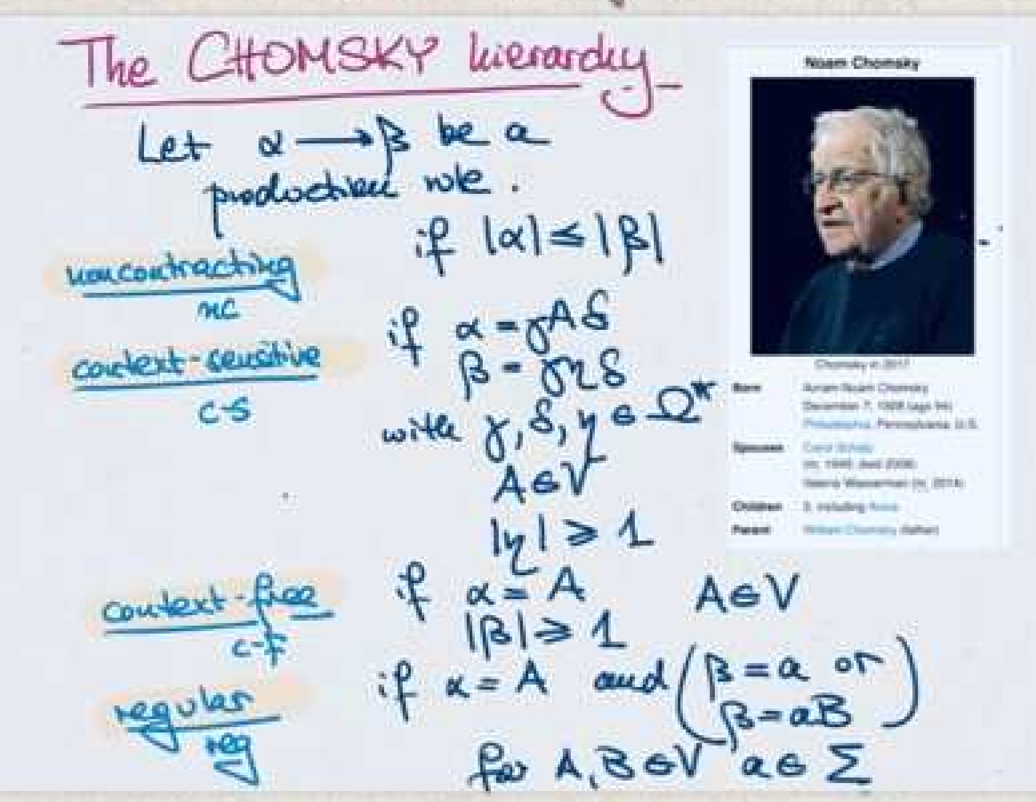
Automata & Formal Languages

13 October 2023

Lecture III, page 7



The CHOMSKY hierarchy

Let $\alpha \longrightarrow \beta$ be a production rule.

- noncontracting (nc): if $|\alpha| \leq |\beta|$
- context-sensitive (c-s): if $\alpha = \gamma A \delta$, $\beta = \gamma \eta \delta$ with $\gamma, \delta, \eta \in \Omega^*$, $A \in V$, $|\eta| \geq 1$
- context-free (c-f): if $\alpha = A$, $A \in V$, $|\beta| \geq 1$
- regular (reg): if $\alpha = A$ and $\left(\beta = a \text{ or } \beta = aB\right)$ for $A, B \in V$, $a \in \Sigma$

Grammar is nc/c-s/c-f/reg if all of its rules are.

Language L is type 0 if $\exists G$ s.t. $\mathcal{L}(G) = L$

type 1 [context-sensitive] if $\exists G$ c-s s.t. $\mathcal{L}(G) = L$

type 2 [context-free] if $\exists G$ c-f. s.t. $\mathcal{L}(G) = L$

type 3 [regular] if $\exists G$ reg s.t. $\mathcal{L}(G) = L$

Lecture III, page 9

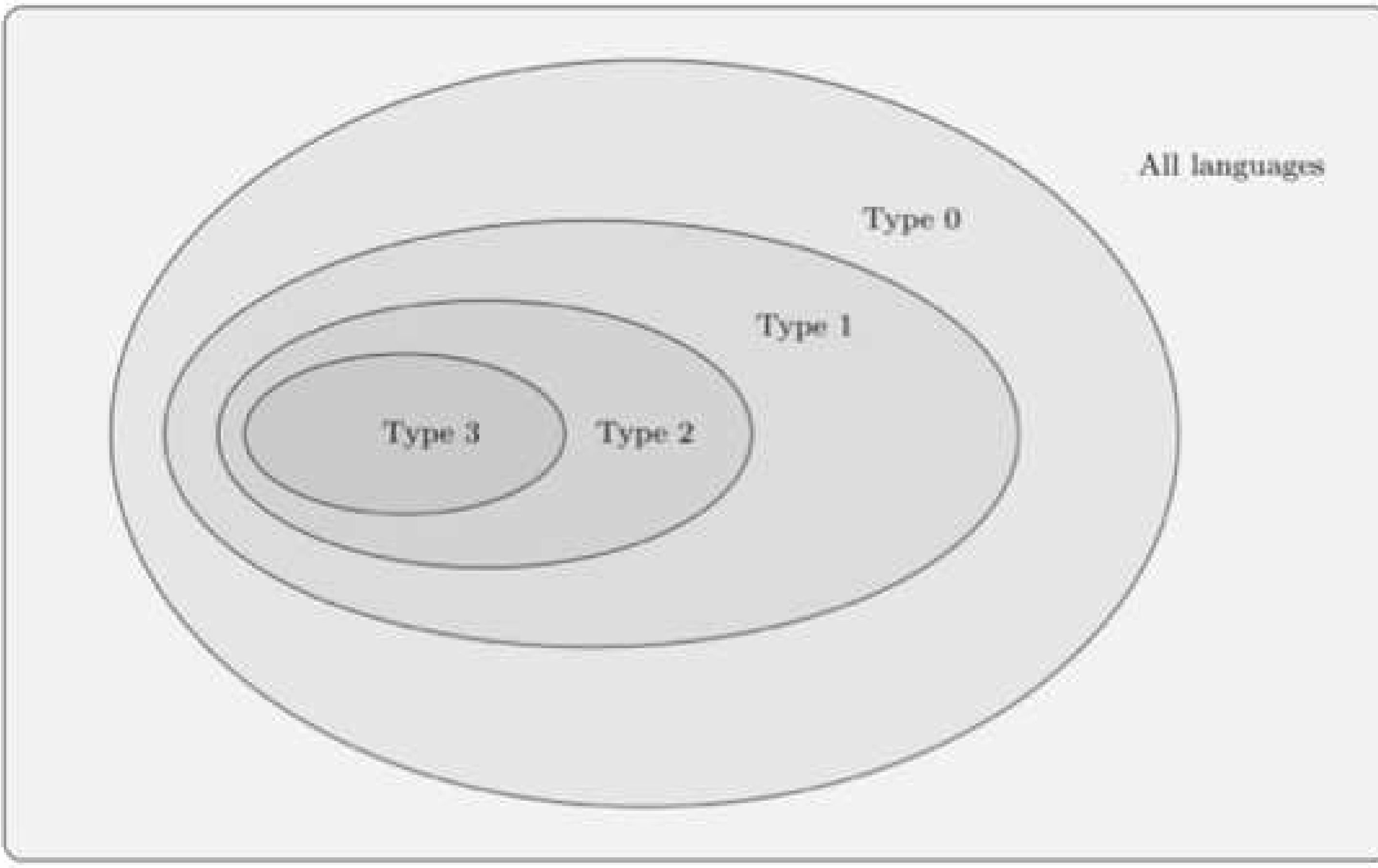


Figure 1: The Chomsky hierarchy

E.g., L is type 2 if THERE IS G context-free s.t. $L = \mathcal{L}(G)$.
L is type 3 if THERE IS G regular s.t. $L = \mathcal{L}(G)$.

$L = \{a^{2n+1} ; n \in \mathbb{N}\}$

REMEMBER THAT WE HAD EIGHT GRAMMARS FOR L.

An analysis of the argument in Example 1.9 shows that if in a grammar all production rules preserve oddness of length and we can provide a derivation of a^{2n+1}, then the grammar will produce the same language. E.g., $G_i = (\{a\}, \{S\}, P_i, S)$ with

$P_0 := \{S \to a, S \to aaS\}$ c-f

$P_1 := \{S \to aSa, S \to a\}$, c-f
$P_2 := \{S \to Saa, S \to a\}$, c-f
$P_3 := \{S \to aaS, S \to aaSaa, S \to a\}$, c-f
$P_4 := \{S \to aaS, S \to Saa, S \to aSa, S \to a\}$, c-f
$P_5 := \{S \to aaS, aSa \to aaa, S \to a\}$, c-s
$P_6 := \{S \to aaS, aaS \to aSa, S \to a\}$, or noncontracting
$P_7 := \{S \to aaS, aaS \to a, S \to a\}$, etc. contracting

All grammars G_0 to G_7 are equivalent & produce the same language L.

Even more is true:
there is a regular grammar G_8 s.t.
$L = \mathcal{L}(G_8)$ [Example 1.17]

Lecture III, page 9:

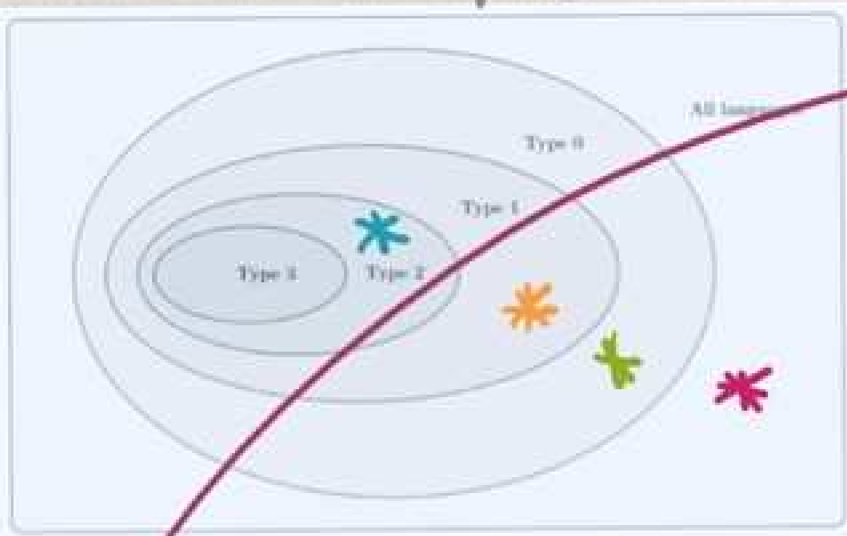


Figure 1: The Chomsky hierarchy

Chomsky defined:

L is type 0 if there is G s.t. L = $\mathcal{L}(G)$.

L is type 1 if it is context-sensitive

—"— 2 ______ context-free

—"— 3 ______ regular.

Observe ① Noncontracting languages are missing. Why?

② Question: Is the diagram PROPER, i.e., is each area in the picture populated by example.

① The reason for this is

THEOREM (Chomsky)

L is noncontracting

$\Longleftrightarrow$

L is context-sensitive.

Cf. Example Sheet #1.

② Properties of a Venn diagram:

find a language that is not type 0 *

find language type 0, but not type 1 *

find language type 1 not type 2 *

find language type 2 not type 3 *

To prove results like this, we need tools to prove that something cannot be done by a particular type of grammar.

*: follows from the fact that there are uncountably many languages, but only countably many type 0 languages.

DECISION PROBLEMS FOR GRAMMARS

G, G' grammars, $w \in W$

	INPUT	QUESTION
WORD PROBLEM	G, w	$\boxed{w \in \mathcal{L}(G)}$?
EMPTINESS PROBLEM	G	$\boxed{\mathcal{L}(G) = \emptyset}$?
EQUIVALENCE PROBLEM	G, G'	$\boxed{\mathcal{L}(G) = \mathcal{L}(G')}$?

A problem is solvable if there is an algorithm that takes the input and produces the correct answer to the question.

PREVIEW : We'll show that all three, in full generality (i.e., for type 0 languages) are unsolvable.

What about restricted versions ?

WORD PR.
EMPTINESS PR for type i languages ?
EQUIV. PR.

Observe (*) If such a problem is solvable for type i & $j \geq i$, then it's solvable for type j.

THEOREM The word problem for noncontracting grammars is solvable.

Corollary The word problem for type 1, 2, 3 grammars is solvable.

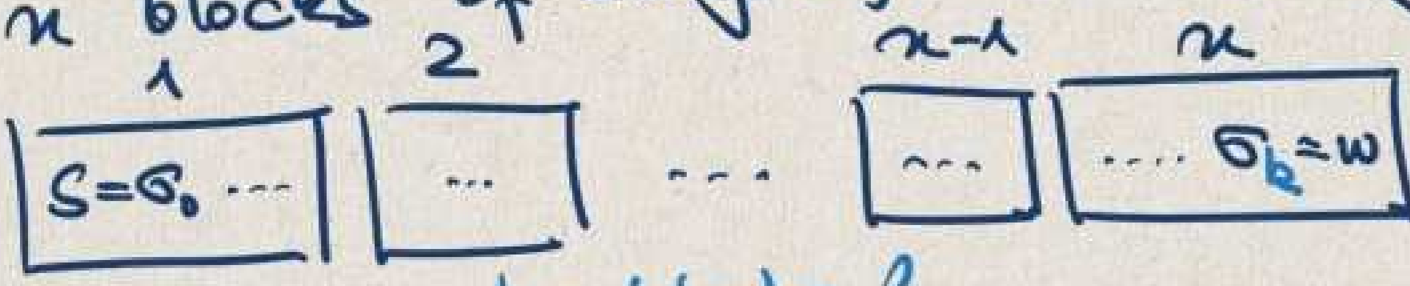
[Follows directly from Thm & (*).]

Proof of Thm Let G be noncontracting & $w \in W$.

If $w \in \mathcal{L}(G)$, there is a derivation

$$S = \sigma_0 \xrightarrow{G}_1 \sigma_1 \xrightarrow{G}_1 \cdots \xrightarrow{G}_1 \sigma_k = w.$$

Since $|w| = n$ for some n and $|S| = 1$ and G is non-contracting, there are at most n blocks of strings of the same length.



How long are the blocks?

In general, we have no idea.

E.g., if $S \to A$, $A \to S$ both in P, then arbitrarily long block 1 instances exist.

However, these must include repeats.

Better question:
How many <u>different</u> string can show up in block ℓ, $1 \le \ell \le k$.

$$|\Omega|^{\ell}.$$

<u>Observation</u> If $\sigma_0, \dots, \sigma_k$ is a derivation of σ_k of minimal length, then it cannot contain repeats.

[If $\sigma_i = \sigma_{i+j}$, shorten the derivation by deleting $\sigma_i, \dots, \sigma_{i+j-1}$.]

So, minimal derivations of w with $|w| = n$ have length at most

$$N := \sum_{i=1}^{n} |\Omega|^{i}.$$

<u>ALGORITHM</u>
1. Given G, w, determine $|\Omega|$, $|w|$ and calculate N.
2. Write down all sequences of strings of length $\le N$ that end in w and check whether they are G-derivations.
3. If one of them is: YES!
4. If not: NO!

q.e.d.

1.7 Closure properties

There are a number of algebraic operations on languages that allow us to combine languages to new languages. Let $L, M \subseteq \mathbb{W}^+$ be any languages over an alphabet Σ.

(a) *Concatenation.* The language LM consists of words vw such that $v \in L$ and $w \in M$.

(b) *Union.* The language $L \cup M$ consists of words either in L or in M.

(c) *Intersection.* The language $L \cap M$ consists of words that are both in L and M.

(d) *Complement.* The language $\overline{L} := \mathbb{W}^+ \backslash L$ consists of nonempty words that are not in L.

(e) *Difference.* The language $L \backslash M$ consists of words in L that are not in M.

$\mathcal{C}$ is closed under one of the operations if the operation transforms elts of $\mathcal{C}$ into elts of $\mathcal{C}$.

$LM := \{wv;\ w \in L \ \&\ v \in M\}$

Lemma 1.19. Let $\mathcal{C}$ be a class of languages. Then the following implications hold:

(a) If $\mathcal{C}$ is closed under union and complementation, then it is closed under intersection.

(b) If $\mathcal{C}$ is closed under intersection and complementation, then it is closed under union.

(c) If $\mathcal{C}$ is closed under intersection and complementation, then it is closed under difference.

(d) If $\mathbb{W}^+ \in \mathcal{C}$ and $\mathcal{C}$ is closed under difference, then it is closed under complementation.

Proof. These are all set algebra consequences of the definitions and de Morgan's Laws $\mathbb{W}^+ \backslash (A \cap B) = \mathbb{W}^+ \backslash A \cup \mathbb{W}^+ \backslash B$ and $\mathbb{W}^+ \backslash (A \cup B) = \mathbb{W}^+ \backslash A \cap \mathbb{W}^+ \backslash B$. Q.E.D.

Let $G = (\Sigma, V, P, S)$ and $G' = (\Sigma, V', P', S')$ be two grammars over the same alphabet Σ.

(a) *Concatenation.* The *concatenation grammar of* G *and* G' is $(\Sigma, V \cup V' \cup \{T\}, P^*, T)$ with a new variable T and $P^* := \{T \to SS'\} \cup P \cup P'$.

(b) *Union.* The *union grammar of* G *and* G' is $(\Sigma, V \cup V' \cup \{T\}, P^*, T)$ with a new variable T and $P^* := \{T \to S, T \to S'\} \cup P \cup P'$.

(1) These are the obvious choices for a concatenation & union grammar.

(2) Obvious:

a. If H is the concatenation grammar of G, G', then
$$\mathcal{L}(H) \supseteq \mathcal{L}(G)\mathcal{L}(G')$$

b. H union gr. of G, G'
$$\mathcal{L}(H) \supseteq \mathcal{L}(G) \cup \mathcal{L}(G').$$

(3) Since $T \to SS'$, $T \to S$, $T \to S'$ are context-free rules, if G, G' are c-f/c-s, then so are the union and concatenation grammars.

(4) Not so for "regular". [Chapter 2]

(5) Unfortunately, we can't always show equality in (2).

[Cf. Example Sheet #1.]

Def A rule $\alpha \longrightarrow \beta$ is called variable-based if $\alpha \in V^*$.

Remark. For every grammar, there is a v-b grammar equivalent.

[Lemma 1.22 of the typed notes.]

READ IT!

Theorem If G, G' are variable-based and $V \cap V' = \emptyset$, then

(a) if H is the union grammar of G, G', then $\mathcal{L}(H) = \mathcal{L}(G) \cup \mathcal{L}(G')$.

(b) if H is the concatenation gr. of G, G', then $\mathcal{L}(H) = \mathcal{L}(G)\mathcal{L}(G')$

Observe that with Remark and the fact that we can make sets of variables disjoint, we find for each G, G' some H producing union or concatenation.

Thus (using (3) from page 8):

The class of type 1 and type 2 languages is closed under union and concatenation.