

1. Si consideri la grammatica

$S \rightarrow Rc \mid cR$
 $R \rightarrow aBR \mid Rab \mid \varepsilon$
 $B \rightarrow b \mid baB$

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

2. Si consideri la grammatica

$S \rightarrow SA \mid d$
 $A \rightarrow aBa \mid aa$
 $B \rightarrow Bb \mid c$

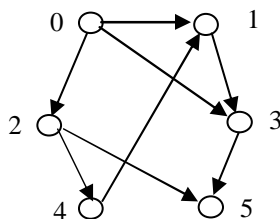
- Eliminare la ricorsione a sinistra e fattorizzare la grammatica
- Si calcolino i First e i Follow e si produca la tabella di analisi
- Dimostrare il funzionamento dell'analizzatore sintattico, facendo vedere lo stack e l'input passo per passo, quando si analizza la stringa: dacbba

3. Il comando Unix "ls -l" mostra il contenuto di una directory. Per ogni file, vengono visualizzati da destra a sinistra, divisi da caratteri di spaziatura: 10 caratteri che definiscono il tipo di file e i diritti di accesso; per le directory, il numero di file che contiene (o 1 per i file); il nome dell'utente proprietario; il gruppo a cui appartiene il file; la dimensione; mese, giorno e ora dell'ultima modifica, se fatta nell'anno corrente, oppure mese, giorno e anno; il nome del file.

-rw-r--r--	1	pip	users	44691	Jun	10	16:58	Fondamenti4.tex
-rwxr-xr-x	1	pip	users	100	Jun	10	17:21	Fourpages
-rw-r--r--	1	pip	users	19968	Jun	12	17:06	MPS2.doc
drwx-----	2	pip	users	2048	Jun	10	17:31	Mail
-rw-r--r--	1	pip	users	22528	Jun	12	10:21	Preventivo.doc
drwxr-xr-x	2	pip	users	2048	Oct	17	2001	bibbiena
drwx-----	2	pip	users	2048	Feb	15	11:24	mail
drwxr-xr-x	2	pip	users	2048	Jun	27	2000	public_html
drwxr-xr-x	3	pip	users	2048	Feb	21	17:24	seminari
drwxr-xr-x	59	pip	users	2048	May	13	11:03	tex
-rwxr-xr-x	1	pip	users	53	Jun	10	17:29	twopage

Si fornisca un'espressione per grep (posix 1003.2) che consenta di individuare tutte le righe relative a directory che: contengono da 8 a 128 files; di cui l'utente "pip" è proprietario; il cui nome inizia per "sys"; creati l'ultimo giorno del mese di dicembre o novembre.

4. Si consideri il seguente grafo



- Si disegni le rappresentazioni attraverso liste di adiacenza e matrici di adiacenza.
- Supponendo di visitare il grafo partendo dal nodo 0, si disegni
 - L'albero di ricerca in profondità
 - L'albero di ricerca in ampiezza
- Si supponga di ordinare i nodi attraverso le visite al punto precedente. Si indichi l'ordinamento (scrivendo la relativa sequenza di nodi) che si ottiene nel caso che si faccia uso dei seguenti metodi:
 - Visita in ampiezza =
 - Visita in profondità =
 - Ordinamento topologico =

5. L'espressione regolare

$[dD]\backslash.([pP][gG])\backslash.[rR]\backslash.[[:space:]]+[0-3][0-9]\backslash.(0[1-9]|1[0-2])\backslash.0[0-2]$

è soddisfatta da:

- ☐ Le stringhe con prefisso dpr o dgr seguito da uno o più spazi e da un numero di 6 cifre che inizia per 0
- ☐ Le stringhe con prefisso d.p.r. o d.g.r. (indipendentemente dalle maiuscole/minuscole), seguito da almeno un spazio e da una data col formato gg.mm.aa per gli anni dal 2000 al 2002.
- ☐ Le stringhe con prefisso d.p.r. o d.g.r. (indipendentemente dalle maiuscole/minuscole) seguito da uno o più spazi e da una data col formato gg.mm.aa per tutti i mesi esclusi quelli fra Ottobre e Dicembre.
- ☐ Tutte le stringhe con un prefisso di tre caratteri separati dal punto ('.') seguito da almeno un carattere di spazio e da tre coppie di cifre separate da un punto.

6. Detto T un insieme finito di simboli terminali e V un insieme finito di simboli non terminali, una grammatica libera dal contesto è caratterizzata da

- ☐ Produzioni della forma $\alpha \rightarrow \beta$ con $\alpha \in V$ e $\beta \in (V \cup T)^*$
- ☐ Produzioni della forma $\gamma \alpha \delta \rightarrow \gamma \beta \delta$ con $\alpha \in (V \cup T)^+$ e $\beta, \gamma, \delta \in (V \cup T)^*$
- ☐ Produzioni della forma $\alpha \rightarrow \beta$ con $\alpha, \beta \in (V \cup T)^*$
- ☐ Produzioni della forma $A \rightarrow b$ con $A \in V$ e $b \in T$

7. Una grammatica LR(k) è definita come

- Una grammatica che non è ambigua
- Una grammatica libera dal contesto per la quale si può costruire un analizzatore ricorsivo ascendente che analizzando la stringa da sinistra a destra riesce a scegliere la riduzione corretta osservando k simboli di previsione
- Una grammatica che permette di costruire un analizzatore che scandisce la stringa da sinistra a destra a blocchi di k simboli
- Una grammatica libera dal contesto che ammette al più k passi di backtracking da parte di un analizzatore ricorsivo ascendente

1. Si consideri la grammatica

$S \rightarrow aBa \mid bAb \mid abSba \mid baSab \mid c$

$A \rightarrow aSa \mid \varepsilon$

$B \rightarrow bSb \mid \varepsilon$

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

2. Si consideri la grammatica

S->SABd | d

A->aAa | ac

B -> Bb | ε

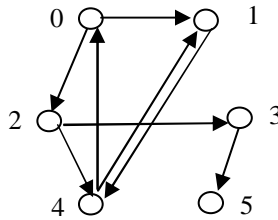
- Eliminare la ricorsione a sinistra e fattorizzare la grammatica
- Si calcolino i First e i Follow e si produca la tabella di analisi
- Dimostrare il funzionamento dell'analizzatore sintattico, facendo vedere lo stack e l'input passo per passo, quando si analizza la stringa: daacad

3. 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 di inizio (o l'ora di inizio se la data è oggi); 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: occupano la cpu più del 75%; sono controllati da un tty il cui nome è un "p" seguito da un intero (ad esempio, p1, p13,...); sono iniziati dopo il 15 giugno; sono una shell bash, cioè sono stati lanciati da un comando "-bash".

5. Si consideri il seguente grafo



- a. Si disegni le rappresentazioni attraverso liste di adiacenza e matrici di adiacenza.
- b. Supponendo di visitare il grafo partendo dal nodo 0, si disegni
 - L'albero di ricerca in profondità
 - L'albero di ricerca in ampiezza
- c. Si supponga di ordinare i nodi attraverso le visite al punto precedente. Si indichi l'ordinamento (scrivendo la relativa sequenza di nodi) che si ottiene nel caso che si faccia uso dei seguenti metodi:
 - Visita in ampiezza =
 - Visita in profondità =
 - Ordinamento topologico =

5. L'espressione regolare

`([Aa][Rr][Pp][Aa])[Rr][[:space:]]-[:space:]](1[4-9]|2[0-3]):([14]5|[03]0)`

è soddisfatta da:

- ☐ Le stringhe con prefisso arr o par seguito da uno o più spazi, da un trattino ('-'), da uno spazio e da un numero di 4 cifre che inizia per 1 o 2.
- ☐ Le stringhe con prefisso arr o par (indipendentemente dalle maiuscole/minuscole), seguito da uno spazio, dal trattino ('-'), da un altro spazio e da un orario con formato hh:mm riguardante tutti i quarti d'ora dopo le 14.
- ☐ Le stringhe con prefisso arr o par (indipendentemente dalle maiuscole/minuscole), seguito da uno spazio, dal trattino ('-'), da un altro spazio e da un orario con formato hh:mm riguardante tutte le ore dopo le 12.
- ☐ Tutte le stringhe con un prefisso di tre caratteri separati dal punto seguito da almeno un carattere di spazio, da un trattino ('-') e da due coppie di cifre separate dai due punti (':').

6. Data una stringa di simboli generata da una qualsiasi grammatica libera dal contesto, l'analisi discendente ricorsiva consiste in

- ❑ La costruzione dell'albero sintattico a partire dalle foglie scegliendo sempre la riduzione corretta
- ❑ La costruzione di un albero sintattico a partire dalla radice, ovvero dallo scopo S del linguaggio, utilizzando eventualmente il backtracking
- ❑ La costruzione dell'albero sintattico attraverso l'espansione della produzione corretta ad ogni passo
- ❑ La costruzione dell'albero sintattico utilizzando un simbolo di previsione per scegliere la riduzione corretta

7. Un analizzatore LR

- ❑ E' costituito da una unità di controllo, uno stack, un cursore sulla stringa di input che procede da sinistra a destra e da una tabella di analisi definita da una matrice Azione e da una Goto
- ❑ Prevede una tabella di controllo che permette di decidere se espandere una produzione della grammatica osservando il prossimo simbolo in ingresso
- ❑ E' una macchina a stati che analizza la stringa da sinistra a destra utilizzando un cursore
- ❑ E' costituito da uno stack che viene inizializzato con lo scopo del linguaggio S, da un cursore che scorre la stringa di ingresso da sinistra a destra e da una tabella di analisi M