Grammatiche generative libere da contesto - I

Prof. A. Morzenti

aa 2008-2009

LE GRAMMATICHE GENERATIVE LIBERE DA CONTESTO

LINGUAGGI LIBERI DA CONTESTO (LINGUAGGI LIBERI)

LIMITI DEI LINGUAGGI REGOLARI

$$\boxed{\begin{array}{l} begin\ \ begin\ \ begin\ ...\ end\ \ end\ \ end \\ L_1 = \left\{ x \mid x = b^n e^n, n \geq 1 \right\} \\ L_1 \ \text{ non è regolare } \quad b^+ e^+ \quad (be)^+ \end{array}}$$

Manca annidamento

Manca vincolo #*b*=#*e*

SINTASSI - Per definire un linguaggio si possono usare delle ***regole di riscrittura***, che attraverso ripetute applicazioni, permettono di generare tutte e solo le frasi del linguaggio. L'insieme di tali regole costituisce una GRAMMATICA (O SINTASSI) GENERATIVA.

ESEMPIO - Palindromi

$$L = \left\{ uu^R \mid u \in \{a,b\}^* \right\} = \{\varepsilon, aa, bb, abba, baab, ..., abbbba, ...\}$$

$frase \rightarrow \varepsilon$ - una frase è una stringa vuota

$frase \rightarrow a \;\; frase \;\; a$ - una frase è una frase racchiusa tra a e a

$frase \rightarrow b \;\; frase \;\; b$ - una frase è una frase racchiusa tra b e b

Una catena di derivazioni:

$$frase \Rightarrow a \;\; frase \;\; a \Rightarrow ab \;\; frase \;\; ba \Rightarrow$$
$$\Rightarrow abb \;\; frase \;\; bba \Rightarrow abb\varepsilon bba = abbbba$$

NB: attenzione a distinguere i due *metasilboli*

$\rightarrow$ separa parti sinistra e destra di un regola

$\Rightarrow$ relazione di derivazione (riscrittura)

ESEMPIO: una lista non vuota di palindromi

$$abba\ bbaabb\ aa$$

$$lista \rightarrow\ frase\ \ lista \qquad frase \rightarrow \varepsilon$$

$$lista \rightarrow\ frase \qquad frase \rightarrow a\ \ frase\ \ a$$

$$frase \rightarrow b\ \ frase\ \ b$$

Simboli ***non terminali***: *lista* (assioma) e *frase* (definisce dei componenti – sottostringhe costituenti: i palindromi).

GRAMMATICA LIBERA DA CONTESTO (BNF - Backus Normal Form – di TIPO 2): è definita da quattro entità

1 *V, alfabeto non terminale,* è un insieme di simboli (detti metasimboli) nonterminali
2. *Σ, alfabeto terminale,* è l'insieme dei caratteri con cui sono fatte le frasi.
3. *P,* insieme delle *regole* (o produzioni) sintattiche
4. *S Є V,* un particolare non terminale detto *assioma*

Ogni regola di P è una coppia ordinata

$$(X,\alpha),\quad X \in V \ \text{ e } \ \alpha \in (V \bigcup \sum)^*$$

$$(X,\alpha) \in P \quad X \rightarrow \alpha$$

$$X \rightarrow \alpha_1, X \rightarrow \alpha_2, \ldots X \rightarrow \alpha_n$$

$$X \rightarrow \alpha_1 \,|\, \alpha_2 \,|\, \ldots \,|\, \alpha_n$$

$$X \rightarrow \alpha_1 \bigcup \alpha_2 \bigcup \ldots \bigcup \alpha_n$$

Per evitare confusioni i metasimboli '→','|','U','ε' non devono essere anche simboli; inoltre gli alfabeti terminali e non terminali devono essere disgiunti.

CONVENZIONI per distinguere simboli terminali e non terminali

- parentesi angolate:

<frase if> → if <cond> then <frase> else <frase>

- grasetto e corsivo:

frase_if → ***if*** *cond* ***then*** *frase* ***else*** *frase*
frase_if → ‘if’ cond ‘then’ frase ‘else’ frase

- maiuscolo vs. minuscolo:

F → if C then D else D

CONVENZIONI USATE NEL TESTO:

- Caratteri terminali - *{a, b, …}*
- Caratteri nonterminali - *{A, B, …}*
- Stringhe di Σ^* fatte di soli terminali - *{r, s, …, z}*
- Stringhe di $(V \cup \Sigma)^*$ fatte di simboli terminali e non - *{α, β,…}*
- Stringhe di soli nonterminali - σ

TIPI DI REGOLE (PD=parte destra, PS=parte sinistra)

terminale: la PD contiene terminali o la stringa vuota	$\to u \mid \varepsilon$
vuota (o nulla): la PD e' vuota	$\to \varepsilon$
iniziale: la PS e' l'assioma	$S \to$
ricorsiva: la PS compare nella PD	$A \to \alpha A \beta$
ricorsiva a sinistra: la PS e' prefisso della PD	$A \to A\beta$
ricorsiva a destra: la PS e' suffisso della PD	$A \to \beta A$
ricorsiva a destra e sinistra: intersezione dei due casi prec.	$A \to A\beta A$
copiatura o categorizzazione: laPD e' un nt singolo	$A \to B$
lineare: al piu' un nt nella PD	$\to uBv \mid w$
lineare a destra (tipo 3): come lineare, con nt suffisso	$\to uB \mid w$
lineare a sinistra (tipo 3): come lineare, con nt prefisso	$\to Bv \mid w$
normale di Chomsky: due nt o un solo terminale	$\to BC \mid a$
normaledi Greibach: un terminale seguito da nonterminali	$\to a\sigma \mid b$
a operatori: due nt separati dal terminale (operatore)	$\to AaB$

DERIVAZIONI E LINGUAGGIO GENERATO

Def. Relazione di derivazione $\Rightarrow$
$\beta, \gamma \in (V \cup \Sigma)^*$
γ deriva da β per la grammatica G
$\beta \underset{G}{\Rightarrow} \gamma$ se $A \to \alpha$ è una regola di G
e $\beta = hAd, \; \gamma = h\alpha d$

potenza, chiusura (riflessiva e) transitiva di $\Rightarrow$
$\beta_0 \overset{n}{\Rightarrow} \beta_n \quad \beta_0 \overset{*}{\Rightarrow} \beta_n \quad \beta_0 \overset{+}{\Rightarrow} \beta_n$

LINGUAGGIO GENERATO DA UN NONTERMINALE O DALL'ASSIOMA
$L_A(G) = \left\{ x \in \Sigma^* \mid A \overset{+}{\Rightarrow} x \right\}$
$L(G) = L_S(G) = \left\{ x \in \Sigma^* \mid S \overset{+}{\Rightarrow} x \right\}$

Se $A \Rightarrow^* \alpha$ α detta ***forma generata da G***

Se $S \Rightarrow^* \alpha$ α detta ***forma di frase***

Se $A \Rightarrow^* \alpha$ $\alpha \in \Sigma^*$ (α terminale) detta ***frase***

ESEMPIO (struttura di un libro). La grammatica G_l genera la struttura di un libro: esso contiene un frontespizio (*f*) e una serie (denotata da nt *A*) di uno o più capitoli, ognuno dei quali inizia con il titolo (*t*) del capitolo e contiene una serie (*B*) di una o piu' linee (*l*).

$$\boxed{\begin{array}{l} S \to fA \\ A \to AtB \mid tB \\ B \to lB \mid l \end{array}}$$

Partendo da *A* si genera la forma *tBtB* e la stringa *tlltl* $\in L_A(G_l)$
Partendo da *S* si generano le forme di frase *fAtlB, ftBtB*

Il linguaggio generato partendo da *B* è $L_B(G_l) = l^+$
$L(G_l)$ essendo generato da G_l è detto ***libero*** o ***libero dal contesto***
E' anche un linguaggio regolare essendo definito anche dall'espressione $f(tl^+)^+$

Due grammatiche G e G' sono equivalenti se generano lo stesso linguaggio, ossia se L(G) = L(G')

G_l

$$S \rightarrow fX$$
$$X \rightarrow XtY \mid tY$$
$$Y \rightarrow lY \mid l$$

G_{l2}

$$S \rightarrow fA$$
$$A \rightarrow AtB \mid tB$$
$$B \rightarrow Bl \mid l$$

Prime regole hanno n.t. rinominati

$$Y \underset{G_l}{\overset{n}{\Rightarrow}} l^n \quad \text{e} \quad B \underset{G_{l2}}{\overset{n}{\Rightarrow}} l^n \quad \text{generano lo stesso linguaggio } L_{\mathrm{B}} = l^+$$

GRAMMATICHE ERRONEE E REGOLE INUTILI: quando si scrive una grammatica si deve badare che tutti i nonterminali siano definiti e che ognuno di essi effettivamente contribuisca alla produzione del linguaggio.

UNA GRAMMATICA G E' PULITA (O RIDOTTA) se valgono le seguenti condizioni:

1. Ogni nonterminale A è *raggiungibile* dall'assioma, ossia esiste una derivazione

$$\boxed{S \overset{*}{\Rightarrow} \alpha A \beta}$$

2. Ogni nonterminale A è *definito*, ossia genera un linguaggio non vuoto

$$\boxed{L_A(G) \neq \varnothing}$$

PULIZIA DELLA GRAMMATICA: algoritmo in due fasi. La prima fase trova L'insieme INDEF dei nonterminali non definiti. La seconda quelli non raggiungibili.

FASE 1- Costruiamo l'insieme complementare DEF=V\INDEF

Inizializziamo DEF con i nt che generano immediatamente una stringa terminale

$$\boxed{DEF := \left\{A \mid (A \to u) \in P, \text{con } u \in \Sigma^{*}\right\}}$$

Si ripete la seguente trasformazione fino a raggiungere un punto fisso:

$$DEF := DEF \cup \{ B \mid (B \to D_1 D_2 \ldots D_n) \in P \wedge \forall i (D_i \in DEF \cup \Sigma)\}$$

Ogni D_i appartiene a DEF o è terminale

A ogni iterazione sono date due possibilità:

1) Si scoprono nuovi non terminali aventi come PD una stringa di elementi tutti definiti o terminali, oppure
2) Si termina

I nonterminali appartenenti a INDEF vengono eliminati.

FASE 2 - Il calcolo dei nonterminali raggiungibili da S si riconduce all'esistenza di un cammino nel grafo della relazione *produce*

$$\boxed{\begin{array}{l} A \; produce \; B \quad \text{se e solo se} \\ A \to \alpha B \beta \text{ con } A \neq B \;\; \alpha, \beta \text{ stringhe qualsiasi} \end{array}}$$

C è raggiungibile da S se, e solo se, esiste nel grafo un cammino da S a C.
I nonterminali non raggiungibili possono essere eliminati.

Spesso si aggiunge anche un terzo requisito di pulizia:

3. G non deve consentire *derivazioni circolari* che non sono essenziali e introducono ambiguità

$$\boxed{\begin{array}{l} A \overset{+}{\Rightarrow} A \\ \text{se } x \text{ e' derivata tramite } A \Rightarrow A \Rightarrow x \\ \text{e' derivabile anche tramite } A \Rightarrow x \end{array}}$$

ESEMPI DI GRAMMATICHE NON PULITE

$1)\ \{S \to aASb, A \to b\} \quad (S\ non\ genera\ alcuna\ frase)$

$2)\ \{S \to a, A \to b\}$ (A irraggiungibile) $\{S \to a\}$ (vers. equiv. pulita)

$3)\ \{S \to aASb \mid A, A \to S \mid c\}$ (circolare su A ed S) $\{S \to aSSb \mid c\}$ (vers. equiv. pulita)

LA CIRCOLARITÀ PUÒ NASCERE PER EFFETTO DI UNA REGOLA VUOTA

$$X \to XY \mid \ldots \quad Y \to \varepsilon \mid \ldots$$

ANCHE SE RIPULITA UNA GRAMMATICA PUÒ PRESENTARE REGOLE RIDONDANTI

(1,4) e (2,5) generano le stesse frasi

$$1.\ S \to aASb \qquad 4.\ A \to c$$
$$2.\ S \to aBSb \qquad 5.\ B \to c$$
$$3.\ S \to \varepsilon$$

RICORSIONE E INFINITEZZA DEL LINGUAGGIO

Quasi tutti i linguaggi artificiali sono infiniti. Da cosa dipende la capacità di una grammatica di generare dei linguaggi infiniti?

PER PRODURRE UN NUMERO ILLIMITATO DI FRASI È NECESSARIO CHE LE REGOLE PERMETTANO LA DERIVAZIONE DI FRASI DI LUNGHEZZA ILLIMITATA. <u>LA GRAMMATICA DEVE AVERE LA PROPRIETÀ DI RICORSIONE</u>

$$A \overset{n}{\Rightarrow} xAy \quad n \geq 1 \ \text{ e' ricorsiva}$$

$$\text{se} \quad n = 1 \ \text{ e' immediatamente ricorsiva}$$

$$A \ \text{ e' non terminale ricorsivo}$$

$$\text{se} \ \ x = \varepsilon \ \text{ e' ricorsiva sinistra}$$

$$\text{se} \ \ y = \varepsilon \ \text{ e' ricorsiva destra}$$

CONDIZIONE NECESSARIA E SUFFICIENTE perché sia infinito il linguaggio L(G), dove G è una grammatica pulita e priva di derivazioni circolari, è che G permetta delle derivazioni ricorsive.

CONDIZIONE NECESSARIA: se non vi fossero derivazioni ricorsive, ogni derivazione avrebbe una lunghezza limitata e L(G) sarebbe finito.

CONDIZIONE SUFFICIENTE:

La presenza di $A \overset{n}{\Rightarrow} xAy$ assicura che $A \overset{+}{\Rightarrow} x^m A y^m$

per m ≥ 1 arbitrario con x e y non contemp. vuote

Inoltre G pulita implica

$S \overset{*}{\Rightarrow} uAv$ (A raggiungibile da S)

e $A \overset{+}{\Rightarrow} w$ (derivazione di A va a buon fine)

Quindi esistono non terminali che generano un linguaggio infinito

$$S \overset{*}{\Rightarrow} uAv \overset{+}{\Rightarrow} ux^m A y^m v \overset{+}{\Rightarrow} ux^m w y^m v, (m \geq 1)$$

UNA GRAMMATICA È PRIVA DI RICORSIONI
SE E SOLO SE
IL GRAFO DELLA RELAZIONE *produce* È PRIVO DI CICLI

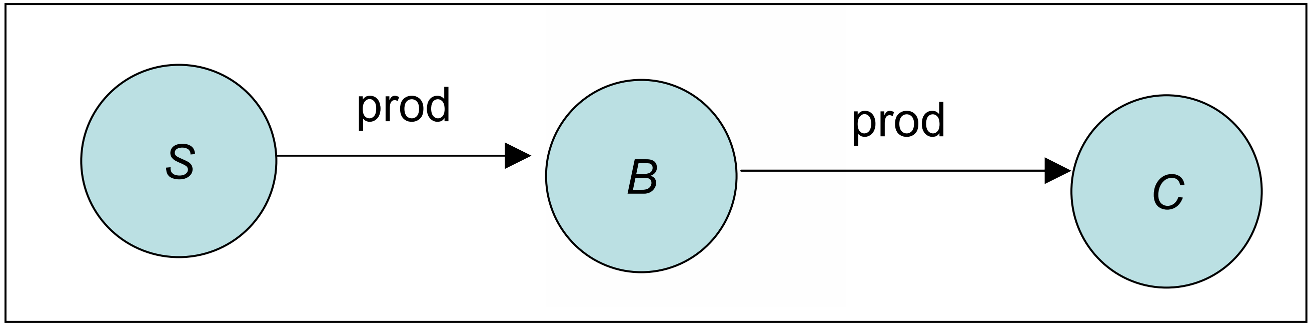
ESEMPIO (Linguaggio finito)

G genera un il linguaggio finito

$$\boxed{\{aabc, acac\}}$$

G

$$S \to aBc$$
$$B \to ab \mid Ca$$
$$C \to c$$



ESEMPIO (espressioni aritmetiche)

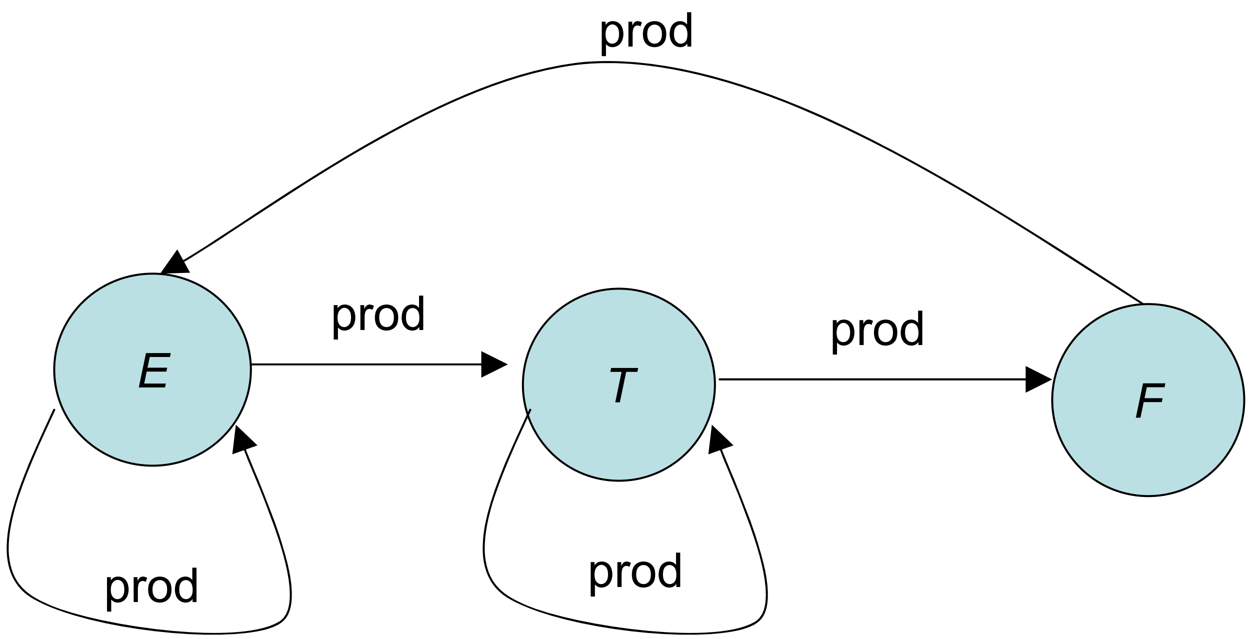
$$G = \left(\{E, T, F\}, \{i, +, *,), (\}, P, E\right)$$

$$E \to E + T \mid T \qquad T \to T * F \mid F \qquad F \to (E) \mid i$$

$$L(G) = \{i, i + i + i, i * i, (i + i) * i, \ldots\}$$

F (fattore) ha ricorsione indiretta (non immediata)
E (espressione) ha ricorsioni immediata di tipo sinistro e non immediata
T (termine) ha ricorsioni immediata di tipo sinistro e non immediata

G è pulita e non circolare, quindi il linguaggio generato è infinito.


prod
E
prod
T
prod
F
prod
prod