

2. Si consideri la grammatica

S → ASB  
A → aA | cA | a  
B → bB | cB | ε

- Si dica quale delle seguenti stringhe appartiene al linguaggio, mostrando, in caso positivo, una generazione
  - accaa
  - ababa
  - bcbcb
- Si disegni un albero sintattico delle stringhe che appartengono alla grammatica
- Si verifichi che la grammatica è ambigua mostrando un esempio

3. Si consideri la grammatica

S → aSa | aC  
C → CB | Cd | d  
B → Bb | c

- Eliminare la ricorsione a sinistra e fattorizzare la grammatica
- Si calcolino i First e i Follow e si produca la tabella di analisi
- Dimostrare il funzionamento dell'analizzatore sintattico, facendo vedere lo stack e l'input passo per passo, quando si analizza la stringa: aaddcba

Gli esercizi b e c sono facoltativi

3. Il comando Unix "last" mostra gli utenti che si sono connessi al sistema. Per ogni connessione, vengono visualizzati da destra a sinistra, divisi da caratteri di spaziatura: il nome dell'utente; il tty da cui la connessione avviene; l'indirizzo della macchina da cui la connessione avviene o, nel caso della connessione attraverso console, il numero del terminale; il giorno della settimana, il mese, il giorno e l'ora della connessione. In fondo alla riga, c'è la stringa "still logged in" se l'utente è ancora connesso. Altrimenti, dopo un "-", c'è l'ora di disconnessione e, fra parentesi tonde, la durata della connessione.

pipipo	ttyp3	venere.dii.unisi	Wed Jun 26 17:58 - 17:58	(00:00)
pipipo	ttyp2	venere.dii.unisi	Wed Jun 26 17:09 - 18:11	(01:01)
pluto	ttyp6	:4.0	Wed Jun 26 15:12	still logged in
pluto	ttyp4	aussie.dii.unisi	Wed Jun 26 15:11	still logged in
paperino	ttyp1	escher.dii.unisi	Wed Jun 26 10:25 - 10:44	(00:19)
paperino	ttyp1	giove.dii.unisi.	Wed Jun 26 10:02 - 10:25	(00:22)
minni	ftp	10.1.100.42	Tue Jun 25 10:47 - 10:48	(00:00)
minni	ftp	10.1.100.42	Tue Jun 25 09:21 - 09:36	(00:15)
pipipo	ftp	firewall.dii.uni	Thu Jun 20 18:47 - 18:47	(00:00)
pipipo	ftp	firewall.dii.uni	Thu Jun 20 17:51 - 17:54	(00:02)

Si fornisca un'espressione per grep (posix 1003.2) che consenta di individuare tutte le righe relative a connessioni che: sono di tipo "ftp"; provengono da una macchina del dominio "dii"; iniziate fra le 22 e le 4 (di notte); durate più di un'ora e mezzo.

1. Si consideri la grammatica

S → AB | ε  
A → aAb | Ab | ε  
B → bBa | bB | ε

- a. Si dica quale delle seguenti stringhe appartiene al linguaggio, mostrando, in caso positivo, una generazione
  - abbba
  - aabaa
  - abba
- b. Si disegni un albero sintattico delle stringhe che appartengono alla grammatica
- c. Si verifichi che la grammatica è ambigua mostrando un esempio

4. Si consideri la grammatica

$S \rightarrow SAb \mid c$   
 $A \rightarrow A(B) \mid A(B,B) \mid \epsilon$   
 $B \rightarrow Bb \mid c$

- a. Eliminare la ricorsione a sinistra e fattorizzare la grammatica
- b. Si calcolino i First e i Follow e si produca la tabella di analisi
- c. Dimostrare il funzionamento dell'analizzatore sintattico, facendo vedere lo stack e l'input passo per passo, quando si analizza la stringa:  $c(c)(cb,c)b$

Gli esercizi b e c sono facoltativi

4. Il comando Unix "ps aux" mostra i processi attivi . Per ogni processo, vengono visualizzati da destra a sinistra, divisi da caratteri di spaziatura: il nome dell'utente proprietario del processo; un numero progressivo che identifica il processo; la percentuale del tempo di CPU usata; la percentuale di memoria usata; la dimensione virtuale (SIZE); la dimensione in memoria (RSS); il tty che controlla il processo; lo stato del processo; la data o l'ora di inizio; il tempo complessivo di CPU usato; il comando che ha iniziato il processo.

USER	PID	%CPU	%MEM	SIZE	RSS	TTY	STAT	START	TIME	COMMAND
at	198	0.0	0.0	892	112	?	S	Jun 12	0:00	/usr/sbin/atd
bin	131	0.0	0.1	820	212	?	S	Jun 12	0:04	/sbin/portmap
pippo	29455	0.0	1.0	1908	1280	p4	S	15:11	0:00	-bash
pippo	29500	0.0	3.5	6216	4524	p6	S	15:14	0:00	emacs surname.tex
pippo	30075	0.0	0.4	956	536	p4	R	18:13	0:00	ps aux
pluto	28271	0.0	1.0	1948	1324	p1	S	11:04	0:00	-bash
root	1	0.0	0.0	260	76	?	S	Jun 12	0:05	init [2]
root	2	0.0	0.0	0	0	?	SW	Jun 12	0:00	(kflushd)
root	3	0.0	0.0	0	0	?	SW	Jun 12	0:03	(kupdate)
root	4	0.0	0.0	0	0	?	SW	Jun 12	0:00	(kpiod)
root	5	0.0	0.0	0	0	?	SW	Jun 12	0:01	(kswapd)
root	74	0.0	0.2	1120	348	?	S	Jun 12	0:05	usr/lib/postfix/mast

Si fornisca un'espressione per grep (posix 1003.2) che consenta di individuare tutte le righe relative a processi che: sono dell'utente "pippo"; occupano fra il 10 e il 35% della memoria; la cui dimensione virtuale (SIZE) è maggiore o uguale a 50000; lanciati da comandi che si trovano nella directory "/usr/sbin".

1. Data l'espressione regolare

$(x(xy|xz))^*xx^*$

- a. Si disegni il diagramma dell'automa deterministico che riconosce il linguaggio descritto da tale espressione, mostrando anche i passi intermedi della costruzione.

- b. Dato l'automa ricavato in precedenza, si derivi l'espressione regolare che rappresenta il linguaggio riconosciuto da tale automa. Ovviamente, l'espressione che si ottiene sarà equivalente a quella di partenza, ma l'esercizio richiede di mostrare i passi necessari per derivarla.

1. Data l'espressione regolare

$$x^*(xz|xy)^*yx$$

- a. Si disegni il diagramma dell'automa deterministico che riconosce il linguaggio descritto da tale espressione, mostrando anche i passi intermedi della costruzione.
- b. Dato l'automa ricavato in precedenza, si derivi l'espressione regolare che rappresenta il linguaggio riconosciuto da tale automa. Ovviamente, l'espressione che si ottiene sarà equivalente a quella di partenza, ma l'esercizio richiede di mostrare i passi necessari per derivarla.