Linguaggi liberi da contesto

sintassi dei linguaggi di programmazione

costruzione automatica di parser

ESEMPIO

Il linguaggio dei palindromi $L_{pal}=\{w \mid w = w^R\}$

Per alfabeto {0,1} contiene 0110, 11011, epsilon

ma non contiene 011 o 1010

L_{pal} non è regolare, lo dimostriamo usando il pumping lemma

Pumping Lemma:

per ogni linguaggio regolare L, esiste una costante n (che dipende da L) e tale che per ogni w in L t.c. |w|>=n allora w=xyz t.c.

(i) y non è epsilon,

(ii) |xy| <=n

(iii) per ogni k>=0, xy^kz è in L

$w=0^n10^n$

w=xyz e dato che |xy|<=n e y non è epsilon, y contiene alcuni 0 e quindi $xy^0z = xz$ non può essere in L_{pal} perché contiene meno 0 a sinistra del solo 1 rispetto agli n che sono a destra per definire il linguaggio L_{pal} possiamo usare una definizione ricorsiva

<u>Base</u>: epsilon (stringa vuota) , 0 e 1 sono in L_{pal}

<u>Induzione</u>: se w è in L_{pal} , allora 0w0 e 1w1 sono in L_{pal}

una grammatica libera dal contesto (context-free) è una notazione formale per esprimere tali definizioni ricorsive

consiste di variabili, terminali e produzioni

Per L_{pal} definiamo G_{pal}

1. P -> epsilon
2. P-> 0
3. P-> 1
4. P-> 0 P 0
5. P-> 1 P 1

base della definizione

passo induttivo

la usiamo G_{pal} per generare L_{pal}

Definizione di grammatica libera da contesto

1. insieme finito T di simboli terminali che formano le stringhe del linguaggio (per L_{pal} 0 e 1)
2. insieme finito V di variabili (anche nonterminali); ogni variabile genera un linguaggio (per L_{pal} c'è solo la variabile P)
3. un simbolo iniziale (S) che genera il linguaggio da definire (in L_{pal} sarà P)
4. un insieme finito (P o R) di produzioni o regole che hanno forma: testa->corpo

dove, la testa è una variabile e il corpo è una stringa in (V U T)*
quindi anche vuota e composta di terminali e/o variabili, la chiameremo <u>forma sentenziale</u>

G_{pal} =({P},{0,1},R,P)
dove R contiene le 5 regole:

1. P -> epsilon
2. P-> 0
3. P-> 1
4. P-> 0P0
5. P-> 1P1

notazione compatta:
P -> epsilon | 0 | 1 | 0P0 | 1P1

Linguaggio di una grammatica

Una CFG serve per stabilire se determinate stringhe appartengono al linguaggio della grammatica

2 strade:

--dalla testa al corpo : derivazione

--dal corpo alla testa : inferenza ricorsiva

<u>derivazione</u>, indicata => è relazione tra forme sentenziali, definita come segue:

sia G=(V,T,R,S) e sia α A β dove α e β sono forme sentenziali ed A è in V, sia A-> γ in R, allora α A β => α γ β

=>* è la chiusura riflessiva e transitiva di =>

significa zero o più passi di =>

base: per qualsiasi forma sentenziale α, $\alpha =>^* \alpha$
induzione : $\alpha =>^* \beta$ e $\beta => \gamma$, allora $\alpha =>^* \gamma$

o anche

$\alpha =>^* \beta$ significa che esiste un n>=1 e $\gamma 1, \gamma 2, \ldots, \gamma n$ t.c.
-- $\alpha = \gamma 1$, $\beta = \gamma n$
--per ogni i in [1..n-1], $\gamma i => \gamma i+1$

osserva: con n=1 => caso base e n>1 induzione

Definizione: linguaggio di una grammatica

Dato G=(V,T,R,S), L(G) ={ w in T*| S =>* w }

linguaggio del nonterminale X, L(X)= { w in T*| X =>* w } derivazioni di Gpal:

P -> epsilon | 0 | 1 | 0P0 | 1P1

P=>0P0=>00P00=>001P100=>0010100

P=>*0010100

inferenza ricorsiva di 0010100

1)P -> epsilon
2)P -> 0
3)P -> 1
4)P -> 0P0
5)P -> 1P1

(i)	**0**	**P**	**3**	**--**
(ii)	**101**	**P**	**5**	**(i)**
(iii)	**01010**	**P**	**4**	**(ii)**
(iv)	**0010100**	**P**	**4**	**(iii)**

L(G) è l'insieme delle stringhe di terminali per cui esiste un'inferenza ricorsiva

Teorema. $L(G_{pal})$ è l'insieme delle palindrome su $\{0,1\}$

Dimostrazione: w in $L(G_{pal})$ se e solo se è palindromo.

(se <=) supponiamo che w sia palindroma. Mostriamo per induzione su |w| che w in L(Gpal).

Base: $|w|=0$ o 1. Se $w=\varepsilon$, allora w in $L(G_{pal})$, se w=0/1 lo stesso.

Induzione: supponiamo che $|w|=n >=2$. Poichè $w=w^R$, deve iniziare e finire con lo stesso simbolo, quindi w=0w'0 o 1w'1 e w' deve essere palindromo, ma essendo $|w'| < n$, per ipotesi induttiva, w' è in $L(G_{pal})$ e quindi, viste le produzioni P->0P0|1P1 di G_{pal} anche w lo è.

(solo se =>) se w in $L(G_{pal})$ allora è palindromo. Induzione sulla lunghezza della derivazione.

<u>Base</u>: 1 sola produzione, w=є o 0 o 1, sono palindromi

<u>Induzione</u>: supponiamo che w sia generata in n+1 passi, e che l'enunciato sia vero per tutte le stringhe generate in n passi. Una tale derivazione deve iniziare con P => 0P0 | 1P1 e poi 0x0|1x1 =w

ma allora $P \Rightarrow^* x$ in n passi e quindi per ipotesi induttiva, x è palindroma per cui anche 0x0 e 1x1 lo sono.

altro esempio: vogliamo una grammatica che generi espressioni tipo, a+b*a1*(b1+aa0)
le operazioni sono * e + e gli operandi sono identificatori che iniziano per a o b e continuano con {a,b,0,1}*
Servono 2 variabili, E che descrive le espressioni e I che descrive gli identificatori.
Il linguaggio generato da I è regolare:

(a+b)(a+b+0+1)*

I->a | b | Ia | Ib | I0 | I1

Le regole per E sono:

E-> I | E+E | E*E | (E)

La grammatica è:

G=({E,I}, {a,b,0,1,*,+,(,)}, R, E)

dove R contiene le regole per I e per E

<u>inferenza</u> <u>ricorsiva</u> di a*(a+b00)

1. E-> I
2. E-> E+E
3. E -> E*E
4. E -> (E)
5. I-> a
6. I -> b
7. I -> Ia
8. I -> Ib
9. I -> I0
10. I -> I1

etichetta	stringa	V	Prod	stringhe usate
(i)	a	I	5	_
(ii)	b	I	6	_
(iii)	b0	I	9	(ii)
(iv)	b00	I	9	(iii)
(v)	a	E	1	(i)
(vi)	b00	E	1	(iv)
(vii)	a + b00	E	2	(v), (vi)
(viii)	(a +b00)	E	4	(vii)
(ix)	a * (a+b00)	E	3	(v),(viii)

esempio: derivazione di a*(a+b00)

E-> I | E+E | E*E | (E)
I-> a | b | Ia | Ib | I0 | I1

E => $\underline{E}$*E => $\underline{I}$*E => a*E => a*(E) => a*($\underline{E}$+E) => a*($\underline{I}$+E) => a*(a+$\underline{E}$) => a*(a+$\underline{I}$) => a*(a+$\underline{I}$0) => a*(a+$\underline{I}$00) => a*(a+b00)

sono possibili scelte diverse per ottenere la stessa stringa leftmost derivation e rightmost derivation

si sceglie sempre la variabile più a sinistra/destra della forma sentenziale:

$\Rightarrow^{lm}$ $\Rightarrow^{rm}$

Notazione:

--a, b sono terminali

--A,B,.. sono variabili

--w, z,.. sono stringhe di terminali

--X, Y sono o terminali o variabili

--α,β,..sono stringhe di terminali e variabili la derivazione di a*(a+b00) che abbiamo visto è leftmost

osserva che esiste anche una derivazione rm di a*(a+b00) forme sentenziali: data $G=(V,T,P,S)$

se $S \Rightarrow^{rm*}$
allora α è una forma sentenziale destra

in modo simile se $S \Rightarrow^{lm*} \alpha$
forma α è forma sentenziale sinistra

esistono forme sentenziali che non sono né destra né sinistra

ESERCIZI

1) Which language generates the grammar G given by the productions

$S \to aSa \mid aBa$

$B \to bB \mid b$

2) Find a CFG that generates the language:

$L(G) = \{ a^n\, b^m\, c^m\, d^{2n} \mid n \geq 0, m > 0\}$.

3) Find a CFG that generates the language

$L(G) = \{ a^n\, b^m \mid 0 \leq n \leq m \leq 2n\}$.

4) Consider the grammar

$S \to abScB \mid \text{epsilon}$

$B \to bB \mid b$

What language does it generate?

Esercizi 5.1.1 trovare grammatiche per:

a) $\{0^n 1^n \mid n>=1\}$
b) $\{a^i b^j c^k \mid i\backslash=j \text{ o } j\backslash=k\}$
c) ! l'insieme di tutte le stringhe in {a,b}* t.c. non siano ww
d) !! l'insieme di tutte le stringhe in {a,b}* con un numero doppio di b rispetto agli a

esercizio 5.1.2 fare facile

esercizio 5.1.4 fare