

Tempo a disposizione: ore 2.

Svolgere gli esercizi 1-4 e 5-8 su due fogli differenti.

1. Si consideri la seguente grammatica G :

$$\begin{aligned} S &\rightarrow aS \mid A \\ A &\rightarrow \epsilon \mid aA \end{aligned}$$

(i) Determinare il linguaggio $L(G)$. (ii) È ambigua tale grammatica? In caso affermativo, produrre una grammatica G' non ambigua equivalente. Altrimenti, verificare se G sia di classe $LL(1)$.

2. Si consideri il linguaggio $L = \{w \in (0|1)^* \mid |w| \geq 2 \text{ ed il penultimo simbolo di } w \text{ è } 1\}$. Si produca un DFA che riconosca L .

3. Si consideri la grammatica G con simbolo iniziale S :

$$\begin{aligned} S &\rightarrow cAS \mid \epsilon \mid B \\ A &\rightarrow a \mid aA \\ B &\rightarrow b \mid bB \end{aligned}$$

(i) Determinare il linguaggio generato $L(G)$. (ii) Verificare che G non è di classe $LL(1)$. (iii) Manipolare la grammatica per ottenerne una equivalente G' di classe $LL(1)$. (iv) Costruire la tabella di parsing $LL(1)$ per G' . (v) Mostrare il funzionamento del parser $LL(1)$ su input caa .

4. Classificare il linguaggio $L = \{a^n b^{n+m} c^m \mid n, m \leq 0\}$, ovvero verificare se è regolare, oppure libero non regolare, oppure non libero.

5. Si dica, motivando la risposta, cosa viene stampato dall'esecuzione del `main` della seguente classe `Test` in Java.

```
class Y extends Throwable {
    int x=20;
}
class X extends Y {
    int x=20;
}
class C {
    void f() throws X, Y {
        throw new Y();
    }
    void g (int sw) throws X, Y {
        if (sw == 1) {f();
            throw new X(); }
        try {f();} catch (Y e) {System.out.println("in_g");}
    }
}

public class Test {
    public static void main(String[] args) throws X, Y {
        C c = new C();
        try {c.g(0);}
        catch (Y e) {System.out.println("in_main");};
        try {c.g(1);}
        catch (Y e) {System.out.println("in_main");};
    }
}
```

Viene stampato:
in_g
in_main

6. Si consideri il seguente frammento in uno pseudo-linguaggio con scope dinamico e parametri di ordine superiore:

```
int x = 700;
int n = 30;
```

```
void g(){
  write(n+x)
}
```

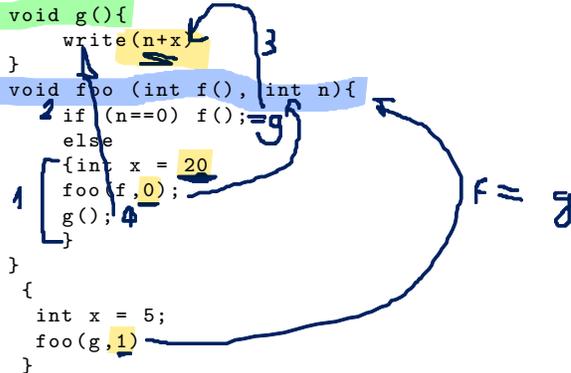
```
void foo (int f(), int n){
  if (n==0) f();
  else
  {int x = 20;
  foo(f, 0);
  g();
  }
```

```
  {
    int x = 5;
    foo(g, 1)
  }
}
```

Output:

20+0

20+1



Si dica cosa stampa il frammento con shallow binding.

7. Si consideri il seguente frammento di codice:

```
int x = 1;
int A[5];
int i;
for (i=0, i<5, i++) A[i]=i;
```

```
int fie(int name w,z){
  x = (w++) + (z++) + (w++) + (z++);
  write(x)
}
```

```
fie(x,A[x]);
}
```

Si dica qual'è il valore stampato e quali assunzioni sono necessarie per garantire tale risultato.

8. Si discuta brevemente la differenza esistente fra polimorfismo universale parametrico e polimorfismo universale di sottotipo.

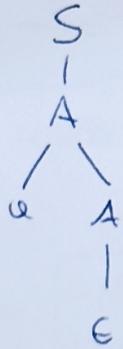
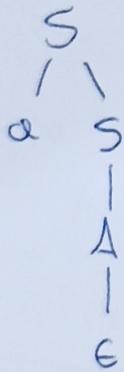
ES 1

$S \rightarrow \epsilon \mid A$

$A \rightarrow \epsilon \mid aA$

$L(G) = \{a^n \mid n \geq 0\}$, INFATTI S SI RISCRIVE IN 0 O PIU' a , MENTRE A SI RISCRIVE IN ϵ O $n \geq 1$ a .

CONSIDERIAMO LA STRINGA $a \in L(G)$



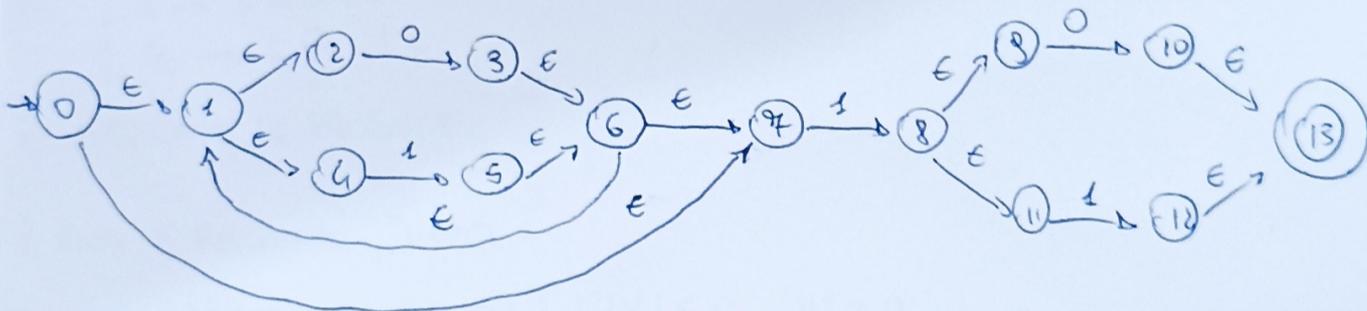
→ DUE ALBERI PER $a \in L(G)$, QUINDI G È AMBIGUA.

$G' [S \rightarrow aS \mid \epsilon]$ È NON-AMBIGUA E PRODUCE LO STESSO LINGUAGGIO.

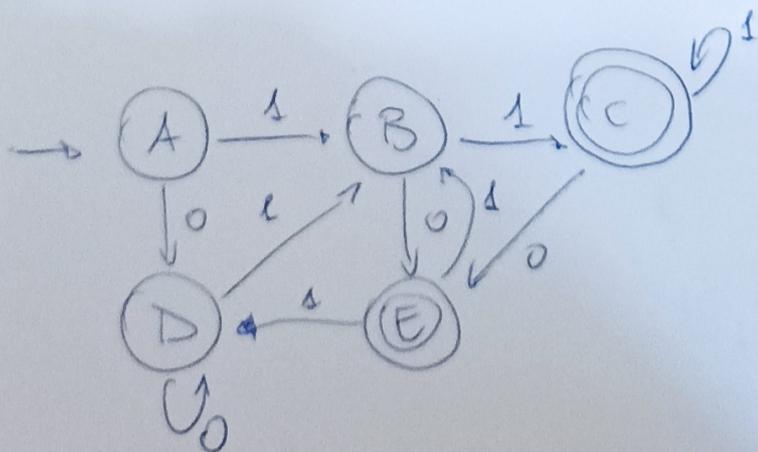
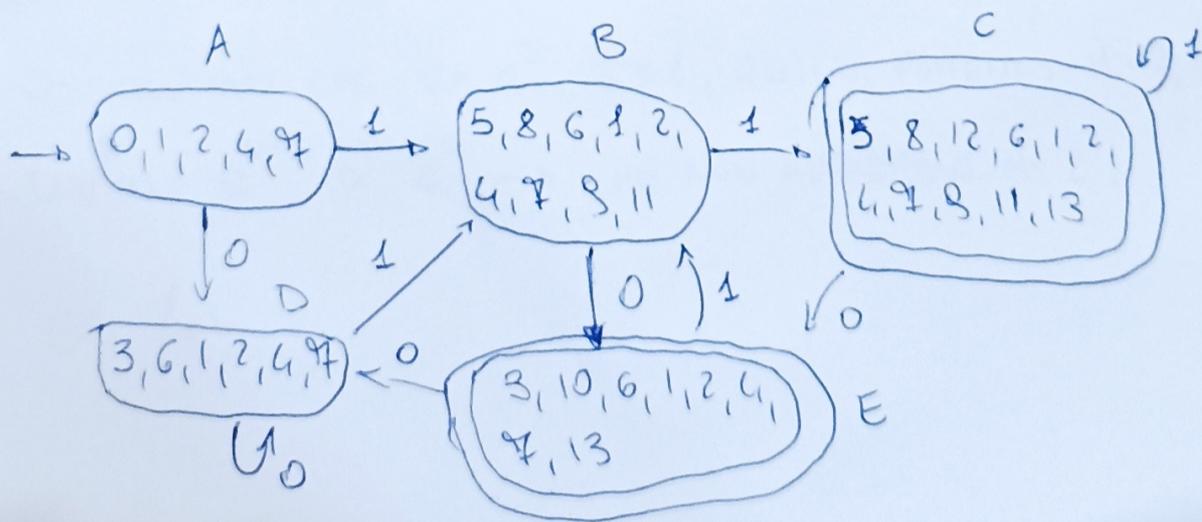
ES. 2

L'ESPRESSIONE $E := (011)^+ 1 (011)$

DESCRIVE L . USATO COSTRUZIONE CANONICA



OTTENIAMO DFA CON COSTRUZIONE PER SOTTO-INSIEMI DA NFA.



$L(G) := \{w \mid \text{ogni occorrenza di } c$
 $\text{è seguita da } n \geq 1 \text{ occorrenze}$
 $\text{di } a, \text{ e } w \text{ termina con } b^*\}$

ES. 3

$L(G) = L[(c a^+)^* b^+]$
 ESPRESSIONE REGOLARE

G NON È LL(1), INFATTI $\text{FIRST}(a) \cap \text{FIRST}(aA) = \{a\} \neq \emptyset$

FATTORIZZIAMO A SINISTRA G OTTENENDO G'

G'

- $S \rightarrow cAS \mid \epsilon \mid B$
- $A \rightarrow aA'$
- $A' \rightarrow A \mid \epsilon$
- $B \rightarrow bB'$
- $B' \rightarrow B \mid \epsilon$

$\rightarrow G' \in LL(1)$

	a	b	c	\$	FIRST	FOLLOW
S		$S \rightarrow B$	$S \rightarrow cAS$	$S \rightarrow \epsilon$	c, a, b	\$
A	$A \rightarrow aA'$				a	c, b, \$
A'	$A' \rightarrow A$	$A' \rightarrow \epsilon$	$A' \rightarrow \epsilon$	$A' \rightarrow \epsilon$	a, e	c, b, \$
B		$B \rightarrow bB'$			b	\$
B'		$B' \rightarrow B$		$B' \rightarrow \epsilon$	b, e	\$

INPUT

c a a \$

c a a \$

c a a \$

c a a \$

c a \$

c a \$

c a \$

\$

\$

\$

STACK

S \$

c A S \$

A S \$

a A' S \$

A' S \$

A S \$

a A' S \$

A' S \$

S \$

\$

ACCEPT

L È LIBERO PERCHÈ

$$G \begin{cases} S \rightarrow AB \\ A \rightarrow aAb \mid \epsilon \\ B \rightarrow bBc \mid \epsilon \end{cases}$$

È LIBERA E LO PRODUCE

L NON È REGOLARE, INPATTI:

$$\text{FISSATO } N > 0, z = UVW : |UV| \leq N, |V| > 0 \\ = a^N b^{2N} c^N$$

OSSERVIATO CHE $V = a^s, s \geq 1$, QUINDI PONENDO $k=2$, ABBIAMO CHE

$$UV^2W = a^{N+s} b^{2N} c^N \rightarrow \text{CHE NON APPARTIENE AD } L.$$

ES. 4!

ES. 5

X È SOTTOCLASSE DI Y, QUINDI
PUÒ ESSERE USATA DOVE DOVREBBE
COMPARIRE Y.

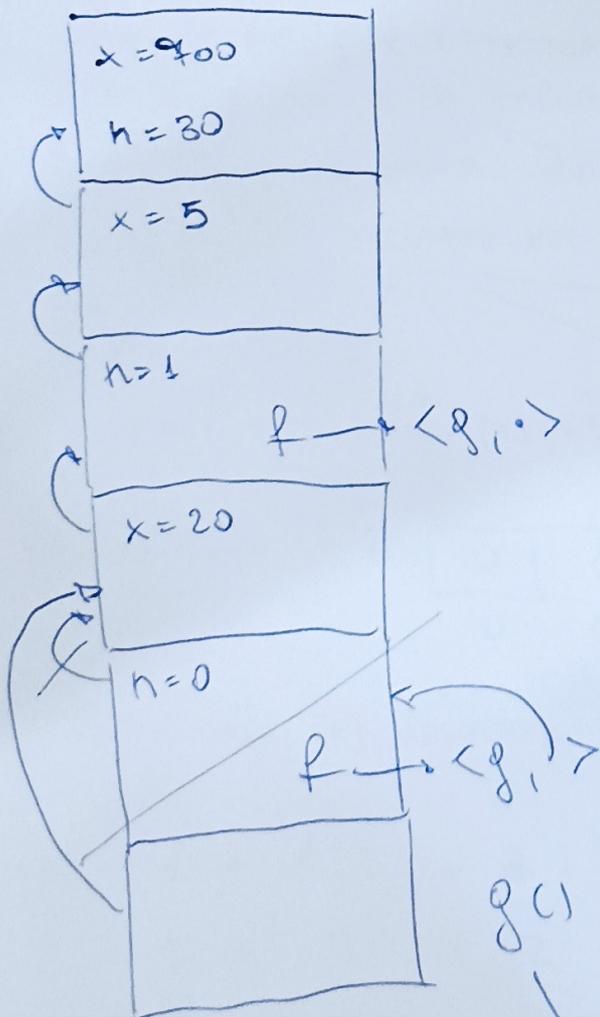
f() SOLLEVA ECCEZIONI DI CLASSE Y, g() SIA X CHE Y
NEL PRIMO BLOCCO TRY... CATCH, SIA g CHE main
CATTURANO ECCEZIONI Y. HA VISTO CHE LE ECCEZIONI RISALGONO
LA CATENA DINAMICA, LA PRIMA STAMPA AVVIENE IN g.

NEL SECONDO BLOCCO TRY... CATCH, IN g SI SEGUE IL
PRIMO RATIO CHE CAUSA UNA CHIAMATA AD f(). QUESTA VOLTA
g NON CATTURA L'ECCEZIONE, QUINDI ESSA RISALE LA CATENA
SINO AL main STAMPANDO LI. INOLTRE L'ESECUZIONE RIPRENDE
DOPO IL BLOCCO TRY... CATCH NEL TRAIN.

ES. 6

SCOPE DINAMICO + SHALLOW BINDING →

→ VALE L'AMBIENTE IN CUI AVVIENE LA CHIAMATA USANDO IL PARAMETRO FORTEALE.



MAIN 1

MAIN 2

foo 1

foo 1'

foo 2 QUI STAMPA
 $0 + 20 = 20$

A QUESTO PUNTO foo 2
TORNINA

QUI g STAMPA $1 + 20 = 21$

ES. 7

ESSENDOCI UNA SOLA X

SI FA SEMPRE RIFERIMENTO AD ESSA.

CON K++ L'ESPRESSIONE PRIMA RESTITUISCE
IL VALORE DI K E POI LO INCREMENTA DI 1.

LA FUNZIONE f(x) EFFETTUA PASSAGGIO PER NOTE, QUINDI SI PASSA
UNA CHIUSURA, E LA VARIABILE VIENE VALUTATA NELL'AMBIENTE IN CUI
f(x) VIENE CHIAMATA. QUINDI f(x) DOPO LA CHIAMATA SARÀ QUALCOSA
COME VALUTATE NEL MAIN.

f(x):

$$x = [x++] + [A[x]++] + [x++] + [A[x]++] \quad (*)$$

INIZIALMENTE A È

0	1	2	3	4
0	1	2	3	4

L'ESPRESSIONE (*) VALUTATA DA SINISTRA A DESTRA SARÀ:

$$\begin{aligned} x &= 1 + A[2] + 2 + A[3] \\ &= 1 + 2 + 2 + 3 \\ &= 8 \end{aligned}$$

A DOPO f(x) SARÀ

0	1	3	4	4
0	1	2	3	4

E VIENE STAMPATO 8.

ES. 8

• Nel POLIMORFISMO PARAMETRICO
UNIVERSALE, ~~IL TIPO~~

UN OGGETTO PUÒ AVERE UN'INFINITÀ DI TIPI CHE SI OTTENGONO PER INSTANZIAMENTO DAI TIPI DEL LINGUAGGIO. AD ESEMPIO DICHIARANDO

$T x$, STIAMO DICENDO CHE L'OGGETTO DI NOTE "X" PUÒ ESSERE DI UN QUALUNQUE TIPO DEL LINGUAGGIO. QUESTO È MOLTO UTILE PER CREARE FUNZIONI POLIMORFE.

• Nel POLIMORFISMO UNIVERSALE DI SOTTOTIPO IL CONCETTO È ^{SIMILE} ~~IL TIPO~~

TUTTAVIA UN OGGETTO PUÒ ESSERE INSTANZIATO SOLO CON ~~IL TIPO~~ ~~OGGETTI~~ SOTTOTIPI DI UN CERTO TIPO. AD ESEMPIO DICHIARANDO

$f: \forall D < T \rightarrow \text{void}$

f È UNA FUNZIONE CHE È DEFINITA SOLO PER I SOTTOTIPI DI T .