Automata & Formal Languages II

SECOND LECTURE

Thursday 17 October 2024

RECAP: Finite set of symbols Ω

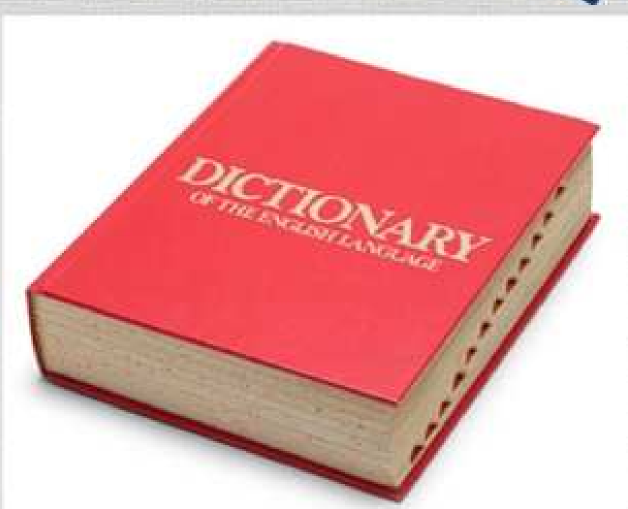
Ω^* : set of Ω-strings $\longrightarrow$ COUNTABLE

ε : empty string

Ω^+ : set of non-empty Ω-strings

Strings $\rightsquigarrow$ "words".

Then language is a set of words.



Real-world languages are finite.

Noam Chomsky

Chomsky in 2017

Born	Avram Noam Chomsky December 7, 1928 (age ~~94~~) 95 Philadelphia, Pennsylvania, U.S.
Spouses	Carol Schatz (m. 1949; died 2008) Valeria Wasserman (m. 2014)
Children	3, including Aviva
Parent	William Chomsky (father)

RECURSION

[GENERATIVE GRAMMARS]

COLOURLESS GREEN IDEAS SLEEP FURIOUSLY

FURIOUSLY SLEEP IDEAS GREEN COLOURLESS

SEMANTICS
- meaningless
- meaningful
  - true
  - false

SYNTAX
- wellformed
- illformed

EXAMPLE

$$\Omega = \{\underset{\text{subject}}{S}, \underset{\text{verb}}{V}, \underset{\text{that}}{T}, \underset{\text{clause}}{C}\} \cup \text{other stuff}$$

Rewrite rules:

$C \longrightarrow SVTC$

$S \longrightarrow$ John

$S \longrightarrow$ Jill

$S \longrightarrow$ Ann

$V \longrightarrow$ knows

$V \longrightarrow$ claims

$V \longrightarrow$ thinks

$T \longrightarrow$ that

$C \longrightarrow$ John likes fish

$$C \longrightarrow SVT\underline{C} \longrightarrow SVT\underline{SVTC} \longrightarrow SVTSVTSVTC$$

S	V	T	S	V	T	S	V	T	C
John	knows	that	Jill	claims	that	Ann	thinks	that	John likes fish.

Example 1.10. Let $\Sigma = \{\mathbf{0}\}$, $V = \{S\}$, $P_0 := \{S \to \mathbf{00}S, S \to \mathbf{0}\}$, and $G_0 := (\Sigma, V, P_0, S)$. Then $\mathcal{L}(G_0)$ is the set of all odd-length words consisting of the letter $\mathbf{0}$.

[Let's prove this in detail: first of all, we notice that each rewrite step either keeps the number of symbols the same or increases it by two. We prove by induction on the length of the derivation, that every string produced by G from S has odd length: the unique string with a derivation of length zero is S which has odd length; of all strings produced by derivations of length n have odd length, say, length $2k+1$, then a string with a derivation of length $n+1$ has either length $2k+1$ or $(2k+1)+2$, thus odd length.

In order to see that the unique word $\mathbf{0}^{2n+1}$ of length $2n+1$ can be produced, we give provide a concrete derivation: we apply the production rule $S \to \mathbf{00}S$ to the start symbol n times to obtain $\mathbf{0}^{2n}S$ and finally apply $S \to \mathbf{0}$ to remove the nonterminal and acquire the desired word $\mathbf{0}^{2n+1}$.]

$$
\begin{aligned}
&P_1 := \{S \to \mathbf{0}S\mathbf{0}, S \to \mathbf{0}\},\\
&P_2 := \{S \to S\mathbf{00}, S \to \mathbf{0}\},\\
&P_3 := \{S \to \mathbf{00}S, S \to \mathbf{00}S\mathbf{00}, S \to \mathbf{0}\},\\
&P_4 := \{S \to \mathbf{00}S, S \to S\mathbf{00}, S \to \mathbf{0}S\mathbf{0}, S \to \mathbf{0}\},\\
&P_5 := \{S \to \mathbf{00}S, \mathbf{0}S\mathbf{0} \to \mathbf{000}, S \to \mathbf{0}\},\\
&P_6 := \{S \to \mathbf{00}S, \mathbf{00}S \to \mathbf{0}S\mathbf{0}, S \to \mathbf{0}\}, \text{ or}\\
&P_7 := \{S \to \mathbf{00}S, \mathbf{00}S \to \mathbf{0}, S \to \mathbf{0}\}, \text{ etc.}
\end{aligned}
$$

Definition 1.11. Let Σ be an alphabet and let $G = (\Sigma, V, P, S)$ and $G' = (\Sigma, V', P', S')$ be two grammars over Σ. Let $f : \Omega \to \Omega'$ be any function and extend it by recursion to Ω^*. We say that f is an *isomorphism between G and G'* if

(i) it is the identity on Σ, i.e., $f(a) = a$ for all $a \in \Sigma$;

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

(iii) the restriction $f{\restriction}V$ is a bijection between V and V'; and

(iv) for each $\alpha, \beta \in \Omega^*$, we have $\alpha \to \beta \in P$ if and only if $f(\alpha) \to f(\beta) \in P'$.

If there is an isomorphism between G and G', we also say that the two grammars are *isomorphic*.

§1.4 Grammars

Assume $\Omega = \Sigma \cup V$, $\Sigma \cap V = \emptyset$, $\Sigma \neq \emptyset \neq V$

ALPHABET / LETTERS / terminal symbol — VARIABLES / non-terminal symbols

$\mathbb{W} := \Sigma^*$ set of words

$L \subseteq \mathbb{W}$ (FORMAL) LANGUAGE

Def 1.7 $G = (\Sigma, V, S, P)$ grammar

if $(\Sigma \cup V, P)$ RWS

$S \in V$ called start symbol

From §1.2, $\mathcal{D}(G, \alpha)$ is the set of derivable strings.

$$\mathcal{L}(G) := \mathcal{D}(G, S) \cap \mathbb{W}$$ Language generated by G.

Special case.

$\Sigma_{01} = \{0, 1\}$ binary alphabet

$\mathbb{B} := (\Sigma_{01})^*$ binary words

§1.2 Rewrite Systems

Fix Ω finite set of symbols.

$\alpha \in \Omega^+$, $\beta \in \Omega^*$ Then $\alpha \longrightarrow \beta$ is called a rewrite rule / production rule

interpreted as "if σ is a string containing α, I can replace it by β"

So, the set of rewrite rules is $\Omega^+ \times \Omega^*$.

[By §1.1, this is countable.]

Def 1.5 (Ω, P) rewrite system (RWS) if P is a finite set of rewrite rules.

Prop 1.6 If Ω fixed, there are ctbly many RWS on Ω.

[By §1.1 & the above.]

EXAMPLES

(1) No production rule contains S as LHS $\Rightarrow \mathcal{D}(G,S) = \{S\}$

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

(2) If every prod. rule contains a variable

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

[Both of these are (easy) induction proofs on the length of derivations]

(3) Example 1.10. $\mathcal{L}(G_0) = \{0^{2n+1} ; n \in \mathbb{N}\}$

"$\subseteq$". Only need to show that every elt. of $\mathcal{D}(G_0,S)$ has odd length.

Proof. Induction, since rules preserve parity.

All rules preserve parity, so

"$\supseteq$". Fix n.

$S \xrightarrow{G_0}_1 00S \xrightarrow{G_0}_1 \cdots \xrightarrow{G_0}_1 0^{2n}S \xrightarrow{G_0}_1 0^{2n}0 = 0^{2n+1}$.

G_1–G_7 give the same language.

G, G' are equivalent if $\mathcal{L}(G) = \mathcal{L}(G')$.

[Blackboard partially erased.]

Interpretation $\sigma, \tau \in \Omega^*$, $R = (\Omega, P)$ is RWS

$\sigma \xrightarrow{R}_1 \tau$ "σ can be rewritten to τ by R in one step"

$:\Longleftrightarrow \exists \alpha, \beta, \gamma, \delta \in \Omega^*$

$\sigma = \alpha\beta\gamma$, $\tau = \alpha\delta\gamma$, $\beta \to \delta \in P$.

Then $\xrightarrow{R}$ is the reflexive, transitive closure of $\xrightarrow{R}_1$:

$\sigma \xrightarrow{R} \tau \Longleftrightarrow \sigma = \tau$ or

$\exists \sigma_0, \ldots, \sigma_n$ s.t. $\sigma_0 = \sigma$, $\sigma_n = \tau$ &

$\forall k < n \quad \sigma_k \xrightarrow{R}_1 \sigma_{k+1}$.

called R-derivation

"σ R-produces τ"

R-generates

R-derives

$\mathcal{D}(R, \alpha) := \{\beta ; \alpha \xrightarrow{R} \beta\}$

Prop 1.12 Isomorphic grammars are equivalent.

Pf. If G, G' are iso by f, by symmetry only NTS $\mathcal{L}(G') \subseteq \mathcal{L}(G)$

$$S \xrightarrow{G} w$$

$$\begin{tikzcd}
S \arrow[r, "G"', "1"] \arrow[d, squiggly] & \sigma_1 \arrow[r, "G", "1"'] \arrow[d, squiggly] & \sigma_2 \arrow[r, "G", "1"'] \arrow[d, squiggly] & \cdots \arrow[r, "G", "1"'] & \sigma_n \arrow[r, "G", "1"'] \arrow[d, squiggly] & w \arrow[d, squiggly] \\
f(S) = S' \arrow[r, "G'", "1"'] & f(\sigma_1) \arrow[r, "G'", "1"'] & f(\sigma_2) \arrow[r, "G'", "1"'] & \cdots \arrow[r, "G'", "1"'] & f(\sigma_n) \arrow[r, "G'", "1"'] & f(w) = w
\end{tikzcd}$$

q.e.d.

$\mathcal{L}(\Sigma, V) := \{\mathcal{L}(G) ; G \in \mathcal{G}(\Sigma, V)\}$

Define $\mathcal{G}(\Sigma, V)$ to be the set of all grammars with Σ, V.

[If $\mathcal{R}$ is the set of RWS over $\Sigma \cup V$ (ctble), then there is a surj.

$$V \times \mathcal{R} \longrightarrow \mathcal{G}(\Sigma, V)]$$

Corollary 1.15 If $|V| = |V'|$, then $\mathcal{L}(\Sigma, V) = \mathcal{L}(\Sigma, V')$.

[Extend bij $f : V \longrightarrow V'$ in the natural way to an iso between $G \in \mathcal{G}(\Sigma, V)$ and an appropriately def'd $G' \in \mathcal{G}(\Sigma, V')$]

$\mathcal{L}(\Sigma) := \{\mathcal{L}(G) ; \exists V \; G \in \mathcal{G}(\Sigma, V)\}$

Claim $\mathcal{L}(\Sigma)$ is countable.

Pf. $\mathcal{L}(\Sigma) = \bigcup_{n \in \mathbb{N}} \mathcal{L}_n$; $\mathcal{L}_n$ the set of languages produced by grammars w/ n variables

But $\mathcal{L}_n$ is ctble (by C 1.15), so $\mathcal{L}(\Sigma)$ is a ctble union of ctble sets, so ctble (by N&S).

qed