Automata & Formal Languages

Third Lecture

11 October 2022

RECAP

Rewrite systems $R = (\Omega, P)$

For a fixed Ω, only countably many rewrite systems.

Grammars $G = (\Sigma, V, P, S)$ $\quad \mathbb{W} = \Sigma^*$ $\quad \mathbb{W}^+ = \mathbb{W} \setminus \{\varepsilon\}$ $\quad \boxed{\mathcal{L}(G)}$

Eight different grammars $G_0, \ldots, G_7$ all producing the same language

$$
\begin{array}{ll}
G_0 := (\{a\}, \{S\}, P_0, S) & P_0 := \{S \to aaS, S \to a\} \\
G_1 := (\{a\}, \{S\}, P_1, S) & P_1 := \{S \to aSa, S \to a\} \\
G_2 := (\{a\}, \{S\}, P_2, S) & P_2 := \{S \to Saa, S \to a\} \\
G_3 := (\{a\}, \{S\}, P_3, S) & P_3 := \{S \to aaS, S \to aaSaa, S \to a\} \\
G_4 := (\{a\}, \{S\}, P_4, S) & P_4 := \{S \to aaS, S \to Saa, S \to aSa, S \to a\} \\
G_5 := (\{a\}, \{S\}, P_5, S) & P_5 := \{S \to aaS, aSa \to aaa, S \to a\} \\
G_6 := (\{a\}, \{S\}, P_6, S) & P_6 := \{S \to aaS, aaS \to aSa, S \to a\} \\
G_7 := (\{a\}, \{S\}, P_7, S) & P_7 := \{S \to aaS, aaS \to a, S \to a\}
\end{array}
$$

$\mathcal{L}(G_0) = \{a^{2n+1} ; n \in \mathbb{N}\}$

<u>Def.</u> G and G' are equivalent if

$$\mathcal{L}(G) = \mathcal{L}(G').$$

<u>GOAL</u>: For a fixed Σ, there are only countably many languages of the form $\mathcal{L}(G)$ for a grammar G.

<u>Def.</u> Let $G = (\Sigma, V, P, S)$ and
$G' = (\Sigma, V', P', S')$ be grammars.
A function $f: \underset{\Sigma \cup V}{\underset{\shortparallel}{\Omega}} \longrightarrow \underset{\Sigma \cup V'}{\underset{\shortparallel}{\Omega'}}$ is called

<u>isomorphism</u> if

(i) $f \upharpoonright \Sigma$ is the identity

(ii) $f(S) = S'$

(iii) $f \upharpoonright V$ is a bijection from V to V'

(iv) $\alpha \longrightarrow \beta \in P \iff$
$f(\alpha) \longrightarrow f(\beta) \in P'$

Reminder: that is the extension of f to Ω^*.

[In Lecture I, we wrote $\hat{f}$ for this extension and remarked that we are usually using the symbol f for the extension as well.]

Proposition 1.11.

Isomorphic grammars are equivalent.

Proof. If f is iso from G to G', then f^{-1} is iso from G' to G.

Thus, by symmetry, it's enough to show "if f is iso from G to G', then
$$\mathcal{L}(G) \subseteq \mathcal{L}(G')".$$

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

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

APPLY f to the derivation

$$S' \underset{(ii)}{=} f(S) = f(\sigma_0) \xrightarrow{G'}_1 f(\sigma_1) \xrightarrow{G'}_1 \cdots \xrightarrow{G'}_1 f(\sigma_n)$$

$$f(\sigma_n) = f(w) = w \text{ by (i)}$$

the f-image of $(\sigma_0, \dots, \sigma_n)$ is a G'-derivation.

Because of (iv)

$$\Longrightarrow \quad w \in \mathcal{D}(G', S') \Rightarrow w \in \mathcal{L}(G').$$

q.e.d.

Proposition 1.12 If $G = (\Sigma, V, P, S)$ and V' is s.t. $|V| = |V'|$, then there are P', S' s.t.

$$\mathcal{L}(G) = \mathcal{L}(G')$$

with $G' = (\Sigma, V', P', S')$.

Proof. If $f : V \longrightarrow V'$ is a bijection, extend it to Ω by letting $f(a) = a$ for all $a \in \Sigma$.

So this satisfies (i) & (iii) of iso def'n.

Define $S' := f(S)$ [so (ii) is satisfied]

Define $P' := \{ f(\alpha) \to f(\beta) ; \alpha \to \beta \in P \}$

Then $\overset{G'}{\overset{\shortparallel}{(\Sigma, V', P', S')}}$ is isomorphic to G and thus by P 1.11,

$$\mathcal{L}(G) = \mathcal{L}(G').$$

q.e.d.

Proposition 1.13. Up to equivalence, there are only countably many languages of the form $\mathcal{L}(G)$ for a grammar G with fixed Σ.

Proof. Write $\mathcal{L}$ for the set of all such languages.

If you fix V, then there are only countably rewrite systems with Σ, V fixed.

So $\mathcal{G}_V$, the set of all grammars with fixed V, is a finite union of countable sets, so countable.

Thus $\mathcal{L}_V := \{\mathcal{L}(G) ; G \in \mathcal{G}_V\}$ is also countable.

By P 1.12, we can define $\mathcal{L}_n := \mathcal{L}_V$ for some (every) set V with $|V| = n$.

But $\mathcal{L} = \bigcup_{n>0} \mathcal{L}_n$, so it's a countable union of countable sets, thus countable. q.e.d.

Consequence:

[uncountably many]

All languages: $\mathcal{P}(W)$

"grammatical languages".

i.e., languages L s.t. there is a grammar G with

$$L = \mathcal{L}(G)$$

[countably many]

§ 1.5 The Chomsky Hierarchy

Properties of production rules

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

- **noncontracting** if $|\alpha| \leq |\beta|$.

$\Uparrow$

- **context-sensitive**

$$\exists A \in V \;\exists \gamma, \delta \in \Omega^* \;\exists \eta \in \Omega^+ \quad (\Omega^+ = \Omega^* \setminus \{\varepsilon\})$$

$$\alpha = \gamma A \delta \quad \beta = \gamma \eta \delta.$$

$\Uparrow$

- **context-free** $\alpha = A \in V$. $|\beta| \geq 1$.

$\Uparrow$

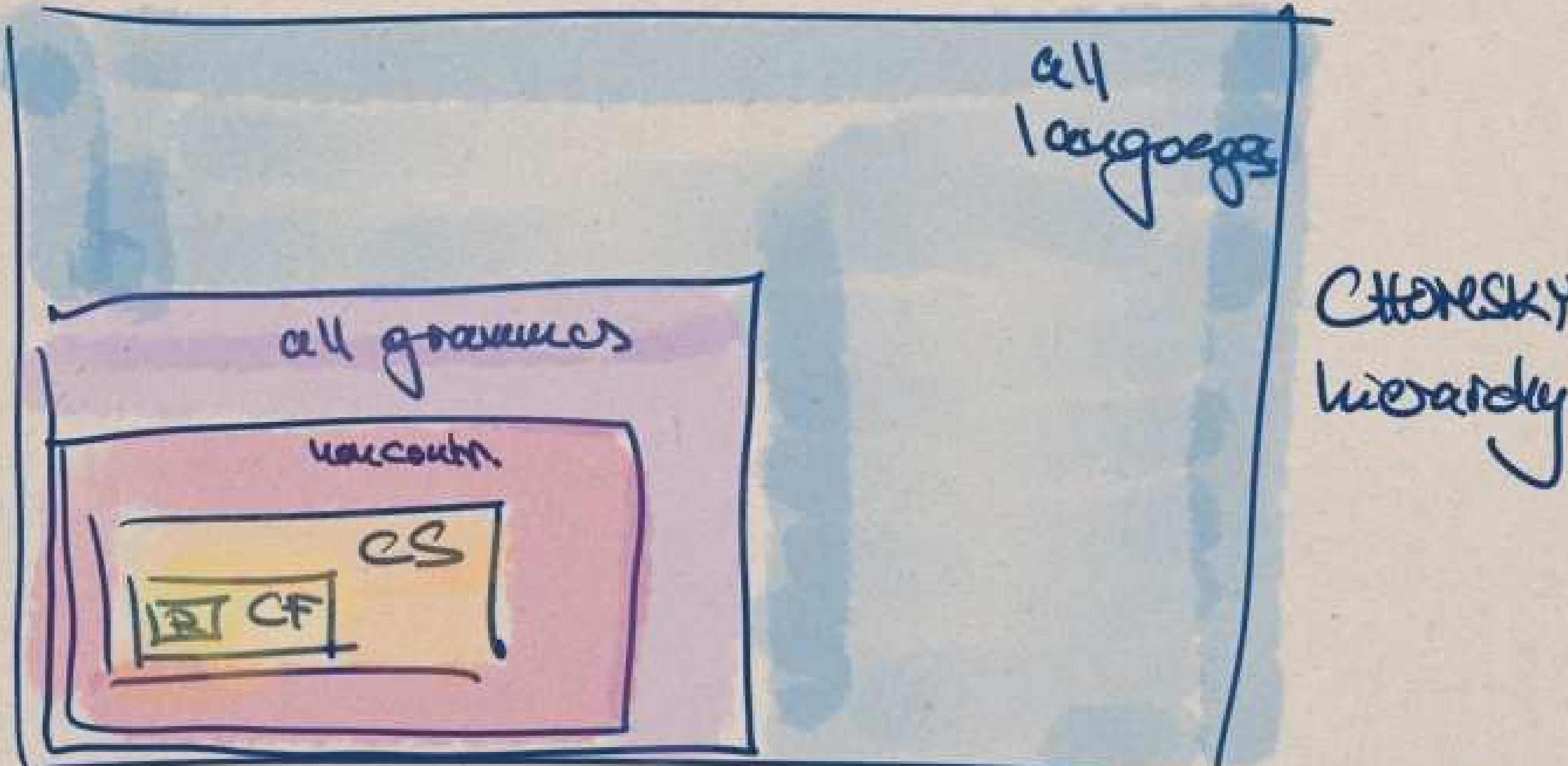
- **regular** if $\alpha = A \in V$ and β is either $a \in \Sigma$, $aB \in \Sigma V$

Let $\mathbb{Q}$ be any of the properties
"noncontracting", "context-s.", "context-f",
"regular".

<u>Then</u>:

A grammar G is $\mathbb{Q}$ iff all of its production rules are $\mathbb{Q}$.

A language L is $\mathbb{Q}$ if there is a grammar G s.t.
G is $\mathbb{Q}$ and $L = \mathcal{L}(G)$.



<u>Q</u>: Is this hierarchy PROPER?

Chomsky's terminology

L is **type 0** if it is of the form $\mathcal{L}(G)$ for some G.

type 1 $\mathcal{L}(G)$ for G context-sensitive

type 2 $\mathcal{L}(G)$ for G context-free

type 3 $\mathcal{L}(G)$ for G regular

What happened to "non-contracting"?

Thm (Chomsky). L is noncontracting iff L is context-sensitive

$\longrightarrow$ ES#1.

Note: The theorem does **NOT** say: G is noncontracting iff G is context-sensitive!

All of these eight grammars are equivalent, but they are in different Chomsky types.

all context-free but not regular

$G_0 := (\{a\}, \{S\}, P_0, S) \quad P_0 := \{S \to aaS, S \to a\}$

$G_1 := (\{a\}, \{S\}, P_1, S) \quad P_1 := \{S \to aSa, S \to a\}$

$G_2 := (\{a\}, \{S\}, P_2, S) \quad P_2 := \{S \to Saa, S \to a\}$

$G_3 := (\{a\}, \{S\}, P_3, S) \quad P_3 := \{S \to aaS, S \to aaSaa, S \to a\}$

$G_4 := (\{a\}, \{S\}, P_4, S) \quad P_4 := \{S \to aaS, S \to Saa, S \to aSa, S \to a\}$

Type 2

$G_5 := (\{a\}, \{S\}, P_5, S) \quad P_5 := \{S \to aaS, aSa \to aaa, S \to a\}$ Type 1

$G_6 := (\{a\}, \{S\}, P_6, S) \quad P_6 := \{S \to aaS, aaS \to aSa, S \to a\}$

$G_7 := (\{a\}, \{S\}, P_7, S) \quad P_7 := \{S \to aaS, aaS \to a, S \to a\}$ Type 0

not even noncontracting

not context-sensitive, but noncontracting

not context-free but context-sensitive

<u>To make things worse</u>

$$P_8 = \{S \to aA, S \to a, A \to aS\}$$

then G_8 is regular and $\mathcal{L}(G_8) = \{a^{2n+1} ; n \in \mathbb{N}\}$

$\Longrightarrow$ ES#1.

One of things motivating our work will be the development of techniques that allows us to separate the Chomsky classes

---

§ 1.6 Decision problems

Three important decision problems:

	INPUT	QUESTION
① WORD PROBLEM	G grammar, w word	$w \in \mathcal{L}(G)$?
② EMPTINESS PROBLEM	G grammar	$\mathcal{L}(G) = \emptyset$?
③ EQUIVALENCE PROBLEM	G, G' grammars	$\mathcal{L}(G) = \mathcal{L}(G')$.

We call a problem SOLVABLE if there is an algorithm that gives the correct answer.

Otherwise: UNSOLVABLE.

All general decision problem will turn out to be

UNSOLVABLE.

So we look at restricted problems:

e.g.,
the word problem for type i grammars.

Next time:

We'll solve the word problem for type 1, 2, 3 grammars.