6 Automi a pila non deterministici

lunedì 4 novembre 2024 12:21

06-automi-a-pila-18

Automata, Languages, and Computing

Automi a pila non deterministici

1

1

Automi a pila non...

Ripasso delle varie forme (chompsky e greibach)

Registrazione audio avviata: 14:06 lunedì 4 novembre 2024

Automi a pila non...

Non esiste un automa a stati finiti in grado di riconoscere a^nb^b.
Abbiamo bisogno quindi di un altro automa: AUTOMI A PILA.
Qui il non determinismo è cruciale: gli automi a pila deterministici e automi a pila non deterministici NON sono equivalenti

Registrazione audio avviata: 14:09 lunedì 4 novembre 2024

Automi a pila

- introduciamo un nuovo modello di calcolo: l'automa a pila non deterministico

$M=\langle \Sigma,\Gamma,Z_0,Q,q_0,F,\delta \rangle$

Σ alfabeto di input
Γ alfabeto dei simboli della pila
$Z_0 \in \Gamma$ simbolo di pila iniziale
Q insieme finito di stati
$q_0 \in Q$ stato iniziale
$F \subseteq Q$ insieme di stati finali
$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow P(Q \times \boxed{\Gamma^*})$

funzione di transizione: a partire dallo stato interno attuale, dal carattere letto sul nastro e dal simbolo affiorante sulla pila, sostituisce il simbolo affiorante sulla pila con una stringa di simboli e si porta in un nuovo stato interno

2

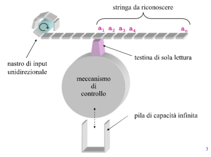
2

Cose nuove:
- Alfabeto di simboli della pila-> nella pila ci finiscono termini di questo alfabeto
- Simbolo di pila iniziale -> Z0

- Funzione di transizione-> compare epsilon: in input l'automa può prendere o il carattere che ha di fronte oppure non fare niente (epsilon).
La transazione dipende anche in funzione di un elemento di gamma (simbolo degli elementi di pila), è il simbolo affiorante nella pila.
Nella pila ci posso mettere un simbolo, tanti simboli oppure nessuno (per questo è gamma asteriscato).

06-automi-a-pila-18

Automa a pila



3

3

Ho una pila in cui inserisco/cancello oggetti. Ha capacità infinita.
Il nastro è unidirezionale e quello che abbiamo sopra è la stringa che vogliamo riconoscere, cioè vedere se la stringa appartiene o no al linguaggio.

Come faccio a cancellare caratteri?
Gamma asteriscato comprende la stringa vuota, quindi se il simbolo affiorante nella pila se lo sostituisco con la stringa vuota è come se lo stessi cancellando

Un passo dell'automa a pila

- se un automa a pila M in uno stato q legge **a** dal nastro e C affiora sulla pila una possibile transizione è

$\delta(q,\mathbf{a},C) = \{\langle q',\varepsilon\rangle,\langle q'',BA\rangle\}$

  - l'automa si porta nello stato q' mettendo ε nella pila e nello stato q'' mettendo BA nella pila
- convenzioni:
  - se metto BA in pila, B è affiorante
  - mettere ε in pila equivale a cancellare l'elemento affiorante

4

4

Va contemporaneamente su q' e q'':
Su q' mette la stringa vuota
Su q'' mette in pila BA

Stringa palindroma con un numero dispari di caratteri, il carattere centrale è una c. w e w sono sul l'alfabeto {0,1}.
È un linguaggio regolare? No, dimostra che non sia regolare -> PL.
Per assurdo prendo una stringa 011c110. prendo i primi n caratteri e vedo che si crea un bilanciamento.
È un linguaggio context free? Si -> creo una grammatica.

06-automi-a-pila-18

Esempio di automa a pila

- automa a pila M che riconosce

$L = \{wc\underline{w} \mid w \in (0+1)^*\}$

- una possibile strategia
  - uso due simboli (B=blu e G=green) per ricordare le occorrenze di 0 e 1 in w
  - nello stato q_1 memorizzo w, nello stato q_2 confronto $\underline{w}$ con i simboli memorizzati
  - l'automa M in questo esempio ha un comportamento deterministico

5

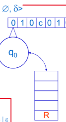
5

Automa a pila M che riconosce $\{wc\underline{w} \mid w \in (0+1)^*\}$ lettura di 010c010

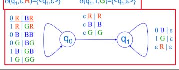
$M=\langle\{0,1,c\}, \{R,B,G\}, R, \{q_0,q_1\}, q_0, \varnothing, \delta\rangle$

Rapp tabellare

$\delta(q_0,0,R)=\{\langle q_0,BR\rangle\}$ $\delta(q_0,0,B)=\{\langle q_0,BB\rangle\}$
$\delta(q_0,0,G)=\{\langle q_0,BG\rangle\}$ $\delta(q_0,c,R)=\{\langle q_1,R\rangle\}$
$\delta(q_0,c,B)=\{\langle q_1,B\rangle\}$ $\delta(q_0,c,G)=\{\langle q_1,G\rangle\}$
$\delta(q_0,1,R)=\{\langle q_0,GR\rangle\}$ $\delta(q_0,1,B)=\{\langle q_0,GB\rangle\}$
$\delta(q_0,1,G)=\{\langle q_0,GG\rangle\}$ $\delta(q_1,0,B)=\{\langle q_1,\varepsilon\rangle\}$
$\delta(q_1,\varepsilon,R)=\{\langle q_1,\varepsilon\rangle\}$ $\delta(q_1,1,G)=\{\langle q_1,\varepsilon\rangle\}$



graficamente



6

6

Q0-> st iniziale

"Se sei nello stato q0, leggi uno 0 e ho R come simbolo affiorante della pila fai BR (r rimane dove sta e ci metti B in pila) e rimani dove stai "

[15.00]

Sigma -> {0,1,c}
Simboli di pila -> {R,B,C}
R-> simbolo di pila all'istante iniziale
Stati -> {q0,q1}
Q0-> stato iniziale
Insieme vuoto -> insieme degli stati finali vuoto. Se è vuoto che vuol dire? Per questi automi abbiamo 2 possibili scelte di riconoscimento pe runa stringa:
1) La stringa è riconosciuta quando l'hai letta tutta e hai lasciato la pila vuota; -> riconoscimento per pila vuota
2) Puoi metterci stati finali e quando l'hai letta tutta (la stringa) verifichi che la computazione sia finita in uno stato finale

Automi a pila non...

Registrazione audio avviata: 14:31 lunedì 4 novembre 2024

Quando arriva c capisci che la pila che hai riempito la devi svuotare. E se svuotandola trovo la corrispondenza con i caratteri, alla fine avrò la pila vuota e la stringa sarà riconosciuta.

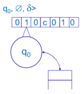
Ho R in pila-> simbolo di pila vuota. Sopra ci metti B (vedi fdt)

06-automi-a-pila-18

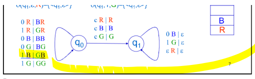
Automa a pila M che riconosce $\{wc\underline{w} \mid w \in (0+1)^*\}$ lettura di 010c010

$M=\langle\{0,1,c\}, \{R,B,G\}, R, \{q_0,q_1\}, q_0, \varnothing, \delta\rangle$

$\delta(q_0,0,R)=\{\langle q_0,BR\rangle\}$ $\delta(q_0,0,B)=\{\langle q_0,BB\rangle\}$
$\delta(q_0,0,G)=\{\langle q_0,BG\rangle\}$ $\delta(q_0,c,R)=\{\langle q_1,R\rangle\}$
$\delta(q_0,c,B)=\{\langle q_1,B\rangle\}$ $\delta(q_0,c,G)=\{\langle q_1,G\rangle\}$
$\delta(q_0,1,R)=\{\langle q_0,GR\rangle\}$ $\delta(q_0,1,B)=\{\langle q_0,GB\rangle\}$
$\delta(q_0,1,G)=\{\langle q_0,GG\rangle\}$ $\delta(q_1,0,B)=\{\langle q_1,\varepsilon\rangle\}$

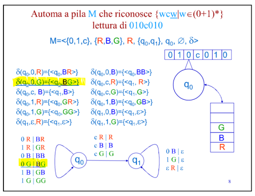


Sono ancora in q0, leggo il secondo elemento della stringa (1): mi ricordo attraverso la pila che sto leggendo un 1 e ci metto sopra G (= ho letto un 1)



7

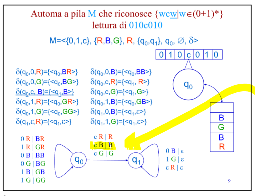
quindi se sono in q0 e leggo 1 e l'elemento affiorante della pila è B metto G in pila. Non tocco B



La pila mi ricorda che ho letto uno 0 (B) e un 1(G). Leggo un altro 0 e B finisce in pila di nuovo (vedi sotto)

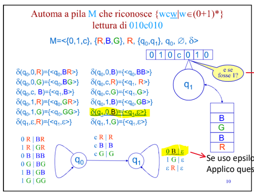
8

06-automi-a-pila-18



Ho caricato nella pila tutti i simboli che matchano gli elementi di w. Se trovo una c? leggo una c, ho una B in pila e ce la lascio, pronto per vedere se fa matching con w

9

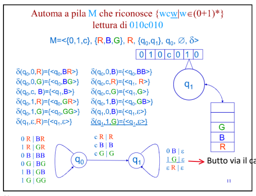


Non c'è una parte della fdt che posso usare nella rappresentazione tabellare.
La fdt non è deterministica, se non specifico nulla si va in un insieme vuoto di stati.
Qui la computazione si ferma, **non c'è modo di riconoscere la stringa**, essa è sicuramente non palindroma.
La pila non è vuota, la computazione si è fermata --> allora non è stata riconosciuta perché la pila non è vuota alla fine della transazione (per convenzione)

Se uso epsilon-> cancello un elemento dalla pila
Applico questa fdt. Butto la B (la cancello perché il matching è andato a buon fine)

10

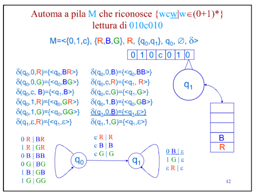
06-automi-a-pila-18



SE LEGGI 0 METTI IN PILA B, SE LEGGI 1 METTI IN PILA G

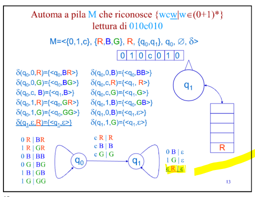
Butto via il carattere G, corrisponde al carattere 1.

11



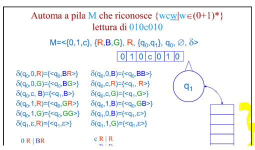
12

06-automi-a-pila-18

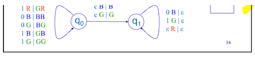


Qui la stringa l'ho letta tutta. Che succede? Posso fare transizione su epsilon.
La stringa è finita, l'ho letta tutta, butto R: riconoscimento per pila vuota

13



OSSERVAZIONI:
Perché l'automa a pila è definito in modo tale che all'inizio ci sia Z0? Quindi con un primo elemento nella pila?
Se buttiamo il simbolo di pila vuota, capiamo che abbiamo finito la lettura della stringa.
La fdt per essere specificata ha bisogno di un simbolo nella pila, se non c'è un simbolo iniziale non posso specificare la fdt corrispondente. Quindi all'inizio un simbolo inziale, Z0, mi serve.



14

Bisogna inoltre specificare tutta la fdt, anche in maniera prolissa. Non si può diminuire. [13.00]

11/2/2024

Automi a pila non...

06-automi-a-pila-18

Registrazione audio avviata: 14:50 lunedì 4 novembre 2024

Automi a pila – computazione

- configurazione di automa a pila: tripla $\langle q,x,\gamma\rangle$ con $q\in Q$ stato interno, $x\in\Sigma^*$ stringa da leggere in input e $\gamma\in\Gamma^*$ stringa attualmente in pila
- relazione di transizione per automa a pila: relazione binaria sulle configurazioni $\vdash$
  $$\langle q,x,\gamma\rangle \vdash \langle q',x',\gamma'\rangle$$
  se e solo se
  $$((x=ax' \wedge \gamma=Z\eta \wedge \gamma'=\zeta\eta \wedge \langle q',\zeta\rangle\in\delta(q,a,Z)) \vee (x=x' \wedge \gamma=Z\eta \wedge \gamma'=\zeta\eta \wedge \langle q',\zeta\rangle\in\delta(q,\varepsilon,Z)))$$
- computazione per automa a pila: chiusura transitiva e riflessiva di $\vdash$, indicata con $\vdash^*$

15

15

CONFIGURAZIONE di un automa a pila: è una "foto" che contiene lo stato corrente, la pila (tutta) e la porzione della stringa di input ancora da leggere (non ancora processata). In che situazione l'automa a pila si trova in un certo istante

RELAZIONE di transizione: relazione binaria sulle configurazioni:
Passi da una configurazione ad un'altra se la fdt te lo consente
Se e soltanto se:

- Dice quando puoi passare da una configurazione all'altra:
  Passi da una configurazione ad un'altra se la fdt te lo consente.

COMPUTAZIONE PER AUTOMI A PILA

Linguaggio riconosciuto

- due definizioni alternative
  - accettazione per pila vuota
    - una stringa è accettata da un automa a pila M se e solo se al termine della scansione della stringa la pila è vuota
      $N(M) = \{x \mid \langle q_0,x,Z_0\rangle \vdash^* \langle q,\varepsilon,\varepsilon\rangle\}$
  - accettazione per stato finale
    - una stringa è accettata da un automa a pila M se e solo se al termine della scansione della stringa M si trova in uno stato finale
      $L(M) = \{x \mid \langle q_0,x,Z_0\rangle \vdash^* \langle q,\varepsilon,\gamma\rangle, q\in F, \gamma\in\Gamma^*\}$
- si può dimostrare che le due definizioni sono equivalenti

16

16

Una stringa è riconosciuta dall'automa quando: una volta definita la relazione binaria di transizione e la sua computazione:

- Riconoscimento **per pila vuota**:
  Quando parto: da q0, ho una stringa e Z0. la computazione dopo diversi passi ho: uno stato qualunque, epsilon (la stringa è stata letta tutta), in pila epsilon
- Riconoscimento **per stato finale**:
  Alla fine posso avere anche la pila "sporca", ma al termine sono in uno stato finale

11/2/2024

Automi a pila non...

06-automi-a-pila-18

Registrazione audio avviata: 15:00 lunedì 4 novembre 2024

Esempio di riconoscimento per pila vuota

- Automa a pila che riconosce ww con w in (0+1)*

$M=\langle\{0,1\},\{R,B,G\},R,\{q_0,q_1\},q_0,\varnothing,\delta\rangle$

$\delta(q_0,0,R)=\{\langle q_0,BR\rangle\}$
$\delta(q_0,1,R)=\{\langle q_0,GR\rangle\}$
$\delta(q_0,\varepsilon,R)=\{\langle q_0,\varepsilon\rangle\}$

Non determinismo

$\delta(q_0,0,B)=\{\langle q_0,BB\rangle,\langle q_1,\varepsilon\rangle\}$
$\delta(q_0,0,G)=\{\langle q_0,BG\rangle\}$
$\delta(q_0,1,B)=\{\langle q_0,GB\rangle\}$
$\delta(q_0,1,G)=\{\langle q_0,GG\rangle,\langle q_1,\varepsilon\rangle\}$

$\delta(q_1,0,B)=\{\langle q_1,\varepsilon\rangle\}$
$\delta(q_1,1,G)=\{\langle q_1,\varepsilon\rangle\}$

$\delta(q_1,\varepsilon,R)=\{\langle q_1,\varepsilon\rangle\}$

17

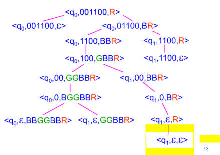
17 Svuota definitivamente la pila

Linguaggio delle stringhe palindrome di lunghezza pari. È non contestuale? Si.
È regolare? No.
È riconoscimento per pila vuota come prima. La difficoltà dove è? Leggo la stringa, ma dove finisce la prima parte della stringa e inizia la prima parte della seconda? Sfrutto il non determinismo: quando-> arrivo ad un carattere contemporaneamente penso:

- Sono ancora nella prima parte della stringa
- Sta iniziando la seconda parte della stringa

Prime due righe-> carico la pila. Terza riga butto il carattere R.

Computazione dell'automa su 001100



18

18

Un nodo è una determinata configurazione dell'automa a pila non deterministico

Accettazione. Perché la stringa sia riconosciuta, mi basta che una delle foglie corrisponda all'accettazione per pila vuota

11/2/2024

Automi a pila non...

06-automi-a-pila-18

Registrazione audio avviata: 15:26 lunedì 4 novembre 2024

Automi a pila deterministici

$M=\langle\Sigma,\Gamma,Z_0,Q,q_0,F,\delta\rangle$ è deterministico se
$\forall\sigma\in\Sigma, \forall Z\in\Gamma, \forall q\in Q \quad |\delta(q,\sigma,Z)|+|\delta(q,\varepsilon,Z)|\leq 1$

la condizione impone che

1. $|\delta(q,\sigma,Z)|$ non sia superiore ad 1
2. se è definita una ε-transizione per un certo stato q e per un certo simbolo di pila Z, non deve essere definita un'altra transizione per q, Z

osservazione: la 2. impone che se M riconosce ε e un'altra stringa, allora è non deterministico

osservazione: si può ovviare a questa limitazione chiudendo tutte le strighe con un carattere speciale "$"

19

19

OSS:
Automi a pila deterministici e non deterministici non riconoscono gli stessi linguaggi. Non useremo i deterministici

wcw non è context free: con il PL lo posso dimostrare:
Se il PL non vale -> non è context free.
Stringa che prendiamo in considerazione:
$0^n1^nc0^n1^n$

DUE DOMANDE A INIZIO LEZIONE

1) La grammatica G è in forma normale di chomsky (G in CNF): "quanti passi per derivare una stringa di sigma* ?" per passo intendiamo una derivazione di una produzione. Ho una stringa x di n caratteri . |x| = n . Quanti passi mi servono se sono in CNF? A DX ho un terminale o 2 non terminali. Le forme di frase possono essere fatte come mi pare. Supponiamo di produrre i terminali solo alla fine quando non posso più produrre non terminali.
Es:
S->AB->ABC->ABCD-> . . . -> aBCD. . .Z-> abCD...Z -> . . . . -> abcd.....z
Per un lungo periodo le forme di frasi sono composte solo da sequenze di non terminali.
Posso applicare le produzioni in modo che siano rispettate queste produzioni, ovvero prima tutti i non terminali e poi i terminali li metto alla fine uno per volta. Ogni volta all'inizio aumento di un non terminale (levo uno e aggiungo 2). Allora di quanti non terminali ho bisogno? Di "n" non terminali, ma partendo da 1 non terminali ho bisogno di n-1 step. Per poi produrre i terminali ho bisogno di n passi. Quindi ho bisogno in totale di 2n-1 passi (n-1+n).

2) G IN GNF (FORMA NORMALE DI GRAIBACH): QUANTI PASSI PER DERIVARE UNA STRINGA DI SIGMA ASTERISCATO? |x|=n.
Che tipo di struttura esce fuori?
Es:
(mi forzo a sostituire sempre il primo non terminale)
(se faccio l'hp di sostituire sempre il non terminale più a sinistra, come sono fatte le forme di frase? Tutti i terminali a sx e i non terminali a dx)
Ogni volta che applico una produzione aggiungo esattamente 1 terminale. -> utilizzo esattamente "n" produzioni
S-> aBCD -> abXYCD -> ...

Per derivare una stringa di una grammatica regolare anche mi servono esattamente n passi (per derivare stringhe di n caratteri).

Automi a pila e linguaggi CF

- Teorema: se L è un linguaggio non contestuale esiste un automa M tale che L = N(M)
- Dimostrazione
  - supponiamo che $\varepsilon\notin L$
    - in seguito rimuoveremo questo vincolo
  - sia $G=\langle\Sigma,V_N,P,S\rangle$ una grammatica in GNF che genera L
  - costruiamo $M=\langle\Sigma,\Gamma,Z_0,Q,q_0,F,\delta\rangle$ come segue
    $\Gamma=V_N$, $Z_0=S$, $Q=\{q\}$, $q_0=q$, $F=\varnothing$
    per ogni $A\rightarrow a\gamma$ con $\gamma\in V_N^*$ stabiliamo che $\langle q,\gamma\rangle \in \delta(q,a,A)$
    gamma

20

20

Già fatto per gli automi non deterministici e i linguaggi regolari, ma stavolta è per le grammatiche CF.
La G che ho di fronte è in forma normale di graibach, e quindi la sua struttura è simile a quella dei linguaggi regolari del tipo:
S-> aXYZ, metto XYZ li metto in pila e li uso per capire quale pezzo di stringa è generato dopo "a".
S->aXYZ|aZW -> non determinismo

Costruzione:
S : si trasforma in simbolo di pila iniziale Z0;
Q (gli stati non sono così importanti per gli automi a pila non det. Qui ho uno stato solo Q ed è anche iniziale);
F = vuoto : riconoscimento per pila vuota
I non terminali Vn-> sono i simboli di pila usati per il riconoscimento.
Gamma -> Sequenza di non terminali eventualmente vuota

11/2/2024

Automi a pila non...

06-automi-a-pila-18

Registrazione audio avviata: 09:47 mercoledì 6 novembre 2024

Dimostrazione (gr. CF → aut. a pila)

- per dimostrare che

Cosa bisogna dimostrare:
se la G genera x (stringa) allora l'automa lo riconosce e viceversa: se l'automa riconosce una stringa x allora esiste una G che la genera. Se alfa = epsilon, se l'automa riconosce la stringa, anche la grammatica la deriva (e viceversa). Io così sto applicando un

per dimostrare che

$<q,x,S> \vdash^* <q,\varepsilon,\alpha>$ se e solo se $S \Rightarrow^* x\alpha$

- dimostriamo per induzione, su i che

  se $<q,x,S> \vdash_i <q,\varepsilon,\alpha>$ allora $S \Rightarrow_i x\alpha$

- e che

  se $S \Rightarrow_i x\alpha$ allora $<q,x,S> \vdash_i <q,\varepsilon,\alpha>$

21

assunto ancora più forte. Se rimane qualcosa in pila e se c'è qualcosa ancora nella forma di frase (ovvero se alfa non è epsilon) allora anche coincidono l'automa e la grammatica.
Dimostrando questa prima riga dimostro una cosa più forte rispetto a ciò che bisogna dimostrare.
Se una G deriva una stringa: S ->* stringa

Se l'automa deve riconoscere una stringa: ho una configurazione di partenza {q0,x,S} e una di arrivo {q,epsilon,epsilon} (se l'automa riconosce la stringa)
Se alfa = stringa vuota, se l'automa riconosce la stringa anche la G la deriva. E viceversa se la grammatica deriva la stringa, l'automa la riconosce.

21

Automi a pila non...

Registrazione audio avviata: 10:01 mercoledì 6 novembre 2024

Dimostrazione (gr. CF → aut. a pila)

- dimostriamo per induzione su i che

  se $<q,x,S> \vdash_i <q,\varepsilon,\alpha>$ allora $S \Rightarrow_i x\alpha$

- passo base: $i=1$ $(x \in \Sigma)$
  - se $<q,x,S> \vdash_1 <q,\varepsilon,\alpha>$ allora, per costruzione, esiste in P una produzione $S \to x\alpha$

22

Caso base: derivazione fatta di un singolo step.
se l'automa in i passi riconosce quella configurazione, allora la grammatica in i passi genera xα

Passo base con i=1. la fdt mi permette di fare questo solo se esiste una produzione S->xα

22

Dimostrazione (gr. CF → aut. a pila)

- dimostriamo per induzione su i che

  se $<q,x,S> \vdash_i <q,\varepsilon,\alpha>$ allora $S \Rightarrow_i x\alpha$

- passo induttivo: $i>1$ $(x \in \Sigma^*, x=ya, a \in \Sigma)$
  - se $<q,x,S> \vdash_i <q,\varepsilon,\alpha>$, allora $\exists\beta$ tale che

    $<q,ya,S> \vdash_{i-1} <q,a,\beta> \vdash_1 <q,\varepsilon,\alpha>$
  - da cui $<q,y,S> \vdash_{i-1} <q,\varepsilon,\beta>$
    - infatti a non può avere effetto su M prima di essere letto
  - per ipotesi induttiva $S \Rightarrow_{i-1} y\beta$
  - inoltre $<q,a,\beta> \vdash_1 <q,\varepsilon,\alpha>$ implica che in P esista una produzione $A \to a\eta$ con $\beta=A\gamma$ e $\alpha=\eta\gamma$
  - quindi $S \Rightarrow_{i-1} y\beta=yA\gamma \Rightarrow ya\eta\gamma=ya\alpha=x\alpha$

23

Passo induttivo.
Ho preso la computazione di i passi e l'ho divisa in due:
1) Computazione di i-1 passi
2) Computazione di 1 passo finale

Così applichiamo l'hp induttiva.
Nell'ipotesi induttiva derivo in i-1 passi yβ

Da S derivo xα (ciò che volevo dimostrare)

23

Automi a pila non...

Registrazione audio avviata: 10:14 mercoledì 6 novembre 2024

Dimostrazione (gr. CF → aut. a pila)

- dimostriamo per induzione su i che

  se $S \Rightarrow_i x\alpha$ allora $<q,x,S> \vdash_i <q,\varepsilon,\alpha>$

- passo base: $i=1$ $(x \in \Sigma)$
  - se $S \Rightarrow_1 x\alpha$ allora esiste in P una produzione $S \to x\alpha$ e quindi, per costruzione, $<q,x,S> \vdash_1 <q,\varepsilon,\alpha>$

24

Passo base
Se faccio un singolo passo, devo usare una produzione da S->xα allora anche attraverso un automa posso farlo
Per costruzione

24

Dimostrazione (gr. CF → aut. a pila)

- dimostriamo per induzione su i che

  se $S \Rightarrow_i x\alpha$ allora $<q,x,S> \vdash_i <q,\varepsilon,\alpha>$

- passo induttivo: $i>1$ $(x \in \Sigma^*, x=ya, a \in \Sigma)$
  - se $S \Rightarrow_i x\alpha=ya\alpha$ allora esiste $A \to a\eta$ in P tale che

    $S \Rightarrow_{i-1} yA\gamma \Rightarrow ya\eta\gamma$ con $\alpha=\eta\gamma$
  - per ipotesi induttiva $<q,y,S> \vdash_{i-1} <q,\varepsilon,A\gamma>$
  - da cui $<q,ya,S> \vdash_{i-1} <q,a,A\gamma>$
  - ma poiché $A \to a\eta$ allora $<q,\eta> \in \delta(q,a,A)$
  - quindi $<q,ya,S> \vdash_{i-1} <q,a,A\gamma> \vdash <q,\varepsilon,\eta\gamma>=<q,\varepsilon,\alpha>$

25

Passo induttivo:
Da S in i passi produco xα. Se è vero esiste una produzione che mi permette di fare questi passaggi.

25

Dimostrazione (gr. CF → aut. a pila)

- rimuoviamo la limitazione che $\varepsilon \notin L(G)$
  - aggiungiamo un nuovo stato q_0 iniziale
  - definiamo la transizione

    $\delta(q_0,\varepsilon,Z_0)=<q_0,\varepsilon>$

    e la transizione

    $\delta(q_0,a,Z_0)=\{<q,\gamma> | <q,\gamma> \in \delta(q,a,Z_0)\}$

26

26

085-automi-a-pila-03

Soluzioni

soluzione esercizio 12 — automa per $S \to a \mid S+S \mid S*S \mid (S)$

portiamo la grammatica in GNF

- portiamo in quasi CNF:

  $S \to SPS \mid SMS \mid ASZ \mid a$

  $P \to +$, $M \to *$, $A \to ($, $Z \to)$
- scegliamo l'ordinamento $S < P < M < A < Z$
- eliminiamo la ricorsione sinistra su S:

  $S \to ASZR \mid aR \mid ASZ \mid a$

  $R \to PSR \mid MSR \mid PS \mid MS$

  $P \to +$, $M \to *$, $A \to ($, $Z \to)$

35

35

Ordinamento dei non terminali, e per ogni produzione Ai->Aj… , dove i>j
Scelto l'ordinamento devo fare l'eliminazione della ricorsione

Da qui poi vado in forma di graibach:

Soluzioni

- sostituiamo a ritroso e semplifichiamo

  $S \to (SZR \mid aR \mid (SZ \mid a$ ε

  $R \to +SR \mid *SR \mid +S \mid *S$

  $Z \to)$ ε

- poniamo: $\Sigma=\{a, +, *, (,)\}$, $\Gamma = \{S, R, Z\}$, $Q = \{q\}$, $q_0= q$, $Z_0 = S$
- costruiamo l'insieme delle transizioni:

  $\delta(q, (, S) = \{<q, SZR>, <q, SZ>\}$

  $\delta(q, a, S) = \{<q, R>, <q, \varepsilon>\} \mid$

  $\delta(q, +, R) = \{<q, SR>, <q, S>\}$

  $\delta(q, *, R) = \{<q, SR>, <q, S>\}$

  $\delta(q,), Z) = \{<q, \varepsilon>\}$

36

Fatta la forma di graibach possiamo creare la funzione di transizione

DEL TIPO
Se il simbolo affiorante è S e leggi sulla stringa (metti nella pila SZR o SZ
Se S a R;
Se S a cancella anche l'elemento affiorante della pila
. . . . . .

Automi a pila e linguaggi CF

- Teorema: se un linguaggio è accettato da un automa a pila mediante pila vuota esiste una grammatica non contestuale che lo genera
- Dimostrazione
  - dato un automa a pila M costruiamo una grammatica non contestuale $G=<\Sigma,V_N,P,S>$ in GNF e mostriamo che $L(G)=N(M)$
  - strategia:
    - la computazione svolta da M è simulata in G da una derivazione in cui ogni forma di frase ha un prefisso di soli terminali (caratteri già letti da M) e un suffisso di soli simboli (la pila di M)

27

27

Dal teorema di prima e questo --> sappiamo che: Gli automi a pila non deterministici riconoscono tutti e soli i linguaggi CF.
Il limite di calcolo della pila in se è quello dei problemi dei linguaggi che puoi descrivere con le grammatiche CF.

COSTRUZIONE DELLA GRAMMATICA A PARTIRE DALL'AUTOMA.
DIM -> NON VA FATTA

dimostrazione (aut. a pila→gr. CF)

costruzione di G a partire da M:

- $V_N = \{S\} \cup \{[q,A,p] \mid$ per ogni $p,q \in Q$, per ogni $A \in \Gamma\}$
  - la categoria sintattica [qAp] rappresenta la sequenza di

q e p sono degli stati e A è un simbolo di pila. Per fare l'insieme di non terminali quante triplette mettiamo? p e q possono essere anche lo stesso stato. Tra i simboli c'è anche Z0.

caratteri che mi permette di andare nell'automa M da q a p rimuovendo A dallo stack

- $S \rightarrow [q_0,Z_0,q]$ per ogni $q \in Q$
- per ogni $\delta(q,a,A)$ di M e per ogni $\langle q_1,B_1....B_m\rangle \in \delta(q,a,A)$, con $a \in \Sigma \cup \{\varepsilon\}$ ed $A,B_1,....,B_m \in \Gamma$:
  - se m=0 allora $[q,A,q_1] \rightarrow a$,
  - altrimenti
  - simbolo app. nella pila
  - $[q,A,q_{m+1}] \rightarrow$
  - caratt presente nella fdt
  - $a[q_1,B_1,q_2][q_2,B_2,q_3]....[q_m,B_m,q_{m+1}]$
  - per tutte le combinazioni $q_2,....,q_{m+1} \in Q$

28

Metti nella grammatica produzioni da S->[q0,Z0,q] per ogni stato appartenente a Q. Ci metto tutti i non terminali raggiungibili da S.

se m=0 (non metti niente in pila)
Altrimenti: tra le produzioni inserisci le produzioni del tipo:

$[q,A,q_{m+1}] \rightarrow a[q_1,B_1,q_2][q_2,B_2,q_3],..[q_m,B_m,q_{m+1}]$

2 vincoli nelle produzioni

1 al posto dei buchi ci metto tutti gli stati poss seguendo la regola: l'ultimo stato uguale al primo successivo

2 e altra regola: q_{m+1} all'inizio e alla fine

28

dimostrazione (aut. a pila→gr. CF)

strategia della dimostrazione

- se x è una stringa di soli terminali, dimostriamo che

  $[q,A,p] \Rightarrow^* x$ se e solo se $\langle q,x,A\rangle \vdash^* \langle p,\varepsilon,\varepsilon\rangle$
- e quindi come caso particolare abbiamo che

  $[q_0,Z_0,p] \Rightarrow^* x$

  se e solo se

  $\langle q_0,x,Z_0\rangle \vdash^* \langle p,\varepsilon,\varepsilon\rangle$
- ne consegue che

  $S \Rightarrow^* x$ se e solo se per qualche stato p $\langle q_0,x,Z_0\rangle \vdash^* \langle p,\varepsilon,\varepsilon\rangle$

29

29

dimostrazione (aut. a pila→gr. CF)

strategia di dimostrazione

per dimostrare che

$[q,A,p] \Rightarrow^* x$ se e solo se $\langle q,x,A\rangle \vdash^* \langle p,\varepsilon,\varepsilon\rangle$

dimostreremo separatamente, ma sempre per induzione su i, che:

(1) se $\langle q,x,A\rangle \vdash_i \langle p,\varepsilon,\varepsilon\rangle$ allora $[q,A,p] \Rightarrow^* x$

(2) se $[q,A,p] \Rightarrow_i x$ allora $\langle q,x,A\rangle \vdash^* \langle p,\varepsilon,\varepsilon\rangle$

30

30

dimostrazione (aut. a pila→gr. CF)

(1) dimostriamo per induzione su i che

se $\langle q,x,A\rangle \vdash_i \langle p,\varepsilon,\varepsilon\rangle$ allora $[q,A,p] \Rightarrow^* x$

i=1 se $\langle q,x,A\rangle \vdash_1 \langle p,\varepsilon,\varepsilon\rangle$ allora

- $x \in \Sigma \cup \{\varepsilon\}$
- $\delta(q,x,A)$ contiene $\langle p,\varepsilon\rangle$

quindi esiste per costruzione $[q,A,p] \rightarrow x$

cosa che garantisce $[q,A,p] \Rightarrow^* x$

31

31

dimostrazione (aut. a pila→gr. CF)

(1) dimostriamo per induzione su i che

se $\langle q,x,A\rangle \vdash_i \langle p,\varepsilon,\varepsilon\rangle$ allora $[q,A,p] \Rightarrow^* x$

i>1 posso riscrivere

$\langle q,x,A\rangle \vdash_i \langle p,\varepsilon,\varepsilon\rangle$ come

$\langle q,ay,A\rangle \vdash \langle q_1,y,B_1,....,B_n\rangle \vdash_{i-1} \langle p,\varepsilon,\varepsilon\rangle$

posso riscrivere inoltre $y=y_1....y_n$ dove ogni y_i ($y_i \in \Sigma^*$) "rimuove" un B_i dalla pila, magari dopo una lunga sequenza di mosse

y_1 è il prefisso di y al termine del quale la pila contiene n-1 simboli;

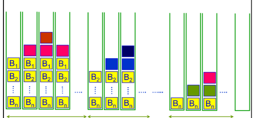
y_1y_2 è il prefisso di y al termine del quale la pila contiene n-2 simboli; ecc....

in generale B_j rimane nella pila mentre M legge la sequenza $y_1....y_{j-1}$

32

32

evoluzione della pila



33

33

dimostrazione (aut. a pila→gr. CF)

(1) dimostriamo per induzione su i che

se $\langle q,x,A\rangle \vdash_i \langle p,\varepsilon,\varepsilon\rangle$ allora $[q,A,p] \Rightarrow^* x$

i>1(continua) allora esistono degli stati $q_2,..,q_{n+1}=p$ tali che

$\langle q_1,y_1,B_1\rangle \vdash^* \langle q_2,\varepsilon,\varepsilon\rangle$

$\langle q_2,y_2,B_2\rangle \vdash^* \langle q_3,\varepsilon,\varepsilon\rangle$

....

$\langle q_n,y_n,B_n\rangle \vdash^* \langle q_{n+1},\varepsilon,\varepsilon\rangle$

(tutte computazioni con meno di i passi)

per ipotesi induttiva

$[q_1,B_1,q_2] \Rightarrow^* y_1$

$[q_2,B_2,q_3] \Rightarrow^* y_2$

....

$[q_n,B_n,q_{n+1}] \Rightarrow^* y_n$

inoltre abbiamo la produzione $[q,A,p] \rightarrow a[q_1,B_1,q_2][q_2,B_2,q_3]..[q_n,B_n,q_{n+1}]$

e quindi $[q,A,p] \Rightarrow^* ay_1y_2....y_n=ay=x$

34

34

dimostrazione (aut. a pila→gr. CF)

(2) dimostriamo per induzione su i che

se $[q,A,p] \Rightarrow_i x$ allora $<q,x,A> \vdash^* <p,\varepsilon,\varepsilon>$

i=1

se $[q,A,p] \Rightarrow_1 x$ allora
- $x \in \Sigma \cup \{\varepsilon\}$
- in G esiste $[q,A,p] \to x$

e quindi $<p,\varepsilon> \in \delta(q,x,A)$

cosa che garantisce $<q,x,A> \vdash <p,\varepsilon,\varepsilon>$

35

35

dimostrazione (aut. a pila→gr. CF)

(2) dimostriamo per induzione su i che

se $[q,A,p] \Rightarrow_i x$ allora $<q,x,A> \vdash^* <p,\varepsilon,\varepsilon>$

i>1 posso riscrivere ($q_{n+1}=p$)

$[q,A,p] \Rightarrow$

$\Rightarrow a[q_1,B_1,q_2][q_2,B_2,q_3]..[q_n,B_n,q_{n+1}] \Rightarrow_{i-1}$

$\Rightarrow_{i-1}$ $x = a\, x_1\, x_2 \ldots\, x_n$

dove $[q_j,B_j,q_{j+1}] \Rightarrow^* x_j$ (con meno di i passi)

dall'ipotesi induttiva abbiamo

$<q_j,x_j,B_j> \vdash^* <q_{j+1},\varepsilon,\varepsilon>$ $(1 \le j \le n)$

36

36

dimostrazione (aut. a pila→gr. CF)

(2) dimostriamo per induzione su i che

se $[q,A,p] \Rightarrow_i x$ allora $<q,x,A> \vdash^* <p,\varepsilon,\varepsilon>$

i>1 (continua)

dall'ipotesi induttiva abbiamo $<q_j,x_j,B_j> \vdash^* <q_{j+1},\varepsilon,\varepsilon>$ $(1 \le j \le n)$

inserendo in pila ("sotto" B_j) $B_{j+1},\ldots,B_n$

$<q_j,x_j,B_jB_{j+1}\ldots B_n> \vdash^* <q_{j+1},\varepsilon,B_{j+1}\ldots B_n>$

inoltre dal primo passo di derivazione:

$[q,A,p] \Rightarrow a[q_1,B_1,q_2][q_2,B_2,q_3]..[q_n,B_n,q_{n+1}]$

abbiamo $<q,x,A> \vdash <q_1,x_1x_2\ldots x_n,B_1\ldots B_n>$

quindi $<q,x,A> \vdash^* <p,\varepsilon,\varepsilon>$

37

37

automi a pila e linguaggi CF

esempio: dato il seguente automa a pila

$M=<\{0,1\},\{X,Z_0\},Z_0,\{q_0,q_1\},q_0,\varnothing,\delta>$

$\delta(q_0,0,Z_0)=\{<q_0,XZ_0>\}$

$\delta(q_0,0,X)=\{<q_0,XX>\}$

$\delta(q_0,1,X)=\{<q_1,\varepsilon>\}$

$\delta(q_1,1,X)=\{<q_1,\varepsilon>\}$

$\delta(q_1,\varepsilon,X)=\{<q_1,\varepsilon>\}$

$\delta(q_1,\varepsilon,Z_0)=\{<q_1,\varepsilon>\}$

descrivere la grammatica G che che genera N(M); di che linguaggio si tratta?

VN= $\{S,\ [q_0,X,q_0], [q_0,Z_0,q_0], [q_0,Z_0,q_1], [q_0,X,q_1], [q_1,X,q_1], [q_1,Z_0,q_1], [q_1,X,q_0], [q_1,Z_0,q_0]\}$

38

38

Esempio.
Costruzione -> trovo tutti i non terminali.

esempio (aut. a pila →grammatica CF)

esaminiamo solo le produzioni raggiungibili dall'assioma

$S \to [q_0,Z_0,q_0]$ $\quad S \to [q_0,Z_0,q_1]$

per il non terminale $[q_0,Z_0,q_0]$ abbiamo che la

$\delta(q_0,0,Z_0)=\{<q_0,XZ_0>\}$ impone

$[q_0,Z_0,q_0] \to 0[q_0,X,q_0][q_0,Z_0,q_0]$

$[q_0,Z_0,q_0] \to 0[q_0,X,q_1][q_1,Z_0,q_0]$

per il non terminale $[q_0,Z_0,q_1]$ abbiamo che la

$\delta(q_0,0,Z_0)=\{<q_0,XZ_0>\}$ impone

$[q_0,Z_0,q_1] \to 0[q_0,X,q_0][q_0,Z_0,q_1]$

$[q_0,Z_0,q_1] \to 0[q_0,X,q_1][q_1,Z_0,q_1]$

39

39

-> prime produzioni.
Vedo ciò che si può produrre da queste due produzioni di S.

esempio (aut. a pila →grammatica CF)

da $\delta(q_0,0,X)=\{<q_0,XX>\}$ abbiamo

$[q_0,X,q_0] \to 0[q_0,X,q_0][q_0,X,q_0]$

$[q_0,X,q_0] \to 0[q_0,X,q_1][q_1,X,q_0]$

$[q_0,X,q_1] \to 0[q_0,X,q_0][q_0,X,q_1]$

$[q_0,X,q_1] \to 0[q_0,X,q_1][q_1,X,q_1]$

inoltre quando m=0 (produzioni di solo 1 terminale)

$[q_0,X,q_1] \to 1$	da	$\delta(q_0,1,X)=\{<q_1,\varepsilon>\}$
$[q_1,Z_0,q_1] \to \varepsilon$	da	$\delta(q_1,\varepsilon,Z_0)=\{<q_1,\varepsilon>\}$
$[q_1,X,q_1] \to \varepsilon$	da	$\delta(q_1,\varepsilon,X)=\{<q_1,\varepsilon>\}$
$[q_1,X,q_1] \to 1$	da	$\delta(q_1,1,X)=\{<q_1,\varepsilon>\}$

40

40

esempio (aut. a pila →grammatica CF)

riassumendo abbiamo

- $S \to [q_0,Z_0,q_0]$ $\quad S \to [q_0,Z_0,q_1]$
- $[q_0,Z_0,q_0] \to 0[q_0,X,q_0][q_0,Z_0,q_0]$
- $[q_0,Z_0,q_0] \to 0[q_0,X,q_1][q_1,Z_0,q_0]$
- $[q_0,Z_0,q_1] \to 0[q_0,X,q_0][q_0,Z_0,q_1]$

  $[q_0,Z_0,q_1] \to 0[q_0,X,q_1][q_1,Z_0,q_1]$
- $[q_0,X,q_0] \to 0[q_0,X,q_0][q_0,X,q_0]$
- $[q_0,X,q_0] \to 0[q_0,X,q_1][q_1,X,q_0]$
- $[q_0,X,q_1] \to 0[q_0,X,q_0][q_0,X,q_1]$

  $[q_0,X,q_1] \to 0[q_0,X,q_1][q_1,X,q_1]$

  $[q_0,X,q_1] \to 1$ $\quad [q_1,Z_0,q_1] \to \varepsilon$

  $[q_1,X,q_1] \to \varepsilon$ $\quad [q_1,X,q_1] \to 1$

rosso: elimino queste e quelle collegate a ritroso

quindi

alla fine viene così

$S \to [q_0,Z_0,q_1]$

$[q_0,Z_0,q_1] \to 0[q_0,X,q_1][q_1,Z_0,q_1]$

$[q_0,X,q_1] \to 0[q_0,X,q_1][q_1,X,q_1]$

$[q_0,X,q_1] \to 1$ $\quad [q_1,Z_0,q_1] \to \varepsilon$

$[q_1,X,q_1] \to \varepsilon$ $\quad [q_1,X,q_1] \to 1$

41

41

esempio (aut. a pila →grammatica CF)

generazione di 00011	riconoscimento di 00011
$S \Rightarrow [q_0,Z_0,q_1]$	$<q_0,\ 00011,\ Z_0>$
$\Rightarrow 0\ [q_0,X,q_1][q_1,Z_0,q_1]$	$<q_0,\ 0011,\ XZ_0>$
$\Rightarrow 00\ [q_0,X,q_1][q_1,X,q_1][q_1,Z_0,q_1]$	$<q_0,\ 011,\ XXZ_0>$
$\Rightarrow 000\ [q_0,X,q_1][q_1,X,q_1][q_1,X,q_1][q_1,Z_0,q_1]$	$<q_0,\ 11,\ XXXZ_0>$
$\Rightarrow 0001\ [q_1,X,q_1][q_1,X,q_1][q_1,Z_0,q_1]$	$<q_1,\ 1,\ XXZ_0>$
$\Rightarrow 00011\ [q_1,X,q_1][q_1,Z_0,q_1]$	$<q_1,\ \varepsilon,\ XZ_0>$
$\Rightarrow 00011\ [q_1,Z_0,q_1]$	$<q_1,\ \varepsilon,\ Z_0>$
$\Rightarrow 00011$	$<q_1,\ \varepsilon,\ \varepsilon>$

42

42

06-automi-a-pila-18

Ambiguità

- problema fondamentale per i linguaggi di programmazione
- una grammatica non contestuale G è ambigua se esiste una stringa x in L(G) derivabile con due alberi di derivazione diversi
- esempio: data la grammatica
  $E \rightarrow E + E \mid E * E \mid a$
  la stringa a+a*a può essere derivata come segue
  $E \Rightarrow E+E \Rightarrow E+E*E \Rightarrow^* a+a*a$
  $E \Rightarrow E*E \Rightarrow E+E*E \Rightarrow^* a+a*a$

43

43

Per una stringa ci possono essere diversi alberi di derivazione.

-> due modi diversi per derivare a+a*a.
Tipicamente l'albero di derivazione associa una semantica al linguaggio (es prima somma e poi moltiplica, o prima moltiplica e poi somma).
Ci sono 2 scelte diverse che portano a due semantiche diverse. Come posso eliminare le ambiguità per i linguaggi che esprimono operazioni aritmetiche?

Ambiguità

- osservazione: ci sono due metodi per evitare l'ambiguità: parentesi e precedenza
- esempio: la grammatica della slide precedente può essere modificata in
  $E \rightarrow (E + E) \mid (E * E) \mid a$
  generando (a+(a*a)) oppure ((a+a)*a)
- esempio: grammatica delle espressioni aritmetiche in cui il prodotto/divisione ha precedenza sulla somma/sottrazione
  $E \rightarrow E + T \mid E - T \mid T$
  $T \rightarrow T * F \mid T / F \mid F$
  $F \rightarrow (E) \mid a$

44

44

Modifico così la grammatica.

Cioè aggiungo le parentesi per preservare una semantica particolare.

Oppure la modifico ulteriormente
F -> fattore
T-> termini
E->espressione(?)

06-automi-a-pila-18

Linguaggi inerentemente ambigui

- un linguaggio CF è inerentemente ambiguo se tutte le grammatiche che lo generano sono ambigue
- esempio di linguaggio non inerentemente ambiguo
  $E \rightarrow E + E \mid E * E \mid a$
  ha una grammatica equivalente non ambigua
  $E \rightarrow a+ E \mid a* E \mid a$
- esempio di linguaggio inerentemente ambiguo
  $\{a^n b^n c^m \mid n,m \geq 1\} \cup \{a^m b^n c^n \mid n,m \geq 1\}$
  infatti, per qualunque grammatica le stringhe $a^n b^n c^n$ possono essere generate in due modi diversi

45

45

Anche il linguaggio CF può essere ambiguo

Bisogna stare attenti alla semantica

Confronto tra linguaggi di tipo 3 e di tipo 2

chiusura

	LR	LCF
unione	si	si
concatenazione	si	si
stella	si	si
intersezione	si	no
complementazione	si	no

problemi di decisione

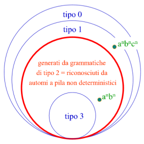
	LR	LCF
w in L?	D	D
$L=\emptyset$?	D	D
$\lvert L \rvert=\infty$?	D	D
L1=L2?	D	?
$L1 \cap L2=\emptyset$?	D	?

46

46

06-automi-a-pila-18

Quadro riassuntivo sui linguaggi CF



47

47