\$ grep -ni "ciao" prova.txt</pre>
---	--

<pre>sim@deb:~\$ grep -niv "ciao" prova.txt</pre>	<pre>sim@deb:~\$ grep -cv "ciao" prova.txt</pre>
---	--

Domanda 09 [punti 7,0]

Data la regex $b(a|c)+b^*(a)?c$ con alfabeto (a,b,c) :

a) costruire l'automa NFSA; **[punti 2,0]**

b) trasformare il precedente NFSA in DFSA con l'algoritmo di subset construction; **[punti 3,0]**

c) minimizzare il DFSA ottenuto. **[punti 2,0]**

IMPORTANTE: qui è sufficiente disegnare il risultato finale senza i vari passaggi intermedi.

