Luigi Rizzi

Linguistica computazionale

I. Elementi di base sulle grammatiche formali

(1) Linguaggio L = insieme di stringhe, sequenze finite di elementi del vocabolario

(2) Dato un vocabolario A, l'insieme di tutte le stringhe costruite su A (A*) più l'operazione di concatenazione (formazione di stringhe complesse a partire da stringhe semplici) costituisce un monoide.

(3) Normalmente, non tutte le stringhe del monoide costituiscono frasi ben formate di L: per es, dato un frammento del lessico italiano, non tutte le sequenze arbitrarie sono frasi dell'italiano:

Il bambino corre
Corre il bambino
*il corre bambino
* bambino corre il
…..

Quindi L con vocabolario A è in genere un sottoinsieme proprio di A*; compito della grammatica di L è di catturare il sottoinsieme in questione generando tutte e sole le stringhe che lo compongono.

Per lo studio delle proprietà formali, è utile considerare linguaggi semplificati, con lessici molto ridotti e sintassi semplice, per es, linguaggi come i seguenti:

ab, aabb, aaabbb, aaaabbbb...
aa, bb, abba, baab, abaaba,...
aa, abab, abbababbab,...

(4) Una grammatica formale è un sistema deduttivo di assiomi e regole di inferenza, che genera le frasi della lingua come suoi teoremi.

(5) Una grammatica formale è definita da una quadrupla :

(i) Un vocabolario terminale V_T
(ii) Un vocabolario non terminale V_N
(iii) Un assioma, o simbolo iniziale $S \in V_N$
(iv) Un insieme di regole di riscrittura $\varphi \rightarrow \psi$

(6) Nella sua forma più generale, la parte sinistra e la parte destra della regola di riscrittura sono stringhe qualsiasi costruite sui due vocabolari, con la sola restrizione che la parte sinistra deve contenere almeno un simbolo di V_N .

(7) Una derivazione è una sequenza di stringhe che parte dall'assioma e arriva fino a generare una stringa del linguaggio tramite le regole, eliminando via via i simboli non terminali.

Esempio:

V_T : {a, b}

V_N : {S, A, B}

Assioma: S

Regole:

S → A B S
S → e
AB → BA
BA → AB
A → a
B → b

Esempio di derivazione:

S
ABS
BAS
BAABS
BAAB
bAAB
baAB
baaB
baab

(8) “e” è la stringa vuota, l’elementi di identità rispetto alla concatenazione: eφ = φe = φ

II. Alberi

(9) Se si pone la restrizione che la parte sinistra della regola consti di un solo simbolo non terminale, la derivazione si può esprimere in forma di albero.

(10) Dominanza (riflessiva, transitiva, antisimmetrica (se x domina y, allora y domina x solo se x=y) e precedenza (irriflessiva, transitiva, asimmetrica (se x precede y, allora y non precede x))

(11) Condizione di esclusività: due nodi qualsiasi in un albero sono in rapporto di precedenza o in rapporto di dominanza.

(12) Condizione di non-incrocio: in un albero, se x precede y, allora tutti i nodi dominati da x precedono tutti i nodi dominati da y.

(13) x e y sono nodi sorella sse sono immediatamente dominati dallo stesso nodo z (madre)

(14) C-comando: x c-comanda y sse il nodo sorella di x domina y

(15) Assioma di corrispondenza lineare: siano X, Y, nonterminali, e x, y, terminali tali che X domina x e Y domina y; allora, se X c-comanda asimmetricamente Y, x precede y. (Kayne 1994)

(16) Siccome lo specificatore c-comanda asimmetricamente il complemento, segue da (15) che lo specificatore precede il complemento.

III. La gerarchia di Chomsky

(17) Chomsky (1956, 59) ha osservato che ponendo restrizioni via via più forti sulla forma delle regole si può stabilire una gerarchia di grammatiche di potere generativo decrescente

Tipo 0: sistemi di riscrittura non ristretti: $\varphi \rightarrow \psi$, con $\varphi \neq \varepsilon$ (alternativamente, con φ contenente almeno un simbolo non terminale.

Tipo 1: sistemi di riscrittura contestuali (context-sensitive): $A \rightarrow \psi / \alpha\_\_\_\beta$, con $\psi \neq \varepsilon$.

Tipo 2: sistemi di riscrittura acontestuali (context free): $A \rightarrow \psi$.

Tipo 3: sistemi regolari: $A \rightarrow xB$, $A \rightarrow x$

(18) Nelle regole dei linguaggi di tipo 0 qualunque stringa non nulla può essere riscritta come qualunque stringa (inclusa la stringa nulla). Questi sistemi di riscrittura caratterizzano i linguaggi ricorsivamente enumerabili, quelli che possono essere generati da una Macchina di Turing.

(19) Un altro modo di concettualizzare le regole dei sistemi di tipo 1 consiste nel notare che riscrivono una sequenza di simboli in un'altra sequenza di simboli non decrescente: cioè l'output ha almeno tanti simboli quanto l'input.

(20) I sistemi di tipo 2 riscrivono un simbolo non-terminale come qualunque stringa di terminali o non-terminali, inclusa la stringa nulla.

(21) I sistemi di tipo 3 hanno un simbolo non-terminale a sinistra della freccia, e lo riscrivono come una stringa di simboli con un solo non-terminale in fondo (regole lineari a destra).

(22) valgono le seguenti relazioni insiemistiche tra i linguaggi generati dai diversi tipi di grammatiche:

(i) i linguaggi di tipo 3 sono propriamente inclusi nei linguaggi di tipo 2;
(ii) i linguaggi di tipo 2 non contenenti ε sono propriamente inclusi nei linguaggi di tipo 1;
(iii) i linguaggi di tipo 1 sono propriamente inclusi nel linguaggi di tipo 0.

Alcuni esempi:

(23) $L = \{x \in \{a, b\}^* \mid$ x contiene un egual numero di a e di b, in qualsiasi ordine$\}$

aaababababbba

G = < {a, b}, {S, A, B}, S, R>

R =
S → aB
S → e
S → bA
B → b
B → bS
A → a
A → aS
A → bAA
B → aBB

Questo linguaggio è di tipo 2 e 0; non è di tipo 1 perché ha una regola „shrinking“, né di tipo 3.

(24) L = {x ∈ {a, b}* | x = $a^n b^n$ (n ≥ 0)}

aaaaaabbbbbb

G = < {a, b}, {S}, S, R>

R =
S → aSb
S → e

Linguaggio di tipo 2 e 0. Non può essere di tipo 3, intuitivamente, perché una volta generata la stringa di a la grammatica non ha modo di “ricordarsi” quante occorrenze di a ha prodotto per riprodurle con b.

(25) L = {x ∈ {a, b}* | x = yz, in cui z è l’immagine speculare di y}

abbababbababba

G = < {a, b}, {S}, S, R>

R =
S → aSa
S → bSb
S → aa
S → bb

Linguaggio di tipo 2, 1, 0, ma non 3

(26) L = {x ∈ {a, b}* | x = $a^n b^m$}

aaaabbbbbbbb

G = < {a, b}, {S, T}, S, R>

R =
$S \rightarrow aS$
$S \rightarrow bT$
$T \rightarrow bT$
$S \rightarrow a$
$S \rightarrow b$
$T \rightarrow b$

Questo è un linguaggio di tipo 3, visto che, una volta passato dalla generazione di a alla generazione di b, non ha il problema di "ricordarsi" il numero delle occorrenze.

Pumping Lemma

(27) Come possiamo sapere se un dato linguaggio non è regolare?

(28) Pumping Lemma: Supponiamo che L sia un linguaggio regolare infinito. Allora qualunque stringa di L deve poter essere divisa in tre parti x, y, z, tali che la stringa $x\ y^n\ z$ (ottenuta "pompando" il segmento di mezzo) apartenga a L.

(30) Quindi, a^nb^n ($n \geq 0$) non è un linbguaggio regolare: prendiamo una stringa, per es aaaaabbbbb

1. se segmentiamo aaa – aa – bbbbb, "pompando" il segmento di mezzo perdiamo l'egual numero di a e b;

2. se segmantiamo aaaaa – bb – bbb, vale la stessa conclusione;

3. se segmentiamo aaaa – ab – bbbb, perdiamo il fatto che una sequenza continua di a sia seguita da una sequenza continua di b;
QED

Le lingue naturali sono regolari?

(31) L'inglese non è una lingua a stati finiti (regolare) (Chomsky 1956, 57, 59)

(32)a If S1 then S2
b Either S3 or S4
c The man who said S5 is arriving today

(33)Queste opzioni danno luogo a strutture di tipo xx^R che non sono regolari. Più semplicemente, ci sono nelle lingua naturali delle strutture di tipo a^nb^n , che già sappiamo non essere regolari per il Pumping Lemma:

(34) Se è vero che, se la ditta è in crisi ci sarà una riduzione di personale, allora ci sarà uno sciopero.

(35) Se … se … se … allora … allora … allora

Le lingue naturali sono contex-free?

(35) Lingue con due identiche stringhe concatenate (xx) non sono context free

(36)Le lingue umane presentano dipendenze attraverso serie di tipo

X1, X2, …Xn,… Y1, Y2, … Yn

Per esempio,

(37) Gianni, Maria, Francesca e Guido sono stati rispettivamente promosso, bocciata, promossa, bocciato

Con il participio n accordantesi con il nome n.

Complessità nella competenza e nell'esecuzione

(37)Le lingue naturali ammettono certe strutture grammaticali che sono difficili da utilizzare. Un caso classico è l'autoincassamento:

(38)Il cane ha inseguito il gatto che ha catturato il topo che ha mangiato il formaggio

(39) Il formaggio che il topo che il gatto che il cane ha inseguito ha catturato ha mangiato era sul tavolo

Gianni è consapevole del fatto che Mario tema la possibilità che
(40) La nonna continuerà a dormire poiché il padre darà una mano se la madre si sveglia quando il bambino piange

(41) Poiché se quando il bambino piange la madre si sveglia il padre darà una mano, la nonna continuerà a dormire