Automata & Formal Languages

Fourth Lecture

Tuesday 22 October 2024

Recap

Chomsky hierarchy

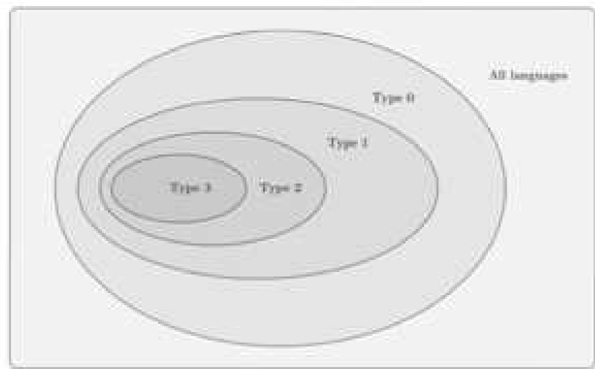

Figure 1: The Chomsky hierarchy

Closure properties

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

$\rightsquigarrow$ CONCATENATION GRAMMAR

$H = (\Sigma, V^*, T, P^*)$ with $V^* = V \cup V' \cup \{T\}$, $P^* = P \cup P' \cup \{T \to SS'\}$

$$\boxed{\text{Proposition 1.26} \quad \text{If } G, G' \text{ are variable-based and } V \cap V' = \emptyset, \text{ then } \mathcal{L}(H) = \mathcal{L}(G)\mathcal{L}(G').}$$

The assumptions are necessary (ES #1).

Remark 1 If G, G' are noncontracting / context-free, then so is H.

__Lemma 1 24__ Every grammar is equivalent to a variable-based grammar
[only variables on LHS]

__Proof__ Fix $G = (\Sigma, V, S, P)$.

(1) For each $a \in \Sigma$, add new variable X_a

$V' := V \cup \{X_a ; a \in \Sigma\}$

Let $X(\alpha)$ be the string in $(V')^*$ of α with all letters a replaced by X.

(2) Define $P' = \{X(\alpha) \to X(\beta) ; \alpha \to \beta \in P\}$

$G' = (\Sigma, V', S, P')$ This is clearly variable-based.

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

(3) We have $S \xrightarrow{G} \alpha$ iff $S \xrightarrow{G'} X(\alpha)$

(4) $P^+ := P' \cup \{X_a \to a ; a \in \Sigma\}$ __RECOVERY RULES__

$G^+ = (\Sigma, V', S, P^+)$ Note G^+ is variable-based

(5) __Claim__ $\mathcal{L}(G) = \mathcal{L}(G^+)$.

["$\subseteq$": $S \xrightarrow{G} w \overset{(2)}{\Rightarrow} S \xrightarrow{G'} X(w) \xrightarrow{G^+} w$ by only recovery rules

$S \xrightarrow{G^+} w$

"$\supseteq$": $S \xrightarrow{G^+} w$. Because G^+ is variable-based, we can move all recovery rules to the right of the derivation

$\underbrace{\boxed{S \xrightarrow{G'} X(w)} \xrightarrow{G^+} w}_{\text{only recovery rules}}$

$\Downarrow (3)$

$S \xrightarrow{G} w$]

q.e.d.

__Remark 2__ If G is noncontracting or context-free, then G^+ is the same.

__Corollary__. Type 0, 1, and 2 languages are closed under concatenation. [If G, G' are in Type k, we can w.l.o.g. assume that $V \cap V' = \emptyset$ by Prop 1 14. Apply Lemma 1 24 to make them variable-based w/o destroying Type k. The concatenation grammar for those gives concatenation by P 1 26 and preserves Type k by Remark 1.]

UNION

$G = (\Sigma, V, S, P)$

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

Form $V^+ = V \cup V' \cup \{T\}$ $\quad T \notin V \cup V'$

$P^+ = P \cup P' \cup \{T \to S, T \to S'\}$

$G^+ = (\Sigma, V^+, T, P^+)$ is the *union grammar*

Remark 3 The union grammar preserves Types 0, 1, 2, but not Type 3.

Prop 1.27 If G, G' are variable-based and $V \cap V' = \emptyset$, then

$$\mathcal{L}(G^+) = \mathcal{L}(G) \cup \mathcal{L}(G')$$

Proof "$\supseteq$" $\quad S \xrightarrow{G} w \implies T \xrightarrow{G^+}_1 S \xrightarrow{G} w$ $\quad$ Exactly the same for G'.

"$\subseteq$" $\quad T \xrightarrow{G^+} w$, then either $T \xrightarrow{G^+}_1 \underbrace{S \xrightarrow{G^+} w}_{\text{all rules in } P}$ or $T \xrightarrow{G^+}_1 \underbrace{S' \xrightarrow{G^+} w}_{\text{all rules in } P'}$

$\implies S \xrightarrow{G} w$ $\qquad \implies S' \xrightarrow{G'} w.$

q.e.d.

Corollary Type 0, 1, and 2 languages are closed under unions. [Proof precisely as before using P 1.27 instead of 1.26.]

Remaining open question: WHAT ABOUT TYPE 3?

§ 1.8 The empty word

If G is noncontracting, $w \in \mathcal{L}(G)$, i.e., $S \xrightarrow{G} w$, then $|w| \geq 1$.

So $\varepsilon \notin \mathcal{L}(G)$.

Remember $\{0^{2n+1}\,;\, n \in \mathbb{N}\}$ is Type 1. $\quad$ Easy to check.

But $\{0^{2n}\,;\, n \in \mathbb{N}\}$ is not since it includes ε. $\quad \{0^{2n}\,;\, n \geq 1\}$ is noncontracting.

How do we get ε without giving up the basic idea of noncontractingness?

Idea Just allow the rule $S \longrightarrow \varepsilon$ (basic ε-production)

Problem This could mess everything up if S occurs in some $\alpha \in \mathcal{D}(G, S)$.

Solution: A rule is called **S-safe** if S does not occur on RHS.

A grammar $G=(\Sigma,V,S,P)$ is called ε-adequate if all rules are S-safe.

Prop 1.29 Every grammar is eq. to an ε-adequate grammar.

Proof Fix $G=(\Sigma,V,S,P)$

Let $T\notin V$, define $V':=V\cup\{T\}$. Let $P':=P\cup\{T\to\alpha;\ S\to\alpha\in P\}$

$G'=(\Sigma,V',T,P')$. This is ε-adeq, since every rule is T-safe.

Claim $\mathcal{L}(G)=\mathcal{L}(G')$. [Straightforward.] HOMEWORK.

Now. If G is any ε-adequate grammar, define $G^\varepsilon=(\Sigma,V,S,P^\varepsilon)$ with $P^\varepsilon:=P\cup\{S\to\varepsilon\}$.

Then $\mathcal{L}(G^\varepsilon)=\mathcal{L}(G)\cup\{\varepsilon\}$. [Straightforward.]

We call G essentially regular, context-free or noncontracting if it's either R/C/N or G^ε for some R/C/N grammar.

Chapter 2 : REGULAR LANGUAGES

Regular grammar

	$A, B \in V$ & $a \in \Sigma$	# letters	# variable	length
TERMINAL RULES	$A \longrightarrow a$	+1	−1	0
NONTERMINAL RULES	$A \longrightarrow aB$	+1	0	+1

$$S \xrightarrow{G} w$$

1. There is precisely terminal rule: at the very end.
2. There are precisely $|w|-1$ nonterminal rules.