

Domande Linguaggi

1. Parlami del concetto di computabilità di una funzione e del problema dell' **HALT** della macchina di **Turing**.
2. Dato un linguaggio:
 - ✖ Dire di che tipo è.
 - ✖ Scrivere una grammatica che lo genera.
3. Dato un automa a stati finiti (potresti anche disegnartelo tu), scrivere la grammatica regolare destra e sinistra, e l'espressione regolare.
Per l'automa fare riferimento slide 9 ASF.
4. A livello intuitivo spiegare il **Pumping lemma**.
5. Quali sono i pro e i contro dell'analisi **Top-down e Bottom-up**
6. Quali sono i componenti fondamentali di un **riconoscitore LR** e in generale come funziona.
(solitamente LR(0)) → Oracolo, stack e controller.

Domande modelli computazionali

1. Differenza tra **chiusura lessicale** e **chiusura dinamica** (con esempio di codice)
2. Differenza tra meccanismo di valutazione **call by name** e **call by value** (oppure tra **modello normale** e **modello applicativo**).
Osservazione: Il modello normale adotta un meccanismo di valutazione di tipo call by name. Il modello applicativo adotta un meccanismo di valutazione di tipo call by value.
3. Qual'è il **modello più utilizzato e perché?** (Il modello normale si usa solo nelle MACRO. Ha molti difetti: richiede macchina virtuale per lazy evaluation, parametri valutati più volte, richieste più risorse a run time in quanto valuta oggetti computazionali. Evita fallimenti non necessari, ma anche l'introduzione di NAN e infinity fanno la stessa cosa in maniera più efficiente. Inoltre è meno efficiente nei casi normali che sono la maggioranza.)
4. Come posso ottimizzare il meccanismo di valutazione della call by name?
5. **Simulare la call by name in Scala, Java e JS.**
(In generale ricordare di: Sostituire i parametri attuali con funzioni che restituiscano quel valore. Sostituire ogni uso di parametro formale con una chiamata alla funzione stessa).
6. A cosa servono le **chiusure** in JS? (rappresentare uno stato privato e nascosto, creare una comunicazione nascosta, creare strutture di controllo).
Citare i 3 pezzi di codice.

7. **Proto e Prototype** in JS.

8. Come posso **modificare** un **prototipo** e quali sono le eventuali **ripercussioni** su oggetti già creati e da creare?

9. Spiegare l'**ereditarietà prototipale** in JS (esempio persona-studente).

10. Che cosa intendiamo con **type augmenting** in JS?

11. Parlare del **costruttore Function** in JS.

12. Quali sono i differenti costruttori in JS e quali sono le differenze principali?

13. Quali sono le **caratteristiche** principali di **Scala**? Spiegare il concetto di **Tratto**.