

Fondamenti di Informatica A

Compito 3 – 20/6/2018

Cognome e Nome _____ matr. _____

Domanda 1. Si consideri una grammatica BNF con simbolo iniziale $\langle A \rangle$, simboli terminali “0” e “1”, e regole di produzione seguenti (omettiamo le virgolette attorno ai simboli terminali per semplicità):

$\langle A \rangle ::= 0 \langle B \rangle \mid 0$

$\langle B \rangle ::= 1 \langle A \rangle \mid 1$

Allora:

V F La grammatica può generare la stringa 010101

V F La grammatica può generare la stringa 001

V F La grammatica può generare la stringa 01

V F La grammatica può generare la stringa 111

Risposte:

- V: in generale la grammatica genera sequenze non vuote di 0 e 1, in cui le cifre si alternano, e 0 è sempre la prima cifra
- F
- V
- F

Domanda 2. Si consideri la macchina di Turing con alfabeto $\{blank, 0, 1\}$, stato iniziale q_0 , e tabella delle istruzioni seguente:

Stato corrente	Simbolo corrente	Nuovo simbolo	Nuovo stato	Spostamento
q_0	1	0	q_0	right
q_0	0	0	halt	---
q_0	blank	0	halt	---

Allora:

V F Se il nastro contiene inizialmente la sequenza 000111, e la testina di lettura-scrittura è posizionata sulla prima cifra a sinistra (quella sottolineata), allora al termine dell'esecuzione il nastro conterrà 000111

V F Se la MdT viene fatta partire con il nastro vuoto (cioè in cui tutte le caselle sono *blank*), allora al termine dell'esecuzione il nastro conterrà un solo simbolo 0

V F Se il nastro contiene inizialmente la sequenza 111, e la testina di lettura-scrittura è posizionata sulla prima cifra a sinistra (quella sottolineata), allora al termine dell'esecuzione il nastro conterrà la sequenza 0000

V F Se il nastro contiene inizialmente la sequenza 010, e la testina di lettura-scrittura è posizionata sulla cifra a destra (quella sottolineata), allora al termine dell'esecuzione il nastro conterrà la sequenza 101

Risposte:

- V
- V
- V
- F: al termine conterrà 0100

Domanda 3. Si consideri l'espressione booleana $R = (A \text{ AND } B) \text{ OR } (A \text{ XOR } B)$. Allora:

- Se A e B hanno valore diverso (uno dei due è *true* e l'altro *false*) allora il risultato è sempre *true*
- Se A e B hanno lo stesso valore (entrambi *true* o entrambi *false*) allora il risultato è sempre *true*
- Se A e B sono entrambi *true*, il risultato è *true*
- Il risultato dell'espressione è sempre *true*, qualunque siano i valori di A e B

Risposte:

-
- : se $A = B = false$, il risultato è *false*
-
- : se $A = B = false$, il risultato è *false*

Domanda 4. Si consideri la seguente funzione ricorsiva in linguaggio C:

```
int f(int n) {
    if (n > 2) {
        return n;
    } else {
        return f(n+2);
    }
}
```

- L'espressione $f(0)$ ha valore 2
- L'espressione $f(5)$ ha valore 5
- L'espressione $f(2)$ ha valore 2
- esiste qualche valore intero x tale che $f(x)$ restituisca un risultato negativo.

Risposte:

- : ha valore 4
-
- : ha valore 4
- : la funzione restituisce sempre un valore strettamente maggiore di 2

Domanda 5. Si considerino due numeri $A = 0100\ 0011_{2C}$ e $B = 0010\ 0010_{2C}$ rappresentati in complemento a due con $N = 8$ bit. Allora:

- A rappresenta un valore positivo
- B rappresenta un valore positivo
- B rappresenta un numero pari
- Il calcolo di $(A + B)$ non genera overflow e produce come risultato un valore negativo

Risposte:

-
-
-
- : sono due valori positivi e nella somma non si genera overflow ($A = 67, B = 34, A + B = 101$)

Domanda 6. Sapendo che inizialmente vale l'asserzione $x < 10$, scrivere negli appositi spazi le asserzioni più specifiche che valgano dopo ciascuna delle istruzioni del seguente frammento di codice (assumere che tutte le variabili siano di tipo `int`).

```
{ x < 10 }  
x = x + 1;  
{ _____ }
```

```
y = 2 * x;  
{ _____ }
```

```
y = y - 1;  
{ _____ }
```

Risposta: una possibilità (non è l'unica) è data dalle asserzioni seguenti:

```
{ x < 10 }  
x = x + 1;  
{ x ≤ 10 }  
y = 2 * x;  
{ x ≤ 10 and y ≤ 20 and y = 2x }  
y = y - 1;  
{ x ≤ 10 and y < 20 and y = 2x - 1 }
```

Si noti che i predicati $y \leq 20$ e $y < 20$ nelle asserzioni precedenti sono in realtà ridondanti in quanto possono essere derivati direttamente dagli altri predicati.