Appunti Di Automi, (LINGUAGGI ACONTESTUALI) Complessità, Calcolabilità

LINGUAGGI ACONTESTUALI

CONTEXT-FREE GRAMMAR

Una CFG e' una quadrupla (V, Σ, R, S)

V : insieme delle variabili

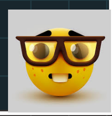
Σ : insieme dei terminali

R : insieme delle regole

$S \in V$: stato iniziale

$V \cap \Sigma = \emptyset$ VARIABILI E TERMINATORI DISTINTI

IMPORTANTE!
a sx di una regola in R c'è sempre un solo termine!



ACONTESTUALITA'

Questi linguaggi si dicono acontestuali perché a sinistra delle regole c'è solo un termine, le transizioni non sono influenzate da altri termini.

NOTAZIONE

Se abbiamo regole del tipo

$A \to B$

$A \to CD$

$A \to K$

possiamo scrivere $A \to B \mid cD \mid k$

PRODUZIONE

Sia G una CFG (V,Σ,R,S), $vwv \in (V \cup \Sigma_\varepsilon)^*$ e esiste le regole $w \to k$ in R

ALLORA

$vwv \Rightarrow vkv$ **PRODUCE**

DERIVAZIONE

dati $u, k \in (V \cup \Sigma_\varepsilon)^*$ diciamo che $u \overset{*}{\Rightarrow} k$ se $\exists$ **DERIVA**

$u_1 \dots u_k \mid u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \dots \Rightarrow u_k \Rightarrow k$

CFL (CONTEXT-FREE LANGUAGE) Sia G CFG (V,Σ,R,S)

un CFL L e' così definito

$L(G)$ dove G e' una CFG $= \{ w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w \}$

Data una CFL possono esistere più derivazioni della stessa stringa w

DERIVAZIONE SINISTRA

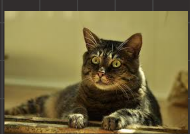
derivazione svolta applicando le regole alle variabile più a sinistra ad ogni passo.

GRAMMATICA AMBIGUA

UNA CFG G e' ambigua se $\exists\, w \in L(G)$ e w ha almeno 2 derivazioni a sinistra.

CLASSE DEI LINGUAGGI ACONTESTUALI

$CFL = \{ L \mid \exists\, G \text{ CFG} \mid L = L(G) \}$

REG ⊆ CFL 

1 **DIMOSTRAZIONE**

Sia $G = (V,\Sigma,R,S)$ una CFG

Sia $D = (Q,\Sigma,\delta,q_0,F)$ una DFA

Sia σ una funzione biettiva

PRESA IN MODO TALE CHE:

$\sigma: Q \to V : q_i \to v_i$

$S = q_0 = V_0$

POI, DATI $q_i, q_J \in Q$ e $a \in \Sigma$

$\delta(q_i, a) \to q_J$ ALLORA $\sigma(V_i) = aV_J$ ALLORA $\boxed{V_i \to aV_J}$ PERCHÉ

$\delta(q_i, a) \to q_J = \sigma(q_i) = a\sigma(q_J)$

$q_F \in F$ ALLORA $\sigma(q_F) = \varepsilon$ ALLORA $\boxed{V_F \to \varepsilon}$

CI ACCORGIAMO ORA CHE:

$\mathcal{L}(G) \Leftrightarrow \mathcal{L}(D)$

$\Rightarrow$ REG $\subseteq$ CFL

IN REALTÀ REG $\subsetneq$ CFL PERCHÉ consideriamo il linguaggio REGOLARE $L = \{0^n 1^n \quad n \in \mathbb{N}\}$

QUESTO È UN LINGUAGGIO ACONTESTUALE MA NON È REGOLARE ⚠

FORMA NORMALE DI CHOMSKY

Una grammatica CFG $G = (V, \Sigma, R, S)$ È IN CNF (CHOMSKY NORMAL FORM)

Se in R ci sono solo regole di queste forme:

$S \to \varepsilon$ SOLO LO STATO INIZIALE VA IN ε

$A \to BC$

$A \to a$

POSSO AVERE REGOLE DA V A Σ O DA V A V, NON MIX

dove $A \in V$, $a \in \Sigma$, B e $C \in V - \{S\}$

$\forall$ CFG G CHE NON È IN CNF ESISTE CFG G' CHE È IN CNF EQUIVALENTE

② DIMOSTRAZIONE (PROCEDIMENTO)

1. Viene aggiunto uno stato S_0 e le regole $S_0 \to S$
2. S_0 è il nuovo stato iniziale 
3. $\forall$ ε-REGOLA (EX: $A \to \varepsilon$) viene eliminata e vengono aggiunte delle regole tipo se $\cancel{A \to \varepsilon}$ e $B \to vAkAw$ allora vengono aggiunte le regole $B \to vAkAw \mid vkAw \mid vAkw \mid vkw$

4. ogni regola tipo $A \to B$ viene eliminata e ogni $B \to u$ dove $u \in (V \cup \Sigma)^*$ viene sostituito con $A \to u$

5. $\forall$ regola $A \to u_1 u_2 u_3 \dots u_k$ dove $k \geq 3$
$\forall i$ $[1, k-2]$ creo
$A \to A_1 u_1$, $A_1 \to A_2 u_2$, ..., $A_{k-2} \to u_k u_{k-1}$

6. ELIMINO $A \to u_1 u_2 \dots u_k$

7. $\forall A \to u_1 u_2 \mid u_1 \in (\Sigma \cup V) \land u_2 \in (\Sigma \cup V)$
Se $u_1 \in \Sigma$
AGGIUNGO una variabile U_1 e una regola $U_1 \to u_1$
AGGIUNGO regole $A \to U_1$
ELIMINO regole $A \to u_1$
Se $u_2 \in \Sigma$ faccio lo stesso

AUTOMI A PILA (STACK)

Un automa a pila PDA P e' (PILE)
una sestupla $(Q, \Sigma, \Gamma, \delta, q_0, F)$
dove
Q e' l'insieme degli stati
Σ e' l'alfabeto dell'automa
Γ e' l'alfabeto dello stack
δ e' la funzione di transizione
q_0 stato iniziale
F insieme degli stati accettanti dell'automa.

$\delta : (Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon) \to P(Q \times \Gamma_\varepsilon)$

quindi se $(q, e) \in \delta(p, a, b)$ viene letto l'input a, l'automa passa dallo stato p allo stato q, b viene sostituito con e.

LA TRANSIZIONE DA p a q e' SCRITTA $p \to q$

STRINGHE ACCETTATE IN UN PDA

Sia P PDA P (PUSH DOWN AUTOMATON) $= (Q, \Sigma, \Gamma, \delta, q_0, F)$, data una stringa $w_0 \dots w_k \in \Sigma_\varepsilon^*$ e $w_0, \dots, w_k \in \Sigma_\varepsilon^*$ la stringa e' accettata da P se esistono una serie di stati $r_1, \dots, r_{k+1}$ e una sequenza di stringhe $S_1, \dots, S_k \in \Gamma_\varepsilon^*$ e

$r_1 = q_0$

$r_{k+1} \in F$

$S_0 = \emptyset$ STACK AL PASSO 0esimo VUOTO

$\forall i \ [1, k]$

$(r_{i+1}, a_{i+1}) \in \delta(r_i, w_i, a_i)$

$S_i = a_i K$

$S_{i+1} = (a_{i+1}) K$

il linguaggio sara' allora riconosciuto da P

$L = L(P)$

L(PDA)

L(PDA) = CLASSE DEI LINGUAGGI RICONOSCIUTI DA UN PDA

$L(PDA) = \{ L \mid \exists \ P \ \text{PDA t.c. } L = L(P) \}$

SCRITTURA STRINGA SU UNO STACK

Sia P PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, F)$

Siano $w_0, \dots, w_k \in \Gamma_\varepsilon^*$

Sia $w = w_0, \dots, w_k \in \Gamma_\varepsilon^*$

per scrivere cio' sullo stack di P devono esistere una serie di stati $r_0, \dots, r_k$

$(r_k, w) \in \delta(p, a, b) \iff \exists \ r_0, \dots, r_k \in Q$ t.c.

$\delta(p, a, b) \ni (r_0, w_k)$

$\delta(r_1, \varepsilon, \varepsilon) = (r_2, w_{k-1})$

$\vdots$

$\delta(r_{k-1}, \varepsilon, \varepsilon) = (r_k, w_1)$

IMPORTANTE!

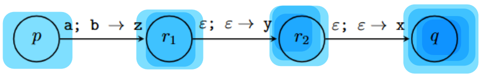
Osservazione 2.5

Dato $(q, c) \in \delta(p, a, b)$ dove δ è la funzione di transizione di un PDA, si ha che:

- Se $b, c = \varepsilon$ (dunque a; $\varepsilon \to \varepsilon$) allora l'automa leggerà a dalla stringa e passerà direttamente dallo stato p allo stato q, senza modificare lo stack
- Se $b = \varepsilon$ e $c \neq \varepsilon$ (dunque a; $\varepsilon \to c$) allora l'automa leggerà a dalla stringa, passerà direttamente dallo stato p allo stato q e in cima allo stack viene aggiunto il simbolo c (**push**)
- Se $b \neq \varepsilon$ e $c = \varepsilon$ (dunque a; $b \to \varepsilon$) allora l'automa leggerà a e se in cima allo stack vi è b, l'automa passerà dallo stato p allo stato q e rimuoverà b dalla cima dello stack (**pop**)

Esempio:

- Dato $(q, xyz) \in \delta(p, a, b)$ si ha che:



POSSIAMO QUINDI AFFERMARE CHE

$CFL \subseteq \mathcal{L}(PDA)$

③ DIMOSTRAZIONE