

1. Enunciato e dimostrazione del pumping lemma.
2. Quando una grammatica e' di classe LL(1)
  - Quando la tabella di parsing LL(1) non contiene conflitti
  - Oppure con il teorema
3. Le grammatiche di classe LL(1) generano linguaggi deterministici?
  - Si in quanto un linguaggio e' libero deterministico sse e' accettato da una grammatica LR(k) per qualunque  $k \geq 0$
  - Un linguaggio e' libero deterministico sse e' generato da una grammatica SLR(1)
  - E noi sappiamo che ogni linguaggio LL(k) con  $k \geq 0$  e' strettamente incluso nei linguaggi generati da grammatiche SLR(1)
  - Da cui le grammatiche di classe LL(1) generano linguaggi deterministici
4. Le differenze tra le tecniche di parsing top-down e bottom-up
  - le tecniche top-down generano un albero di derivazione canonica sinistra, e partono a verificare se la stringa fa parte della grammatica partendo dal simbolo iniziale della grammatica, e verifica quindi se a partire da esso in una serie di passi di derivazione possiamo riprodurre la stringa in questione
  - le tecniche bottom-up generano un albero di derivazione canonica destra, e si parte dalla stringa in questione, a ritroso cercando quale produzione si e' utilizzata per generare la stringa fino a raggiungere il simbolo iniziale S
5. Definizione di un DPDA
  - un DPDA e' formato da :
    - insieme degli stati
    - insieme dei simboli in input
    - insieme dei simboli sulla pila
    - funzione di transizione
    - stato iniziale
    - bottom (simbolo iniziale della pila)
    - insieme degli stati finali
6. Gli automi a pila deterministici a quale classe di linguaggi corrispondono
  - Ai linguaggi liberi deterministici
7. Esempio di un linguaggio libero non deterministico
  - $L = \{ww^R\}$
8. Le grammatiche libere che linguaggi generano
  - Linguaggi liberi
9. Cos'e' un compilatore
  - un compilatore e' un componente che e' in grado di convertire un programma scritto di un linguaggio  $L'$  in un linguaggio scritto in  $L$
  - Esege una funzione del tipo  $C^L(L',L') : P^L' \rightarrow P^L$
10. Definizione di DFA
  - DFA
    - insieme degli stati Q
    - insieme dei simboli dell'input E

- funzione di transizione :  $G : Q \times E \rightarrow Q$
  - stato iniziale
  - insieme degli stati finali
11. Come e' strutturato nelle sue fasi un compilatore
    - Analizzatore lessicale
    - Analizzatore sintattico
    - Analizzatore semantico
    - Codice intermedio
    - Ottimizzazione
    - Codice oggetto/finale
  12. Che tipo di strumenti si possono utilizzare per generare analizzatori lessicali
    - Lex
  13. Struttura di un file .lex
    - Prologo
    - Regole
    - Funzioni ausiliarie
  14. Cos'e' un item LR(0)
    - E' una produzione di una grammatica con aggiunta di un punto che indica quanto parte della produzione e' stata' gia' esaminata dal parser
  15. Cosa si intende con turing-completezza
    - Un formalismo si dice turing completo se e' espressivo almeno tanto quanto una macchina di turing
  16. Qual e' il teorema legato alla turing completezza
    - Un linguaggi di programmazione si dice turing completo se
      - ha un'operazione sequenziale
      - ha un operatore condizione
      - ha un operatore di ripetizione indeterminata
      - ha un operatore di assegnamento
  17. Cosa dice la prefix property e perche' e' interessante per certi argomenti
    - La prefix property dice che non esistono x,y in linguaggio tali che x sia prefisso di y
    - Un linguaggio libero deterministico e' accettato da una DPDA per pila vuota se gode della prefix property
  18. Differenze tra item LR(0) e item LR(1)
    - Item LR(1) e' formato da una coppia (item LR(0), lookahead)
  19. I linguaggi regolare sono chiusi per unione ?
    - Si, in quanto i linguaggi regolari sono chiusi sia per intersezione che per complementazione
  20. Definizione di grammatica libera
    - $S \rightarrow \epsilon$ ,  $A \rightarrow a$  con  $a$  in  $(NT \cup T \cup \epsilon)^*$  e  $A$  in  $NT$
  21. Gerarchia di Chomsky
    - Grammatiche regolare,  $S \rightarrow \epsilon$ ,  $A \rightarrow a$ ,  $A \rightarrow aB$
    - Grammatiche libere,  $S \rightarrow \epsilon$ ,  $A \rightarrow a$  con  $a$  in  $(NT \cup T)^*$
    - Grammatiche dipendenti,  $aBb \rightarrow acb$  con  $a,b$  in  $(NT \cup T)^*$ ,  $B$  in  $NT$

- Grammatiche monotone,  $A \rightarrow B$  con  $|A| \leq |B|$
  - Grammatiche generali,  $A \rightarrow B$  senza vincoli
22. Che cosa è un interprete
- È un componente che permette di eseguire istruzioni in un linguaggio L' su una macchina scritta in linguaggio L
23. Cosa è un albero di derivazione
24. Le grammatiche ambigue sono un problema per i parser?
- I parser non sono generabili da una grammatica ambigua
25. Cosa dice il pumping theorem per il linguaggi liberi
26. Come sono classificati i linguaggi LL e LR rispetto k
- $LL(0) < LL(1) < \dots < LL(k) < SLR(1)$
27. Quando un linguaggio si dice di classe x ( LL(1), LR(0), etc... )
- Quando c'è una grammatica di class x che lo genera
28. Parlare di YACC
29. Come è definito un NFA
- NFA
    - insieme degli stati
    - insieme dei simboli dell'alfabeto
    - funzione di transizione :  $\gamma : Q \times E \rightarrow P(Q)$
    - stato iniziale
    - insieme degli stati finali
30. Il riconoscimento per pila e vuota sono equivalenti?
- non sono equivalenti dato lo stesso PDA, ma in generale per uno stesso linguaggio possiamo sviluppare sia un PDA per stato finale che per per pila vuota
31. I parser LR,LL sono dei PDA per pila vuota?
- si
32. Sintassi e semantica delle espressioni regolari
- $r ::= \text{vuoto} \mid \text{eps} \mid a \mid rr \mid r|r \mid r^*$
  - $L[\text{vuoto}] = \text{vuoto}$
  - $L[\text{eps}] = \{\text{eps}\}, L[a] = \{a\}$
  - $L[rr] = L[r]L[r]$
  - $L[r|r] = L[r] \cup L[r]$
  - $L[r^*] = L[r]^*$
33. Cosa cambia tra riempire una tabella LR(0) e SLR(1)
- Quando l'item è una maniglia, ovvero si tratta di una reduce in LR(0), la reduce va messa in ogni  $T \cup \{\text{eps}\}$ , mentre in SLR(1) si mette solo nei follow della testa della produzione
34. Cosa si intende per semantica statica, differenze con semantica dinamica
- Semantica statica sono regole di semantica verificabili a compile time
35. Un compilatore esegue semantica statica o dinamica, e in quale fase?
- Semantica statica, dopo l'analisi sintattica
36. Come si trasforma un NFA in un DFA e se è sempre possibile
- Con algoritmo di costruzione per sottoinsiemi
37. Come si possono eliminare le produzioni epsilon da una grammatica libera

- Definisco  $N$  come l'insieme dei simboli che presentano produzioni epsilon o che in qualche passo di derivazione sono epsilon
  - Quindi per ogni produzione che contiene uno o piu' simboli in  $N$ , allora definisco  $F =$  come l'insieme della parti di tali simboli
  - Per ogni elemento di  $F$  aggiungo una produzione uguale alla originale a meno di dell'elemento, se rimane solo epsilon no
38. Cosa e' un analizzatore lessicale
- E' un componente che preso in input una stringa genera una lista di token (classe, valore), ovvero verificare che le parole presenti nella stringa appartengano al linguaggio
39. Definire grammatica dipendente dal contesto
- $aBb \rightarrow awb$ , con  $a, b$  in  $(NT \cup T)^*$  e  $B$  in  $NT$
40. Quando due stati di un DFA sono equivalenti
- $q1, q2$  sono equivalenti se  $L[M, q1] = L[M, q2]$  con  $M$  DFA
  - oppure come per ogni  $a \rightarrow \delta(q1, a) = \delta(q2, a)$  con un DFA minimo
41. Spiegare brevemente come funziona l'algoritmo a scala
42. Una volta individuati gli stati equivalenti si puo' sviluppare una automa minimo?
- si, gli stati a coppie non marcati risultano essere equivalenti
43. Parlare del parser LALR(1)
- LA - lookahead
  - L - left to right
  - R - rightmost
  - (1) un lookahead
  - sono simili ai parser LR(1) in cui pero' gli stati con stesso nucleo sono
  - fusi
44. Cosa dice il teorema di Jacopini Bohom
- Un linguaggio di programmazione si dice turing completo se
    - ha un operatore di comando sequenziale
    - ha un operatore di condizione
    - ha un operatore di ripetizione indeterminata
    - ha un operatore di assegnamento
45. Quando e' che due DFA sono equivalenti
- $M1$  e  $M2$ , DFA, sono equivalenti se riconoscono lo stesso linguaggio
  - $L[M1] = L[M2]$
46. Cosa si intende per bootstrapping
- E' una tecnica per implementare un compilatore
47. Definizione di grammatica libera
- $S \rightarrow \epsilon$ ,  $A \rightarrow a$  con  $a$  in  $(NT \cup T \cup \epsilon)^*$  e  $A$  in  $NT$
48. Definizione di grammatica dipendente da contesto
- $aBb \rightarrow awb$ , con  $a, b$  in  $(NT \cup T)^*$  e  $B$  in  $NT$  e  $w$  in  $(NT \cup T)^*$
49. I linguaggi regolari sono tutti quanti di classe LL(1)
50. Cosa si intende per vincoli contestuali
- I vincoli contestuali sono controlli relativi alla semantica statica di un linguaggio

51. Cosa e' un albero di sintassi astratta
- E' la rappresentazione della struttura sintattica di una stringa riconosciuta da un parser, ed e' ottenibile da un albero di sintassi concreta rimuovendo lo zucchero sintattico
52. Cosa dice il problema della fermata
- stabilire se un programma puo' definire in tempo finito se dato un certo programma e un certo input si puo dimostrare che termina
  - $H(P,x) = \{ 1 \text{ se } P(x) \text{ termina, } 0 \text{ se } P(x) \text{ non termina } \}$
  - $K(G) = \{ 1 \text{ se } H(G,G) == 1, 0 \text{ se } H(G,G) == 0 \}$
  - $G(M) = \{ 1 \text{ se } K(M) == 0, \text{ diverge se } K(M) == 1 \}$
  - $G(G) = \{ 1 \text{ se } K(G) == 0 \Rightarrow P(G) == \text{non termina}, \text{ diverge se } K(G) == 1 \Rightarrow H(G,G) == 1 \Rightarrow G(G) \text{ termina} \}$
53. Come e' definito il First(a)
- si dice b first di a se  $a \Rightarrow^* bC$
  - eps e' first di a se a e' annullabile
54. Come e' definito il Follow(a)
- si dice b follow di a se  $S \Rightarrow^* abC$
  - \$ e' follow di a se  $S \Rightarrow^* ba$