

# FAC SIMILE 2

3 gennaio 2022

## Indice

<b>1</b>	<b>Domande</b>	<b>2</b>
1.1	Domande su automi a stati finiti . . . . .	2
1.2	Domande su epsilon chiusure . . . . .	2
1.3	Domande su espressioni regolari . . . . .	3
1.4	Domande su pumping lemma . . . . .	3
1.5	Domande su grammatiche . . . . .	3
1.6	Domande su derivazioni . . . . .	4
1.7	Domande su automi a pila . . . . .	4
1.8	Domande su grammatiche LL(1) . . . . .	5
1.9	Domande su attributi . . . . .	5
1.10	Domande su analisi statica . . . . .	5
<b>2</b>	<b>Esercizi</b>	<b>6</b>
2.1	Esercizi su minimizzazione . . . . .	6
2.2	Esercizi su espressioni regolari . . . . .	6
2.3	Esercizi su grammatiche . . . . .	6
2.4	Esercizi su grammatiche LL(1) . . . . .	7
2.5	Esercizi su traduzione . . . . .	7

# 1 Domande

## 1.1 Domande su automi a stati finiti

### 1. Automi a stati finiti 3 MULTI Single

Dato il DFA  $A$

	$a$	$b$
$q_0$	$q_0$	$q_0$
$*q_1$	$q_0$	$q_3$
$\rightarrow q_2$	$q_1$	$q_1$
$q_3$	$q_1$	$q_0$

determinare quale delle seguenti stringhe è accettata da  $A$ .

- $abaaa$
- $aaaba$
- $ababb$
- $ababa$

Total of marks: 1

## 1.2 Domande su epsilon chiusure

### 1. Epsilon chiusura 3 MULTI Single

Detta  $\delta$  la funzione di transizione dell'automa seguente

	0	1	$\varepsilon$
$\rightarrow q_0$	$\emptyset$	$\{q_1\}$	$\emptyset$
$q_1$	$\emptyset$	$\emptyset$	$\{q_2\}$
$q_2$	$\emptyset$	$\emptyset$	$\{q_3, q_5\}$
$q_3$	$\{q_4\}$	$\emptyset$	$\emptyset$
$q_4$	$\emptyset$	$\emptyset$	$\{q_3, q_5\}$
$*q_5$	$\emptyset$	$\emptyset$	$\emptyset$

quale dei seguenti insiemi corrisponde a  $\hat{\delta}(q_2, 00)$ ?

- $\{q_1, q_4, q_5\}$
- $\{q_3, q_4, q_5\}$
- $\{q_0, q_4, q_5\}$
- $\{q_2, q_4, q_5\}$

Total of marks: 2

### 1.3 Domande su espressioni regolari

#### 1. Espressioni regolari 2

Data l'espressione regolare  $E = (c^* + c + (\varepsilon + b)(a + c))^*$  sull'alfabeto  $\{a, b, c\}$ , determinare quale delle seguenti stringhe appartiene al linguaggio generato da  $E$ .

- $bbc$
- $acbaa$
- $abbaa$
- $baabb$

Total of marks: 3

### 1.4 Domande su pumping lemma

#### 1. Pumping lemma 2

Si supponga di voler dimostrare il pumping lemma per il linguaggio regolare  $L$  riconosciuto dal seguente DFA con 5 stati:

	$b$	$c$
0	0	0
* 1	0	0
2	1	3
3	2	0
$\rightarrow$ 4	1	3

Si consideri ora la stringa  $w = cbcb b \in L$  di lunghezza 5. Per concludere la dimostrazione vista a lezione viene scelta una particolare scomposizione  $xyz$  di  $w$ . Quale?

- $x = cb, y = c, z = bb$
- $x = c, y = bcb, z = b$
- $x = c, y = bc, z = bb$
- $x = c, y = b, z = cbb$

Total of marks: 4

### 1.5 Domande su grammatiche

#### 1. Grammatiche 2

Data la grammatica  $G$  avente simbolo iniziale  $E$

$$\begin{aligned} A &\rightarrow \varepsilon \mid AcA \\ B &\rightarrow Eb \\ \rightarrow E &\rightarrow BA \mid \varepsilon \end{aligned}$$

individuare quale, tra le seguenti, è una stringa generata da  $G$ .

- $cbb$
- $bb$
- $cc$

Total of marks: 5

## 1.6 Domande su derivazioni

### 1. Derivazioni 1 ☐ MULTI ☐ Single

Data la grammatica  $G$

$$\begin{aligned} A &\rightarrow \varepsilon \\ C &\rightarrow AbD \\ D &\rightarrow A \mid CE \\ E &\rightarrow Aac \end{aligned}$$

individuare quale, tra le seguenti, è una derivazione canonica **destra** di  $G$ .

- $C \Rightarrow AbD \Rightarrow AbCE \Rightarrow AbAbDE \Rightarrow AbAbDAac$
- $C \Rightarrow AbD \Rightarrow AbCE \Rightarrow AbCAac \Rightarrow AbAbDAac$
- $C \Rightarrow AbD \Rightarrow AbA \Rightarrow Ab \Rightarrow b$
- $C \Rightarrow AbD \Rightarrow AbCE \Rightarrow AbAbDE \Rightarrow AbAbCEE$

Total of marks: 6

## 1.7 Domande su automi a pila

### 1. Automi a pila 1 ☐ MULTI ☐ Single

Sia  $P = (\{q\}, \{a, b, d\}, \{a, b, d, B, D, E\}, \delta, q, E, \{q\})$  un automa a pila che esegue la mossa

$$(q, bd, bD) \vdash_P (q, d, D)$$

Quale delle seguenti relazioni giustifica questa mossa?

- $(q, \varepsilon) \in \delta(q, d, d)$
- $(q, \varepsilon) \in \delta(q, \varepsilon, B)$
- $(q, \varepsilon) \in \delta(q, b, b)$
- $(q, bD) \in \delta(q, \varepsilon, B)$

Total of marks: 7

## 1.8 Domande su grammatiche LL(1)

### 1. Grammatiche LL(1) 3 ☐ MULTI ☐ Single

Le seguenti produzioni appartengono a una grammatica  $G$  avente simbolo iniziale  $C$ :

$$A \rightarrow \varepsilon \mid ddc \mid Edc$$

Quale tra le seguenti relazioni è **sicuramente** vera senza conoscere l'intera grammatica?

- $d \in \text{FOLLOW}(E)$
- $\text{FOLLOW}(C) \subseteq \text{FOLLOW}(A)$
- $c \in \text{FOLLOW}(A)$

Total of marks: 8

## 1.9 Domande su attributi

### 1. Attributi 1 ☐ MULTI ☐ Single

La seguente produzione con associate **regole semantiche**

$$A \rightarrow BCDE \quad \{A.s_0 = E.s_2, A.s_2 = E.s_2, B.e_2 = A.e_1, D.e_1 = C.s_0\}$$

in cui gli  $s_i$  (se presenti) sono attributi **sintetizzati** e gli  $e_i$  (se presenti) sono attributi **ereditati**, appartiene a una SDD:

- **che può essere L-attribuita ma non S-attribuita**
- che può essere S-attribuita
- che non è L-attribuita

Total of marks: 9

## 1.10 Domande su analisi statica

### 1. Analisi statica 2 ☐ MULTI ☐ Single

Ricordando le SDD viste a lezione per il calcolo dell'attributo *stack*, determinare il numero massimo di operandi contemporaneamente presenti sulla pila durante la valutazione della seguente espressione.

$$x - (3 - (8 - x))$$

**Nota:** per interpretare correttamente la struttura dell'espressione è fondamentale tenere presenti le usuali convenzioni di priorità e associatività di operatori e connettivi, elencati di seguito in ordine crescente di priorità:  $||$ ,  $\&\&$ , relazioni,  $!$ ,  $+$  e  $-$ ,  $*$  e  $/$ . Tutti gli operatori e i connettivi binari sono associativi a sinistra.

- 1

- 4
- 2
- 3

Total of marks: 10

## 2 Esercizi

### 2.1 Esercizi su minimizzazione

#### 1. Minimizzazione 1 ESSAY

Minimizzare il seguente DFA. Mostrare la **tabella di transizione completa** del DFA minimo i cui stati sono le classi di equivalenza risultanti dalla minimizzazione, anche nel caso in cui il DFA proposto sia già minimo.

	0	1
0	0	0
* 1	0	0
* 2	1	3
* 3	1	3
→ 4	1	2

Total of marks: 14

### 2.2 Esercizi su espressioni regolari

#### 1. Espressioni regolari 1 ESSAY

Definire una espressione regolare che generi il linguaggio delle stringhe di  $a$ ,  $b$  e  $c$  che non iniziano né finiscono con  $c$  e in cui le  $c$  (se presenti) sono tutte adiacenti.

Usare la seguente sintassi per scrivere l'espressione regolare, in cui  $\text{eps}$  rappresenta l'espressione regolare  $\varepsilon$  ed  $n$  rappresenta un simbolo dell'alfabeto. Ogni deviazione dalla sintassi indicata verrà considerata un **errore**.

$E \rightarrow \text{eps} \mid n \mid E + E \mid EE \mid E^* \mid (E)$

Total of marks: 18

### 2.3 Esercizi su grammatiche

#### 1. Grammatiche libere ESSAY

Definire una grammatica libera dal contesto che generi il linguaggio delle stringhe di  $a$ ,  $b$  e  $c$  in cui il numero di  $a$  è uguale al numero di  $b$  e le  $a$  (se presenti) precedono le  $b$  (se presenti). Nel riportare la soluzione adottare le seguenti convenzioni:

- Usare le lettere maiuscole A–Z per indicare **variabili** e le lettere minuscole a–z per indicare **simboli terminali**.
- Usare la sequenza di caratteri  $\rightarrow$  per separare la testa dal corpo di una produzione.
- Scrivere **esclusivamente** le produzioni della grammatica, una per riga e senza lasciare righe vuote tra esse. È comunque consentito l'utilizzo della barra verticale  $|$  per definire sulla stessa riga più produzioni con la stessa testa.
- Si intende che il **simbolo iniziale** della grammatica è la variabile che compare in testa alla **prima produzione** scritta.

Ad esempio, la grammatica con produzioni  $S \rightarrow \varepsilon \mid aSb$  che genera il linguaggio  $\{a^n b^n \mid n \geq 0\}$  può essere scritta come

$S \rightarrow$

$S \rightarrow aSb$

oppure come

$S \rightarrow \mid aSb$

Total of marks: 22

## 2.4 Esercizi su grammatiche LL(1)

### 1. Grammatiche LL(1) 1 ESSAY

Data la seguente grammatica, il cui simbolo iniziale è  $E$ , mostrare FIRST e FOLLOW di tutte le variabili e gli insiemi guida di tutte le produzioni. La grammatica è LL(1)?

$$\begin{array}{lcl}
 B & \rightarrow & cdD \\
 C & \rightarrow & b \\
 D & \rightarrow & DbB \mid \varepsilon \\
 \rightarrow E & \rightarrow & CdD
 \end{array}$$

Total of marks: 26

## 2.5 Esercizi su traduzione

### 1. Traduzione 2 ESSAY

Utilizzando le SDD per la generazione del codice intermedio viste a lezione, determinare il valore dell'attributo sintetizzato  $S.code$  per il seguente comando, assumendo che il valore dell'attributo ereditato  $S.next$  alla radice dell'albero sintattico annotato sia STOP.

```
while (!(z <= y && x < z)) z = y * 1;
```

Attenersi **rigorosamente** alle regole semantiche viste a lezione, anche se il codice è privo di senso, può essere semplificato o ottimizzato, causa un loop infinito o una divisione per zero, ecc.

**Nota 1:** la tabella in calce all'esercizio riassume le regole di traduzione più significative viste a lezione. Il comando da tradurre può contenere altri operatori o costrutti rispetto a quelli mostrati in tabella. A causa di limiti tecnici, nella tabella l'operatore di congiunzione logica è indicato con `and` invece che con `&&`.

**Nota 2:** per interpretare correttamente la struttura del comando è fondamentale tenere presenti le usuali convenzioni di priorità e associatività di operatori e connettivi, elencati di seguito in ordine crescente di priorità: `| |`, `&&`, relazioni, `!`, `+` e `-` (a pari priorità), `*` e `/` (a pari priorità). Tutti gli operatori e i connettivi binari sono associativi a sinistra.

Produzioni	Regole semantiche
$E \rightarrow n$	$E.code = ldc\ n.v$
$E \rightarrow x$	$E.code = iload\ x$
$E \rightarrow E_1 + E_2$	$E.code = E_1.code \parallel E_2.code \parallel iadd$
$B \rightarrow E_1\ R\ E_2$	$B.code = E_1.code \parallel E_2.code \parallel if\ icmpR\ B.true \parallel goto\ B.false$
$B \rightarrow B_1\ and\ B_2$	$B_1.true = newlabel()$ $B_1.false = B.false$ $B_2.true = B.true$ $B_2.false = B.false$ $B.code = B_1.code \parallel B_1.true : B_2.code$
$S \rightarrow x = E;$	$S.code = E.code \parallel istore\ x \parallel goto\ S.next$
$S \rightarrow if\ (B)\ S_1\ else\ S_2$	$B.true = newlabel()$ $B.false = newlabel()$ $S_1.next = S.next$ $S_2.next = S.next$ $S.code = B.code \parallel B.true : S_1.code \parallel B.false : S_2.code$
$S \rightarrow while\ (B)\ S_1$	$B.true = newlabel()$ $B.false = S.next$ $S_1.next = newlabel()$ $S.code = S_1.next : B.code \parallel B.true : S_1.code$

Total of marks: 30