## Fondamenti di informatica II - esame del 14 febbraio 2020

- 1. Illustrare la notazione O(.) O grande e fornire la definizione di problema computazionalmente "facile".
- 2. Definire il problema decisionale Vertex Cover e mostrare che esso appartiene alla classe NP. (Suggerimento: fornire un algoritmo non deterministico che "indovina" l'insieme di nodi da cercare).
- 3. Si consideri il seguente programma per il calcolo dei numeri di Fibonacci

```
int fib (int n) {
  if (n < 2)
      return 1;
  return fib(n-1) + fib(n-2);
}</pre>
```

Specificare il numero di chiamate ricorsive effettuate quando n = 4. Esprimere la complessità in funzione del parametro n.

- 4. Quali sono gli ingressi e il risultato calcolato dalla macchina di Turing universale? Perché il risultato è considerato importante?
- 5. Cosa si intende per problema non decidibile (o non calcolabile)? Fornire un esempio di problema non decidibile.
- 6. Descrivere un analizzatore sintattico di tipo top-down specificando l'input dell'analizzatore, i possibili risultati calcolati e un possibile algoritmo per l'analisi.
- 7. Descrivere un automa a stati finiti che riconosce il linguaggio formato dalle stringhe binarie di a e b costituito da tutte e sole le stringhe binarie contenenti un numero dispari di a e un numero dispari di b (in qualunque posizione; quindi l'automa accetta le stringhe aaabbb, ababab, aabaabab e rifiuta le stringhe aabbb, b e aabaaba).
- 8. Sia data la grammatica G = (N, T, P, S) con insieme dei simboli non terminali  $N = \{S, A\}$ , insieme simboli terminali  $T = \{a, b\}$ , assioma S e produzioni specificate nel seguito:

```
S \rightarrow Sa (1)

S \rightarrow Ab (2)

S \rightarrow A (3)

A \rightarrow Ab (4)

A \rightarrow b (5)
```

Fornire un albero di derivazione della stringa bbbaa; specificare il tipo di grammatica e discutere se la grammatica è ambigua o no.

9. Fornire una grammatica per generare il linguaggio delle espressioni aritmetiche avente cinque simboli terminali: "id" (che rappresenta un identificatore), i simboli di operazione "+" e "-" e le parentesi tonde "(" e ")". Ovviamente la grammatica deve generare tutte e sole le stringhe che sono aritmeticamente corrette (ad esempio non deve generare la stringa "id+((id-id)" o la stringa "id++id-id").

L'eventuale costruzione di una grammatica non ambigua sarà particolarmente apprezzata.