

ORALE 06/06/2025

DOMANDE GORRIERI: MODULO 1

In alcune domande c'è la botta – risposta, precisa e secca, motivandola senza dilungarsi troppo.

In altri casi la domanda richiede di parlare un po'.

Se non ricordi qualcosa prova ad aiutarti e cerca di farti arrivare a concludere la risposta fornendoti qualche input.

In generale, aggiusta il tiro scegliendo le domande in base a come stai andando.

A volte parte dagli errori fatti nello scritto.

A volte chiede di fornire (o ti fornisce e chiede di spiegare qualcosa su) degli esempi.

(dove c'è la > ho appuntato le risposte date)

mentre tra () ho segnato delle considerazioni/ precisazioni

● candidato 1:

- Lei è felice che ha preso 30? Vuole la lode?

- Definizione di DPDA (te la fa scrivere sul foglio)

(Il prof dice che è di partenza questa domanda è difficilotta)

- vincoli del nondeterminismo

- I DFA possono essere visti come PDA

> sì, PDA senza PILA (parte potenziamento DFA/NFA)

- DFA che linguaggi riconoscono? Mentre i PDA?

- Esempio di linguaggio libero deterministico che non è regolare

(ti lascia pensare se ti vede confuso, gli fa costruire il DPDA, vuole che sei molto preciso però sui simbolismi ma ti lascia pensare)

- Riconoscere per pila vuota e stato finale è la stessa cosa per un DPDA

> In generale no, ma fa eccezione se un linguaggio gode della prefix property

- Cosa vuol dire che un linguaggio gode della prefix property?

- La classe dei linguaggi liberi deterministici coincide con la classe dei linguaggi LR(1)?

> sì, perché c'è un teorema che lo afferma...

- Perché abbiamo studiato i Parser LL(1) e gli altri (LR, SRL ecc...)

- Differenza tra Parser SRL(1) e LARL(1)

- Quindi un Parser SRL(1) e uno LARL(1), se ne costruisco le rispettive tabelle di Parsing, hanno gli stessi stati?

> No, perché la differenza è sulla quantità di reduce

- Gli LL(k) per ogni k includono i linguaggi liberi deterministici?
- Tesi di Turing cosa dice?

● Candidato 2:

- (Ha sbagliato il secondo esercizio, si comincia da lì)
- Descriva le regole di semantica operativa ES e_0 and e_1
- Grammatica Libera definizione
- Che differenza c'è fra la forma della grammatica regolare e la grammatica libera definizione
- Invece le grammatiche dipendenti da contesto che forma hanno?
- Esempio di linguaggio libero che non sia regolare, perché?
- Come può dimostrare che tale linguaggio non sia regolare
 - > Pumping Lemma a rovescio
- Intestazione del Pumping Lemma (ti dà il foglio per scriverla)
- (Ti dà una manina se non ti ricordi i giusti per ogni ed esiste).
- Chiede l'intestazione della negazione del Pumping Lemma
- Dimostrazione del Pumping Lemma (ti dà il foglietto, ti fa scrivere e tu parli spiegandogliela)
- Cos'è l'Halting Problem?
- (Se sei poco preciso lo dice, poi ti dà una mano)

● Candidato 3:

- Definizione di DFA (te la fa scrivere e ti chiede di essere molto preciso)
- Come è definita la funzione di transizione delta?
- Se volessi fare l'NFA come cambia la funzione di transizione delta?
- Che linguaggio riconosce questo automa?
- Se volessi riconoscere il linguaggio complementare di questo qua (della domanda precedente) come posso fare?
- Chiusura dei linguaggi regolari
- Dimostra che i linguaggi regolari sono chiusi per unione
- (Il prof ti corregge se vuole che usi un linguaggio più tecnico, ha corretto un "generare" con "descrivere")
- Si può fare la stessa cosa (allo stesso modo) per altre proprietà?
 - > Sì, unione, concatenazione e ripetizione si usano le espressioni regolari
- Per le altre proprietà invece? Come si fa per l'intersezione?
- Ma i linguaggi liberi sono chiusi per intersezione? Perché?

- Esempio di un linguaggio libero la cui intersezione con un altro non è libero
- Cos'è Lex? Come lo uso?
 - > vuole sapere cosa prende in input Lex, ecc...
- A grandi linee, com'è organizzato un file .l?
- Come si costruisce l'automa canonico LR(1)?
 - > Costruzioni per sottinsiemi e funzioni
- Qual'è l'item LR(1) di partenza dell'automa canonico LR(1)
- Per le produzioni per S, come lo calcola il simbolo di look-ahead?
 - > è il dollaro \$ (detto così, secco, gli andava bene)
- E in casi più complicati come si calcola il simbolo di look-ahead?

• Candidato 4:

(Il prof dice che c'è un po' di discrepanza tra le due parti, conclude dicendo "Dica la verità, le è piaciuta di più la mia parte" e sorride)

- Cos'è un compilatore?
- Quando un compilatore si dice corretto?
 - (Richiede precisione nei termini)
- Spieghi che cos'è la macchina intermedia?
- Di solito cos'è che si compila e si interpreta nei linguaggi? Cioè quali tipi di primitive vengono compilate o interpretate? Per esempio, un for verrebbe di solito compilato o interpretato? (compilato)
- Definizione di macchina fisica
- Che differenza c'è fra il linguaggio macchina fisica e il linguaggio macchina di Java (Virtual Machine).
 - Quali sono le primitive di una macchina fisica?
- Cosa dice il teorema di Jacopini-Bohm?
 - (Vuole che sei molto preciso, ma ti dà una mano)
- Perché il teorema di Jacopini è interessante?
- Sa cosa vuol dire programmazione strutturata?
 - (relativo alla seconda parte del corso)
- Condizione necessaria e sufficiente per avere una grammatica di classe LL(1)
 - (ti fa completare il teorema scrivendolo)
- Com'è fatta una tabella di Parsing LL(1)?
 - (Righe come son fatte? Colonne come son fatte?)
- Come si riempie la tabella di Parsing LL(1)?
- Dove metterebbe (in che casella della tabella di parsing) la seguente produzione?
 - (Non ho i raggi X, il prof l'ha scritta sul foglio)

- Senta un po', mi faccia un esempio di una grammatica che non è LL(1) ma è LL(2)?

(Può essere anche semplice, basta che abbia le caratteristiche richieste)

(Il prof ti aiuta, disegna l'esempio e ti chiede di spiegare perché è LL(2) e non LL(1))

- Cosa sta quel "2" in LL(2)?

- Fattorizza la grammatica (ti chiede di fattorizzare la grammatica che ti ha fornito)

- Trovato il linguaggio di questa grammatica (sempre quella di questo esempio), dire se è di classe LL(1)?

- Cos'è YACC? (Ci tiene molto che si dica che è un "generatore", qualsiasi altro termine viene contestato)

● Candidato 5: (entra con voto alto)

- Scriva l'intestazione del Pumping Theorem

- Dimostrazione (almeno a grandi linee)

(ti da' una mano ma se non capisce quello che intendi ovviamente ti chiede chiarimenti)

- Mi saprebbe dire le proprietà di chiusura dei linguaggi liberi? Come si dimostrano?

> unione, concatenazione, ripetizione

(Vuole che glielo fai vedere come l'ha fatto in classe)

(Il candidato è in difficoltà, il prof cerca di dare una mano scrivendo l'inizio del ragionamento)

(Alla fine della domanda ci scherza su, "Difficile?")

- E le altre proprietà? Complementazione? Intersezione? Ripetizione sono chiusi i linguaggi liberi?

(Fino ad ora il candidato aveva risposto unione e concatenazione correttamente)

- Mi dia (scrivere su foglio) la sintassi delle espressioni regolari

(Chiede comunque precisione, per esempio che A^* contenga tutti i possibili simboli dell'alfabeto)

- Come associa un linguaggio a ciascuno di questi simboli un'espressione regolare?

> funzione $L : \text{Exp-Reg} \rightarrow P(A^*) \dots$

● Candidato 6: (entra con voto alto)

- Mostri come si può associare ad ogni espressione regolare un NFA.

(Il prof fornisce l'inizio della dimostrazione, fa il primo punto ma poi cambia domanda)

- I linguaggi regolari sono inclusi nei linguaggi liberi deterministici? Perché?

- Quando un linguaggio è regolare?

> quando può essere descritto da un'espressione regolare o quando può essere riconosciuto da un NFA

(Domanda fatta per aiutare il candidato a rispondere alla precedente)

- I linguaggi LL(1) sono linguaggi liberi deterministici?

- I linguaggi regolari sono LL(1)?

> Sì

- Saprebbe dimostrarmelo? (Dimostrazione non fatta dal candidato)

- Fare un esempio di linguaggio libero non deterministico

- Dimostri che questo linguaggio è libero

(Il candidato risponde con Pumping Theorem, ma in realtà non va bene, serve proprio per dimostrare che un linguaggio NON è libero)

> costruisco un PDA o costruisco la grammatica relativa, ti fornisce un esempio (lascia libera scelta su quale scegliere)

- La grammatica (il candidato ha scelto di costruire la grammatica) che ha costruito è regolare? È libera? Perché?

- Come è fatta una grammatica dipendente da contesto?

> vedere classificazione di Chomsky

- Invece come sono fatte le grammatiche monotone?

> vedere classificazione di Chomsky

- Sono più potenti dal punto di vista dell'espressività le grammatiche libere da contesto o le grammatiche monotone?

> quelle monotone

- I linguaggi generati da grammatiche libere da contesto e quelli generati dalle grammatiche monotone sono della stessa classe o di classi separate

> sono della stessa classe

- Che tipo di automa serve per riconoscere grammatiche dipendenti da contesto?

(Il prof dice che l'ha appena menzionato a lezione)

- Com'è fatto un Parser di tipo Top down? Mi fa vedere un esempio, uno qualunque?

Anzi, mi faccia un Parser Top down per questa grammatica (il prof scrive un esempio sul foglio e lascia il tempo di calcolarla)

- Data la tabella di parsing LL(1) che lei (il candidato) ha costruito per questo esempio (lo stesso precedente), data la stringa "aa", come la riconosco?

(Mostrare che consultando la tabella di parsing si riconosce)

- Ultima domanda che non ho sentito, comunque era teorica

• Candidato 7:

- Mi dia la Semantica Operazione strutturata IS di e_0 and e_1
(assiomi e regole di inferenza)
- Quali sono i costrutti del linguaggio che abbiamo usato a lezione
 - > sum, sub, ... per le espressioni aritmetiche, per quelle booleane invece eq, or, ..., per i comandi invece ...(Gli sta elencando solo i nomi dei costrutti, senza scriverli e descriverli)
- Mi saprebbe dire quando un simbolo appartiene ad un $\text{First}(\alpha)$ di una certa stringa α ?
(Il prof scrive qualcosa sul foglio: quando un certo simbolo terminale appartiene al $\text{First}(\alpha)$ e chiede che cosa significa)
- e quando un certo simbolo appartiene al $\text{Follow}(A)$ dove A è un simbolo nonterminale?
- Perché abbiamo studiato i First e i Follow?
- Ci interessano i First e i Follow nel parser LR(1)? Mentre per SRL(1)?
- I linguaggi regolari sono inclusi nella classe dei linguaggi LL(0)?
(Il prof fornisce un esempio e chiede se il linguaggio è LL(0)?)

• Candidato 8:

- Scriva (cede il foglio) il linguaggio associato ad ogni espressione regolare
 - > fornisce sintassi $r := \mid \epsilon \mid a \mid r|r \mid r \cdot r \mid r^*$e poi la semantica:
 - linguaggio associato all'insieme vuoto
 - linguaggio associato ad epsilon
 - linguaggio associato ad a
 - linguaggio associato ad r_1 concatenato r_2
 - ...(Vuole che sei molto preciso anche nelle parentesi)
- Fammi un esempio di un linguaggio regolare
 - > Il candidato sceglie a^*
- Che linguaggio è quello associato a a^* (scritto come insieme specifica il prof)
 - > $L[a^*] = \{ a^n \mid n \geq 0 \}$

- Questo linguaggio è regolare (il prof scrive un altro linguaggio semplice sul foglio)? Come faccio a dimostrarlo?

> Si può provare a costruire l'NFA che lo riconosce?

- Come è strutturato un compilatore, cioè in quali fasi è articolato?

- Cos'è l'analisi semantica e cosa fa? (sempre parlando delle fasi del compilatore)

- Che differenza c'è tra semantica statica e semantica dinamica?

● Candidato 9: (entra con un voto alto)

- I linguaggi regolari sono inclusi nei linguaggi LL(1)?

- I linguaggi regolari sono inclusi negli LR(0)?

> No

- Che proprietà hanno? (aggiunge, sono sicuramente LL(0) ma hanno una proprietà particolare, quale?)

- Definizione di un DFA?

- DFA e NFA sono equivalenti? Perché?

> Per ogni NFA posso costruire un DFA equivalente

$DFA \subset NFA$

- La costruzione per sottoinsiemi da cosa parte e come inizia?

- Cos'è la semantica dinamica? Quando l'abbiamo studiata?

- Cos'è la semantica statica? Cosa controlla?

> controlla i vincoli sintattici contestuali (cioè un controllo sul testo)
ecc... (questa era discorsiva)

- Cosa dice la tesi di Church Turing? Perché non è dimostrabile?

DOMANDE GIALLORENZO: MODULO 3



Domande Giallorenzo

Spazia molto sulle risposte che gli dai, quindi è solito rimanere su un solo argomento per persona, a meno che non si chieda la domanda extra per alzare il

voto

1. Definisci un tipo ricorsivo prodotto. Quale potrebbe essere un suo abitante/valore contenuto?
2. Differenza tra definizione estensionale e intensionale? Come si combina col sub-typing?
3. Coercizione e casting, cosa sono e differenze
4. Conversioni canoniche e casting, confronto con estensionale e intensionale
5. Sottotipaggio e definizione degli enum
6. Parlami di inferred typing, manifest e algoritmi di inferenza
7. Quali sono i tipaggi di un linguaggio (nominale, strutturale, duck typing)
8. Descrivimi l'algoritmo d'unione e il suo funzionamento
9. Definizione e utilizzo di oggetti. È necessario il polimorfismo per avere gli oggetti? (no)
10. Elenca i tipi di polimorfismo
11. Controvarianza e covarianza, quando le troviamo e come mai?
12. Sub-typing, cos'è?
13. Classi, oggetti, interfacce e prototipi, differenze e caratteristiche
14. Che tipaggio usa Java?
15. Come funzionano equivalenza e comparazione nel typing nominale?

Domande Giallorenzo 1

16. Polimorfismo di sovraccarico e indirizzamento dinamico, quali sono le differenze?
17. I 2 metodi di implementazione del dispatch dinamico, qual è il più efficiente e quello meno?
18. In una classe, dove vivono i metodi e i campi e come vengono memorizzati?
19. Parlami del problema della classe fragile
20. Garbage collection, quali implementazioni abbiamo visto e come funzionano?
21. Garbage collector con il costo più basso di memoria?

22. Spiega i problemi di un garbage collector stop-the-world. In quali tipi di applicazioni potrebbero creare dei problemi? (applicazioni real-time)
23. Alternative per applicazioni real-time?
24. Limiti del contatore dei puntatori? (liste cicliche)
25. Parlami delle eccezioni, in particolare come sono gestite in Java
26. Ci sono punti di collegamento / similitudini tra eccezioni e tipaggio? (tipaggio nominativo)
27. Pilastri della OOP, espandere e spiegarli tutti e 4 singolarmente (occhio, lui vuole il Dynamic Dispatch al posto del Polimorfismo)
28. Come si fa in memoria il Dynamic Dispatch? (come si memorizzano vtable e funzioni)
29. Come funziona la catena dinamica (forse ho sentito male, non ho presente) sulle superclassi?
30. Differenza tra row-major e col-major, perché preferire un tipo rispetto ad un altro? (caching)
31. Array in relazione a weakly-typed e C
32. Linguaggi strongly-typed? Rust vs C con i puntatori
33. A cosa servono i sistemi di tipi?

Domande Giallorenzo 2

34. Come funziona il sistema di tipi di JavaScript e Python? (linguaggi dinamici in generale)
35. Strongly vs weakly-typed, quali proprietà hanno e in cosa differiscono?
36. Controlli statici sui tipi contro i controlli dinamici. (piccola parentesi anche sull'alting problem)
37. Ereditarietà, singola e multipla
38. Definisci i tipi di dati astratti
39. Differenze tra abstract datatype e oggetti