Linguaggi liberi da contesto

--dagli anni 1960’s usate per parser

--recentemente DTD in XML

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, usiamo 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
Base: 0 e 1 sono in L_{pal}
Induzione: 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}

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

1-3 sono la base, mentre 4 e 5 sono la parte induttiva della grammatica

la usiamo per generare L_{pal}

Definizione di grammatica libera da contesto

1. insieme finito di simboli terminali che formano le stringhe del linguaggio (per L_{pal} 0 e 1)
2. insieme finito di variabili (anche nonterminali); ogni variabile genera un linguaggio (per L_{pal} c’è solo la variabile P)
3. un simbolo iniziale che genera il linguaggio da definire (in L_{pal})
4. un insieme finito di produzioni o regole della forma testa-> corpo

dove, la testa è una variabile e il corpo è una stringa (anche vuota) di terminali e variabili

G_{pal} =({P},{0,1},A,P)
dove A contiene le 5 produzioni viste prima

esempio: vogliamo una grammatica per 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 genera le espressioni e I che genera gli identificatori.
Il linguaggio generato da I è regolare
(a+b)(a+b+0+1)*
I->a
I->b
I->Ia
I->Ib   I -> I0   I ->I1

Le produzioni per E sono:

E-> I

E-> E+E

E-> E*E

E-> (E)

La grammatica è:

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

dove P contiene le produzioni per I e per E

notazione compatta : E-> I | E+E | E*E | (E)

Derivazioni per mezzo di una grammatica

Una CFG serve per stabilire se determinate stringhe appartengono al linguaggio di una data variabile

2 strade:
--dal corpo alla testa : inferenza ricorsiva
--dalla testa al corpo : derivazione

inferenza ricorsiva di a*(a+b00)

	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)

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

derivazione

relazione =>

sia G=(V,T,P,S) e sia α A β dove α e β in (V u T)* ed A in V

sia A->γ in P, allora α A β => α γ β

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

significa zero o più passi di =>

base: per qualsiasi stringa α =>* α

Induzione : α =>* β e β => γ, allora α =>* γ

oppure

α =>* β significa che esiste un n>=1 e $\gamma 1$, $\gamma 2$,..., γn t.c.

-- α= $\gamma 1$, β= γn

--per ogni i in [1..n-1], γi => $\gamma i+1$

esempio: derivazione di a*(a+b00)

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

E => E*E => I*E => a*E => a*(E) => a*(E+E) => a*(I+E) => a*(a+E) => a*(a+I) => a*(a+I0) => a*(a+I00) => a*(a+b00)

ordini diversi per ottenere la stessa stringa

e produzioni diverse danno stringhe diverse leftmost derivation e rightmost

si sceglie sempre la variabile più a sinistra/destra

$\Rightarrow_{lm}$ $\Rightarrow_{rm}$

Notazioni:

--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 lm

esiste anche una derivazione rm di a*(a+b00)

Definizione: linguaggio di una grammatica
Dato G=(V,T,P,S), L(G) ={ w in T*| S =>* w }

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 $L(G_{pal})$.

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.
Base: 1 sola produzione, w=ϵ o 0 o 1, sono palindromi
Induzione: 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.

forme sentenziali:

data $G=(V,T,P,S)$

$S \Rightarrow^* \alpha$

allora è una forma sentenziale

può avere terminali e variabili

se $S \Rightarrow_{rm}^* \alpha$

allora è una forma sentenziale destra

in modo simile se $S \Rightarrow_{lm}^* \alpha$

forma sentenziale sinistra

esistono forme sentenziali che non sono né destra né sinistra 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

esercizio 5.1.4