La jerarquía de Chomsky: Donde los árboles dejan ver el bosque

Carlos Martín-Vide

Grammar

A (formal) grammar is a construct $G = (N,T,S,P)$, where:

- N, T are alphabets (nonterminal and terminal), with $N \cap T = \emptyset$,
- $S \in N$ (axiom), and
- P is a finite set of productions (w,v) such that w, $v \in (N \cup T)^*$ and w contains at least one letter from N. [(w,v) is usually written $w \to v$.]

Immediate derivation

Given $G = (N,T,S,P)$ and $w, v \in (N \cup T)^*$, an immediate or direct derivation (in 1 step) $w \Rightarrow_G v$ holds iff:

- there exist $u_1, u_2 \in (N \cup T)^*$ such that $w = u_1 \alpha u_2$ and $v = u_1 \beta u_2$, and
- there exists $\alpha \rightarrow \beta \in P$.

Derivation

Given $G = (N,T,S,P)$ and $w, v \in (N \cup T)^*$, a derivation $w \Rightarrow^*_G v$ holds iff:

- either $w = v$, or
- there exists $z \in (N \cup T)^*$ such that $w \Rightarrow^*_G z$ and $z \Rightarrow^*_G v$.

  [$\Rightarrow^*_G$ denotes the reflexive transitive closure and $\Rightarrow^+_G$ the transitive closure, respectively, of $\Rightarrow_G$.]

Language

The language generated by a grammar is the set:

$$L(G) = \{w : S \Rightarrow^*_G w \text{ and } w \in T^*\}$$

Only infinite languages are interesting.

For any natural language:

- The set of phonemes is finite (and small).
- The set of words is finite (and large) if some "special words" are excluded.
- The set of sentences is infinite (but how large?).

Types of grammars

Grammars can be classified according to different criteria. The most usual one is the form of their productions

Unconstrained grammar

G is 0 or RE iff there are no restrictions on the form of the productions: everything at the left-hand side and the right-hand side of the rules is allowed.

Context-sensitive grammar

G is 1 or CS iff every production is of the form:

$u_1 A u_2 \rightarrow u_1 w u_2$

with u_1, u_2, $w \in (N \cup T)^*$, $A \in N$ and $w \neq \lambda$ (except possibly for the rule $S \rightarrow \lambda$, in which case S does not occur on any right-hand side of a rule).

Context-free grammar

G is 2 or CF iff every production is of the form:

$A \rightarrow w$

with $A \in N$, $w \in (N \cup T)^*$.

Regular (finite-state) grammar

G is 3 or REG iff every production is of any of the forms:

$A \rightarrow wB$ (or $A \rightarrow Bw$)

$A \rightarrow w$

with $A, B \in N$, $w \in T^*$.

Language family

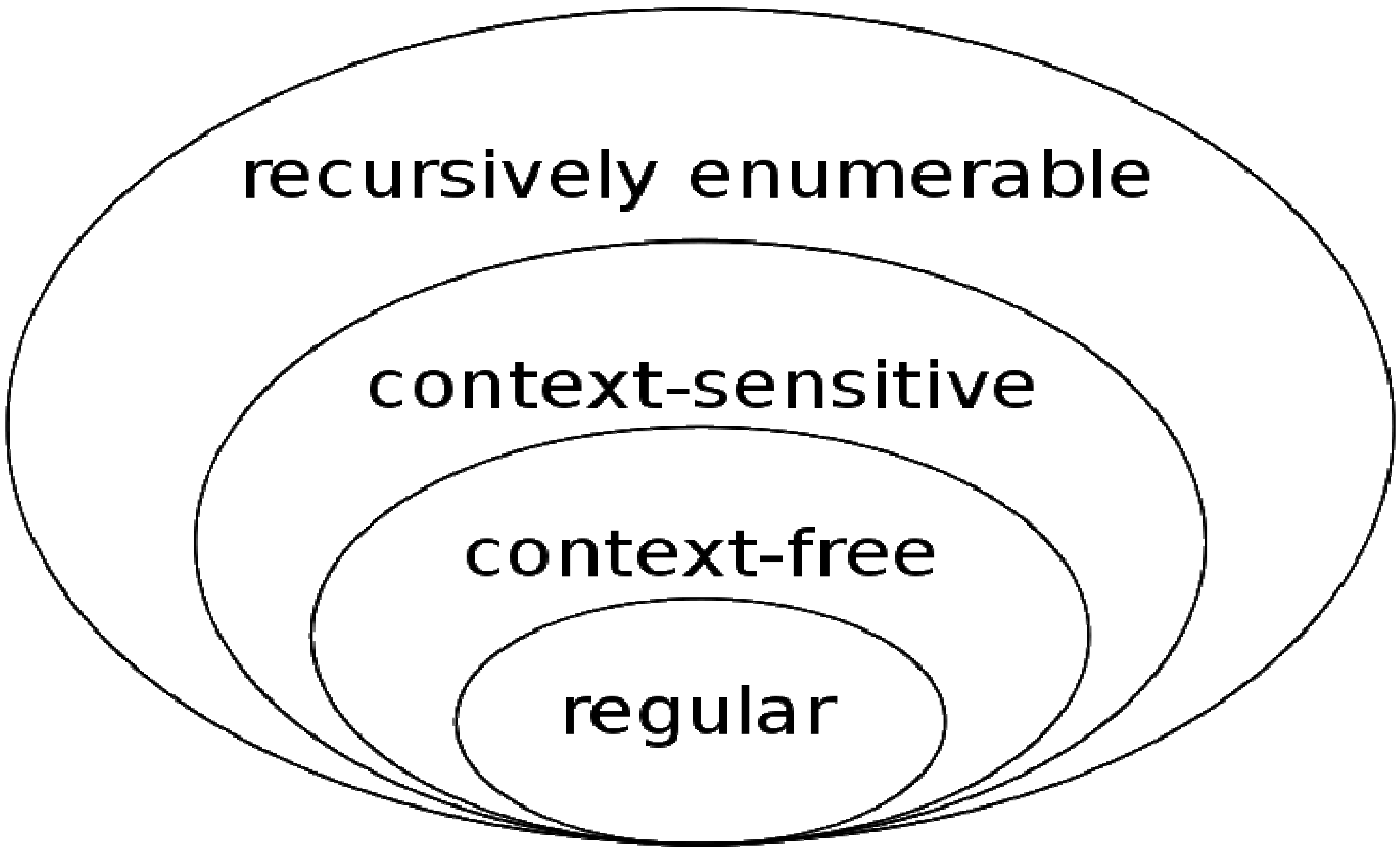
A language is of type i (i = 0, 1, 2, 3) if it is generated by a type i grammar.

The family of all type i languages is denoted by L_i.

[Note that while every grammar generates a unique language, one language can be generated by several different grammars.]

Chomsky hierarchy of languages

$$L_3 \subset L_2 \subset L_1 \subset L_0$$



Where natural languages are in the Chomsky hierarchy?

- Concentric location: mildly context-sensitive (various formalisms: TAG, HG, LIG, CCG...)
- Orthogonal

Grammar equivalence

Two grammars are said to be:

- (weakly) equivalent if they generate the same string language,
- strongly equivalent if they generate both the same string language and the same tree language. [each one of the trees is associated with one string and represents the way how the string is derived in the grammar]

Derivation tree

A derivation tree is defined as T = (V,D), where V is a set of nodes or vertices and D is a dominance relation, which is a binary relation in V that satisfies:

- (i) D is a weak order:
  - (i.a) reflexive: for every a ∈ V : aDa,
  - (i.b) antisymmetric: for every a, b ∈ V , if aDb and bDa, then a = b,
  - (i.c) transitive: for every a, b, c ∈ V , if aDb and bDc, then aDc.
- (ii) root condition: there exists r ∈ V such that for every b ∈ V : rDb,
- (iii) nonbranching condition: for every a, a′, b ∈ V , if aDb and a′Db, then aDa′ or a′Da.

Special cases of dominance

For every a, b $\in$ V :

a strictly dominates b (aSDb) iff aDb and a $\neq$ b; hence SD is a strict order in V :

(i) irreflexive: it is not the case that aSDa,

(ii) asymmetric: if aSDb, then it is not the case that bSDa,

(iii) transitive: if aSDb and bSDc, then aSDc.

a immediately dominates b (aIDb) iff aSDb and there does not exist any c such that aSDc and cSDb.

Degree of a node

The degree of a node is:

$$deg(b) = |\{a \in V : bIDa\}|.$$

Consequences:

- b is a terminal node or a leaf iff $deg(b) = 0$,
- b is a unary node iff $deg(b) = 1$,
- b is a branching node iff $deg(b) > 1$,
- T is an n-ary derivation tree iff all its nonterminal nodes are of degree n.

Independent nodes

Two nodes a, b are independent of each other (aINDb) iff neither aDb nor bDa.

Family relations among nodes

a is a mother node of b (aMb) iff aIDb.

a is a sister node of b (aSb) iff there exists c such that cMa and cMb.

The mother relation has the following features:

(i) there does not exist any a $\in$ V such that aMr, and

(ii) if b $\neq$ r, then it has just one mother node.

Derivation subtree (constituent)

Given $T = (V,D)$, for every $b \in V$, a derivation subtree or a constituent is:

$T_b = (V_b,D_b)$

where $V_b = \{c \in V : bDc\}$ and xD_by iff $x \in V_b$ and $y \in V_b$ and xDy.

C-command

Given T = (V,D), for every a, b ∈ V : a c-commands b (aCCb) iff:

(i) aINDb,

(ii) there exists a branching node that strictly dominates a, and

(iii) every branching node that strictly dominates a dominates b.

Asymmetric c-command

a asymmetrically c-commands b iff aCCb and it is not the case that bCCa

Preservation and isomorphism of derivation trees

Given two derivation trees $T = (V,D)$, $T' = (V',D')$ and $h : V \rightarrow V'$:

h preserves D iff for every $a, b \in V$: $aDb \rightarrow h(a)D'h(b)$.

h is an isomorphism of T in T′ ($T \approx T'$) iff h is a bijection and preserves D.

[Note that a mapping $f : A \rightarrow B$ is a bijection iff:

(i) f is one-to-one or injective: for every $x, y \in A$, if $x \neq y$ then $f(x) \neq f(y)$ or, equivalently, if $f(x) = f(y)$ then $x = y$, and

(ii) f is onto or exhaustive: for every $z \in B$, there exists $x \in A$ such that $f(x) = z$.]

Isomorphic derivation trees

Any two isomorphic derivation trees share all their properties:

- aSDb iff h(a)SD′h(b),
- aIDb iff h(a)ID′h(b),
- deg(a) = deg(h(a)),
- aCCb iff h(a)CCh(b),
- a is the root of T iff h(a) is the root of T′,
- depth(a) = depth(h(a)),
  [depth(a) = |{b ∈ V : bDa}| − 1]
- height(T) = height(T′).
  [height(T) = max{depth(a) : a ∈ V }]

Labelled derivation tree

Once one has an T = (V,D), one may enrich its definition to get a labelled derivation tree:

T = (V,D,L)

where (V,D) is a derivation tree and L is a mapping from V to a specified set of labels.

Isomorphism of labelled derivation trees

Given $T = (V,D,L)$ and $T' = (V',D',L')$, one says $T \approx T'$ iff:

(i) $h : V \to V'$ is a bijection,

(ii) h preserves D,

(iii) for every $a, b \in V$: $L(a) = L(b)$ iff $L'(h(a)) = L'(h(b))$.

Terminally ordered derivation tree

A terminally ordered derivation tree is T = (V,D,<), where (V,D) is a derivation tree and < is a strict total (or linear) order on the terminal nodes of V, i.e. a relation that is:

(i) irreflexive: for every terminal a, it is not the case that a < a,

(ii) asymmetric: if a < b, then it is not the case that b < a,

(iii) transitive: if a < b and b < c, then a < c, and

(iv) connected: either a < b or b < a.

Precedence

Given T = (V,D,<), for every b, c, d, e ∈ V : b <′ c (b precedes c) iff:

if bDd, d is terminal, cDe and e is terminal, then d < e.

Exclusivity condition

The following exclusivity condition completely orders a tree: Given T = (V,D,<), for every b, d ∈ V , if bINDd, then either b <′ d or d <′ b).

Consequence:

Every two nodes of the tree must hold one, and only one, of the dominance and precedence relations.

Gracias