proprietà delle CFG e forme normali

La forma normale di Chomsky (CNF) per le CFG prevede che tutte le produzioni siano:

A -> a, con a in T

A-> BC, con B e C variabili

non c’è ε !!

Teorema. Per ogni CFG G, esiste una CFG G’ in CNF tale che L(G’)=L(G)-{ε} ripuliamo la CFG

--dai simboli inutili

--dalle produzioni unitarie, i.e. A->B

--dalla ε

un simbolo X è <u>utile</u> per una CFG se

$S \Rightarrow^* \alpha X \beta \Rightarrow^* w$

un simbolo è generatore se

$X \Rightarrow^* w$

è raggiungibile se

$S \Rightarrow^* \alpha X \beta$

quindi utile = generatore + raggiungibile facile calcolare i simboli generatori e quelli raggiungibili

gli altri sono i non generatori e i non raggiungibili

<u>prima</u> si eliminano i non generatori
e <u>poi</u> i non raggiungibili

restano solo i simboli utili

esempio:

S-> AB | a

A -> b

non generatori : B

eliminiamo le produzioni che li contengono

resta S -> a e A -> b

ma A è diventato irraggiungibile, quindi resta

S -> a

Attenzione all’ordine

Teorema 7.2 data la CFG G, costruiamo G1 in 2 passi:

1) Otteniamo G2 da G eliminando da G le variabili non generatrici e le produzioni in cui queste variabili compaiono
2) Otteniamo G1 da G2 eliminando da G2 i simboli non raggiungibili

G1 contiene tutti e soli i simboli utili di G e quindi L(G1)=L(G)

Dimostrazione: tutti i simboli e produzioni usati in $S \Rightarrow^*_G w$ sono mantenute in G1, i simboli inutili sono eliminati.

calcolo dei simboli generatori $\prod$:

-base: i simboli terminali sono generatori, $\prod = T$

-induzione: aggiungiamo a $\prod$ tutte le variabili X t.c. esiste X -> α in cui α ha solo simboli in $\prod$

si continua fino a quando $\prod$ cresce

Calcolo dei simboli raggiungibili R

base: R contiene S

induzione: Aggiungere ad R i simboli in β tale che A-> β per A in R

In modo simile:

<u>calcolo dell'insieme Z dei simboli che producono ε</u>

base: X -> ε mettiamo X in Z

induzione: aggiungiamo a Z ogni Y tale che
Y-> α con i simboli di α tutti in Z

Per una qualsiasi CFG G possiamo costruire G' t.c. L(G')=L(G)-{ε} e senza ε-produzioni

Idea della costruzione di G':
Prendiamo una qualsiasi produzione p di G, A->X1…Xk, se m dei k Xi sono in Z, allora G' ha 2^m produzioni ottenute da p annullando tutti i sottoinsiemi delle m Xi

non si considera m=k, infatti in quel caso A sarebbe in Z
Le produzioni A -> ε sono eliminate

Esempio:

S-> AB

A-> a AA | ε

B-> bBB | ε

Z={A,B,S} quindi G' ha

S-> AB | A | B

A->aAA | aA | a

B-> bBB | bB | b

Eliminazione delle produzioni unitarie

A -> B

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

al posto di E -> T, mettiamo E -> F | T * F
e poi E -> I | (E)
e poi E -> a | b | Ia....

attenzione: $A \rightarrow B \rightarrow C \rightarrow A$

per evitare il trabocchetto calcoliamo tutte le coppie (A,B) t.c. $A \Rightarrow^* B$ con sole produzioni unitarie

le chiamiamo coppie unitarie

se (A,B) e $B \Rightarrow \alpha$ allora $A \rightarrow \alpha$

calcolo delle coppie unitarie:

base: (A,A) per tutti A in V

step: (A,B) è unitaria e B -> C
allora (A,C) è una coppia unitaria

esempio:

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

F -> I | (E)

T -> F | T * F

E -> T | E +T

(E,E),(T,T),(F,F),(I,I)

(E,E) e E -> T   (E,T)

(E,T) e T -> F   (E,F)

(E,F) e F -> I (E,I)

analogamente ((T,F), (T,I), (F,I)

vedere testo per soluzione completa

sappiamo
eliminare i simboli inutili
eliminare le produzioni ε
eliminare le produzioni unitarie

La forma normale di Chomsky (CNF) per le CFG prevede che tutte le produzioni siano:

A -> a, con a in T

A-> BC, con B e C variabili

Vogliamo trasformare una qualsiasi grammatica in FNC

partiamo da una grammatica che non ha simboli inutili, né produzioni ε, né produzioni unitarie, quindi le produzioni sono A -> a oppure A -> α con α di 2 o più simboli i terminali b di α li sostituiamo con nuove variabili B che hanno B -> b

se | α|=2 siamo a posto

ae | α| > 2 , per esempio α=B1B2..Bk, allora definiamo k-2 nuove variabili: C1...Ck-2 e sostituiamo la produzione originale A->B1...Bk con A -> B1C1, C1 -> B2C2, C2 -> B3C3,...Ck-2 -> Bk-1Bk

Teorema 7.17.

Negli alberi di derivazione di una CFG in CNF è vero che se il cammino più lungo è di lunghezza n, allora il prodotto w è t.c. $|w| <= 2^{n-1}$

Dimostrazione: per induzione su n

Base: n=1 l'albero consiste della radice e di una foglia etichettata con un terminale.

Prodotto w = foglia quindi $|w|=1 = 2^{1-1}$

Induzione: altezza n. La radice usa A->BC
B e C sono radici di alberi di altezza al massimo n-1,

quindi i loro prodotti, per ipotesi induttiva, sono di lunghezza al più 2^{n-2}, e $2 * 2^{n-2} = 2^{n-1}$

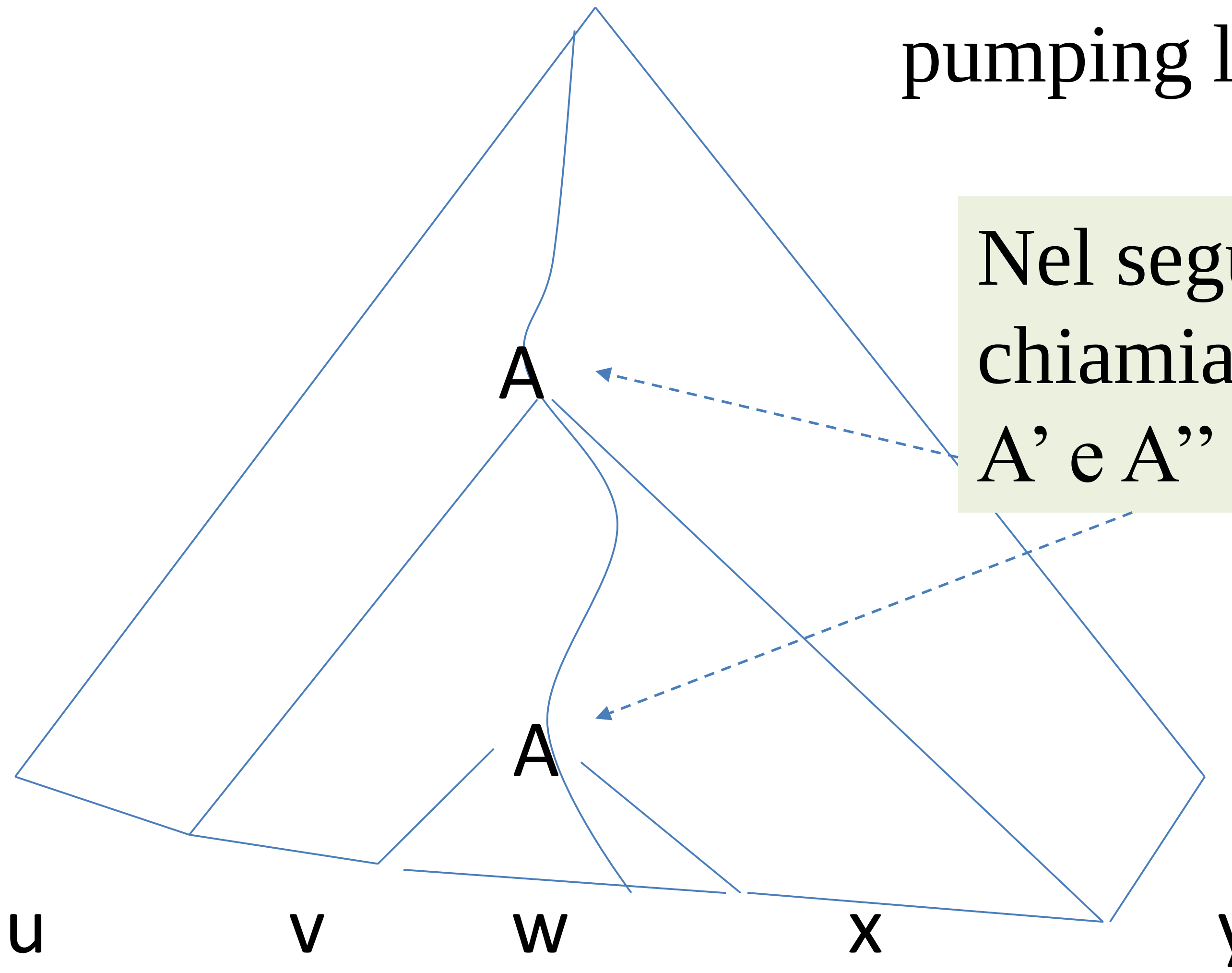
Conseguenza:

se m è il numero delle variabili della grammatica in CNF:
un albero con prodotto w t.c. $|w|>=2^m$ deve avere un cammino di lunghezza >= m+1

e allora su quel cammino almeno una variabile ripete

Quindi nel seguito $n = 2^m$

pumping lemma



Nel seguito li chiamiamo A' e A''

$$z = uvwxy$$

con $|vwx| <= n$ e $vx\ != \varepsilon$

In $z = uvwxy$ deve essere vero che $|vwx| <= n$ e $vx != \varepsilon$:
Perché l'altezza di A' è <= m+1 (infatti in un albero di questa altezza c'è ripetizione) e quindi il suo prodotto vwx ha lunghezza $<= 2^m=n$

Inoltre il cammino da A' a A'' è costruito con produzione X->BC e ogni nonterminale deve produrre qualche terminale (per la CNF) e quindi $|vx|>0$

Basta ora osservare che dall'albero della figura precedente possiamo generare un numero infinito di alberi che sono tutti della grammatica di partenza:

--in A' possiamo incollare l'albero con radice A'', ottenendo l'albero con prodotto uwy

--possiamo incollare in A'' l'albero con radice A', ottenendo il prodotto uv^2wx^2y

--dall'albero appena costruito, potremmo di nuovo incollare in A'' l'albero con radice A' ottenendo il prodotto uv^3wx^3y

--in questo modo possiamo ottenere alberi di derivazione con prodotto uv^iwx^iy per qualsiasi $i>=0$.

pumping lemma

<u>Teorema:</u>

Sia L un CFL esiste una n t.c. se z è in L con $|z|>=n$, allora z= uvwxy t.c.

1) $|vwx| <=n$
2) vx diverso da ε
3) per ogni $i >=0$, uv^iwx^iy è in L

Esempio: $L=\{0^n1^n2^n \mid n>0\}$ non è CF

Per il pumping lemma, esiste n tale che ogni z più lunga di n è z=uvwxy. Scegliamo $z= 0^n1^n2^n$ sappiamo che $|vwx| <= n$ e quindi o contiene degli 0 o contiene dei 2, ma non entrambi.

Supponiamo che vwx non contenga 2.
Quindi ripetendo vx (non vuota) contiene almeno uno 0 o 1. Per cui uwy ha meno 0 e/o meno 1 di 2

analogo ragionamento mostra il caso in cui vwx non contiene 0