Automata & Formal Languages

Fourth Lecture

13 October 2022

RECAP

Chomsky hierarchy:

NONCONTRACTING / CONTEXT-SENSITIVE	Type 1
CONTEXT-FREE	Type 2
REGULAR	Type 3

Decision problems:

- Word problem: Is $w \in \mathcal{L}(G)$?
- Emptiness problem: Is $\mathcal{L}(G) \neq \emptyset$?
- Equivalence problem: Is $\mathcal{L}(G) = \mathcal{L}(G')$?

Stated w/o proof:
Neither of the three problems is solvable in general!

Goal: SHOW THAT THE WORD PROBLEM FOR NONCONTRACTING GRAMMARS IS SOLVABLE!

Lemma 1.17 If G is noncontracting, $w \in \mathbb{W}$, then there is a bound N depending only on $|w|$ and $|\Omega|$ s.t.

$$w \in \mathcal{L}(G) \iff w \text{ has a } G\text{-derivation of length} \leq N.$$

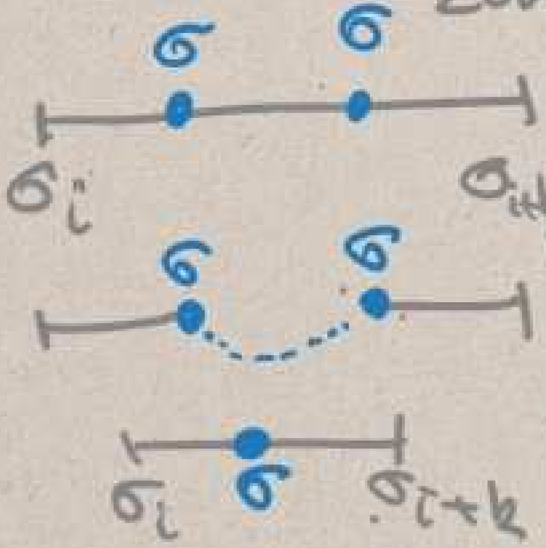
Proof. Let's have any G-derivation of w:

$$S = \sigma_0 \xrightarrow[1]{G} \sigma_1 \xrightarrow[1]{G} \cdots \xrightarrow[1]{G} \sigma_n = w$$

$1 \longrightarrow |w|$ LENGTHS

Because G is noncontracting, this is a nondecreasing seq. of nat. numbers.

Zoom and consider a stretch of the derivation whose $|\sigma_i|$ does not change



Assume that $|\sigma_i| = |\sigma_{i+k}|$.

If there are two places with the same string σ, we can reduce the derivation.

Let now $(\sigma_0, \dots, \sigma_k)$ be a G-derivation of w of minimal length.

Then we know by the pigeon hole principle that

$$k \leq \sum_{l=1}^{|w|} |\Omega|^l =: N$$

q.e.d.

<u>COROLLARY</u> The word problem for noncontracting grammars is solvable.
($\Rightarrow$ c.-s., c.-f., reg.)

<u>Proof.</u> Check systematically every single sequence of strings up to length N from L 1.17 and check whether it is a derivation $S \xrightarrow{G} w$.
If so, answer "yes".
If not, answer "no". q.e.d.

§1.7 Closure properties

L, M languages	Operations on languages	If $\mathcal{C}$ is a class of languages [e.g., "regular", "context-free", ...]
Concatenation	LM	$L, M \in \mathcal{C} \Rightarrow LM \in \mathcal{C}$
Union	$L \cup M$	$L, M \in \mathcal{C} \Rightarrow L \cup M \in \mathcal{C}$
Intersection	$L \cap M$	$L, M \in \mathcal{C} \Rightarrow L \cap M \in \mathcal{C}$
Complementation	$W^+ \setminus L$	$L \in \mathcal{C} \Rightarrow W^+ \setminus L \in \mathcal{C}$
Difference	$L \setminus M$	$L, M \in \mathcal{C} \Rightarrow L \setminus M \in \mathcal{C}$

$\mathcal{C}$ is closed under...

We are interested in:

WHICH CLASSES ARE CLOSED UNDER WHICH OPERATIONS?

[Note: Some closure properties imply others, e.g., by

$$\overline{\overline{A} \cup \overline{B}} = A \cap B$$

[De Morgan]

Let $G = (\Sigma, V, P, S)$
$G' = (\Sigma, V', P', S')$ be grammars.

Then $H = (\Sigma, V \cup V' \cup \{T\}, P^*, T)$ is called the CONCATENATION GRAMMAR

where T is a new variable $T \notin V \cup V'$

$$P^* := P \cup P' \cup \{T \longrightarrow SS'\}$$

and $H' = (\Sigma, V \cup V' \cup \{T\}, P^{**}, T)$ is called the UNION GRAMMAR

where T is a new variable

$$P^{**} = P \cup P' \cup \{T \longrightarrow S, T \longrightarrow S'\}$$

Remark 1 Obviously

$$\mathcal{L}(G)\mathcal{L}(G') \subseteq \mathcal{L}(H)$$
$$\mathcal{L}(G) \cup \mathcal{L}(G') \subseteq \mathcal{L}(H')$$

Remark 2 In general, $\supseteq$ does not hold (ES#1) since there could be undesired interaction between P, P'.

We can assume by P 1.12 that $V \cap V' = \emptyset$. But that's not enough.

<u>Remark 3</u> $T \to S$, $T \to S'$ are regular
$T \to SS'$ is context-free

So if G, G' are c-f/c-s, so are H, H'

If G, G' are regular, so is H'.

$\Longrightarrow$
if $\mathcal{L}(H) = \mathcal{L}(G)\mathcal{L}(G')$
and $\mathcal{L}(H') = \mathcal{L}(G) \cup \mathcal{L}(G')$

then reg., c-f, c-s. languages closed under union and c-f, c-s languages closed under concat.

Closure properties we proved

<u>Def.</u> A production rule is called variable-based if all symbols occ. on the LHS are variables.

A grammar is v.-b. if all rules all.

<u>Note</u> Regular & c-f grammars are by definition variable-based.

<u>Theorem</u> If G, G' are variable-based and do not share variables ($V \cap V' = \emptyset$), then
$$\mathcal{L}(H) = \mathcal{L}(G)\mathcal{L}(G')$$
$$\mathcal{L}(H') = \mathcal{L}(G) \cup \mathcal{L}(G').$$

Before proving this:

<u>Lemma 1.21</u> Every grammar is equivalent to a variable-based grammar.

[Trick: given $G = (\Sigma, V, P, S)$ take a new variable for each $a \in \Sigma$, call it X_a.

Write for $\alpha \in \Omega^*$, $X(\alpha)$ is the string where each letter has been replaced by the appropriate X-var.

$$G' = (\Sigma, V', P', S)$$
where
$$V' = V \cup \{X_a ; a \in \Sigma\}$$
$$P' = \{X(\alpha) \longrightarrow X(\beta) ; \alpha \longrightarrow \beta \in P\} \cup \{X_a \longrightarrow a ; a \in \Sigma\}$$

Then $\mathcal{L}(G') = \mathcal{L}(G)$ and
G' is variable-based.
And $P \longmapsto P'$ preserves context-freeness and context-sensitivity.]

This implies the mentioned closure properties on page 6.

---

§ 1.8 Remark on ε.

By induction, noncontracting grammars do not produce ε.
Thus we usually work with W^+ instead of W.

Why not fix this by allowing the rule
$$S \longrightarrow \varepsilon.$$
In general just adding $S \longrightarrow \varepsilon$ to a grammar might again have unintended consequence since S could show up in complex derivable strings: $\alpha S \beta \xrightarrow[S \to \varepsilon]{} \alpha\beta$.

A rule is S-safe if it does not contain S on the RHS.

A grammar is ε-adequate if all rules are S-safe.

Then if G is ε-adequate then

$$G' = (\Sigma, V, P \cup \{S \to \varepsilon\}, S)$$

has the property

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

Making grammars ε-adequate:

Take new start symbol T

$$(\Sigma, V \cup \{T\}, P \cup \{T \to S\}, T).$$