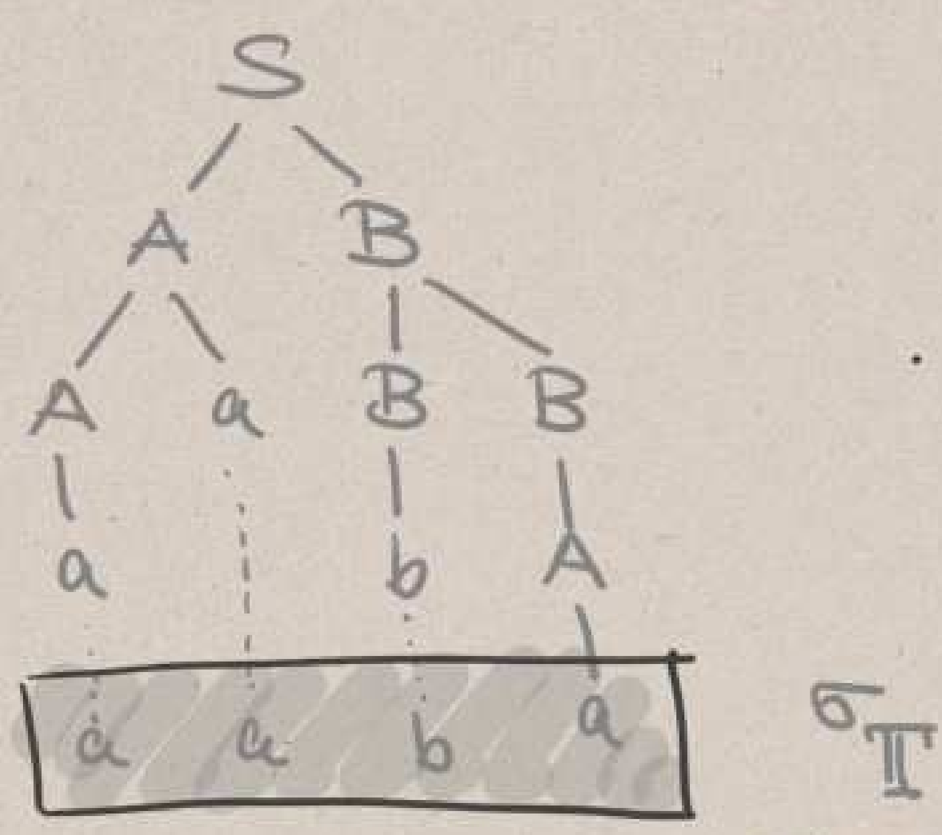
XII AUTOMATA & FORMAL LANGUAGES

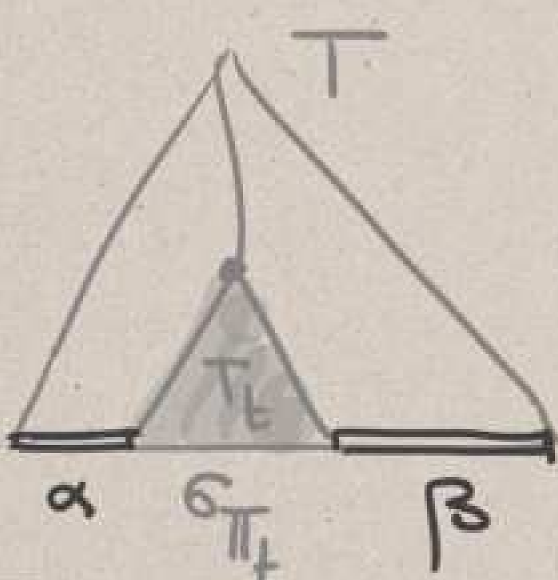
TWELFTH LECTURE

All Saints' Day
1 NOVEMBER 2022

PARSE TREES

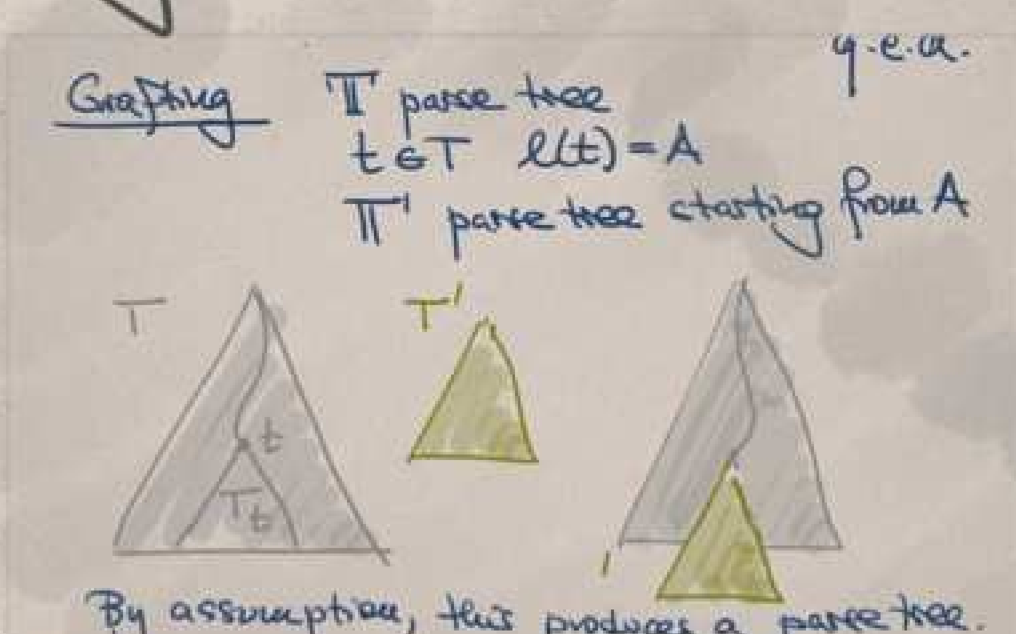
Recap





GRAFTING

FROM LECTURE XI



$$\sigma_{\Pi^*} = \alpha\, \sigma_{\Pi'}\, \beta$$

where $\sigma_{\Pi} = \alpha\, \sigma_{\Pi_t}\, \beta$ and Π' is grafted onto t to produce $\Pi^* = \mathrm{graft}(\Pi, t, \Pi')$

Chomsky Normal Form

$G = (\Sigma, V, P, S)$ is in **Chomsky Normal Form (CNF)** if all rules are of the form:

UNARY → $A \longrightarrow a$ or

BINARY → $A \longrightarrow BC$

If G is in CNF and $w \in \mathcal{L}(G)$, then every G-derivation of w has length $2|w|-1$.

Goal: Show that every context-free grammar is equivalent to one in CNF.

Problematic rules:

Type A: any rule $A \longrightarrow \alpha$ with $|\alpha| > 1$ that contains letters in α

Type B: rules $A \longrightarrow B$ unit rules

Type C: rules $A \longrightarrow \alpha$ where $\alpha \in V^n$ and $n > 2$.

Eliminating Type A rules

$A \longrightarrow \alpha$
with $|\alpha| \geq 2$
& α contains
a letter

Reminds us of "variable-based"!
Remember what we did there:

if $A \longrightarrow \alpha$ is a rule
add for each $a \in \Sigma$ a new variable X_a
and map $\alpha \longmapsto X(\alpha)$

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

α with all occ. of any a replace with the corresponding X_a.

Then as before

$$P' := \{A \longrightarrow X(\alpha) ; A \longrightarrow \alpha \in P\}$$
$$\cup \{X_a \longrightarrow a ; a \in \Sigma\}$$

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

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

Eliminating Type B rules (= unit rules)

$A \longrightarrow B$
where $A, B \in V$.

Def. A grammar is called unit closed if
for all $A \longrightarrow B \in P$ and
$B \longrightarrow \alpha \in P$,
we also have $A \longrightarrow \alpha \in P$.

Lemma For each G there is a unit closed G'
s.t. $\mathcal{L}(G) = \mathcal{L}(G')$.

Proof Just take the closure.
Note: At most $|V| \times |P|$
new rules are added.

Lemma If G is a unit closed contextfree grammar
then removing all unit rules from
P produces an equivalent
grammar G'.

Proof. Clearly, $\mathcal{L}(G') \subseteq \mathcal{L}(G)$, so
only need to show $\supseteq$.
We prove it by showing that no G-derivation of a word that uses a unit rule can be its shortest derivation.

Let $w \in \mathcal{L}(G)$ with $S \xrightarrow{G} w$ that uses a unit rule. Write

$$S \xrightarrow{G} \boxed{\alpha A\beta \xrightarrow[1]{G} \alpha B\beta} \xrightarrow{G} w .$$

where this is the last occurrence of a unit rule.

Since B is a variable, we find some step after the use of $A \to B$ that removes B.

So:

$$\underbrace{S \xrightarrow{G} \alpha A\beta}_{n} \underbrace{\xrightarrow[1]{G}}_{1} \underbrace{\alpha B\beta \xrightarrow{G} \gamma B\delta}_{m} \xrightarrow[1]{G} \overbrace{\gamma\xi\delta \xrightarrow{G} w}^{k}$$

$B \to \xi$ is the first rule after $A \to B$ that uses B

Total length: $n+m+k+2$.

Since $\alpha B\beta \xrightarrow{G} \gamma B\delta$ did not use any B-rule,

we also get $\alpha A\beta \xrightarrow{G} \gamma A\delta$ with the same derivation (length m) with B replaced by A.

since G is context-free

We have $A \rightarrow B \in P$

$B \rightarrow \xi \in P$

$\underset{\text{unit closure}}{\Longrightarrow} \quad A \rightarrow \xi \in P,$

and thus

$$\underbrace{S \xrightarrow{G} \alpha A \beta}_{n} \underbrace{\xrightarrow{G} \gamma}_{m} A \delta \xrightarrow[1]{G} \gamma \underbrace{\xi \delta \xrightarrow{G} w}_{k}$$

So: length $n+m+k+1$

$< n+m+k+2.$

q.e.d.

Eliminating Type C rules

$A \longrightarrow \alpha$
where $|\alpha| \geq 3$
and all symbols in
α are variables

Idea is Replace

$$A \longrightarrow A_1 \dots A_n \quad \text{by}$$

$$A \longrightarrow A_1 X_1$$
$$X_1 \longrightarrow A_2 X_2$$
$$\vdots$$
$$X_{n-2} \longrightarrow A_{n-1} A_n$$

Define for $A \longrightarrow \alpha = A_1 \dots A_n$ and
new variables $X_1, \dots, X_{n-2}$ and

$$P_{A \to \alpha} := \{ A \to A_1 X_1, \dots, X_{n-2} \to A_{n-1} A_n \}$$

Then $P' := P \setminus \{A \to \alpha\} \cup P_{A \to \alpha}$
produces the same language.

Theorem (Chomsky)

There is an algorithm that transforms any c-f grammar into a grammar G' in CNF s.t.

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

Proof. $G \longmapsto G_0$

Step 1 Remove problems of Type A.

Step 2 Form the unit closure of G_0, call it G_1.

Step 3 Remove unit rules from G_1, call it G_2.

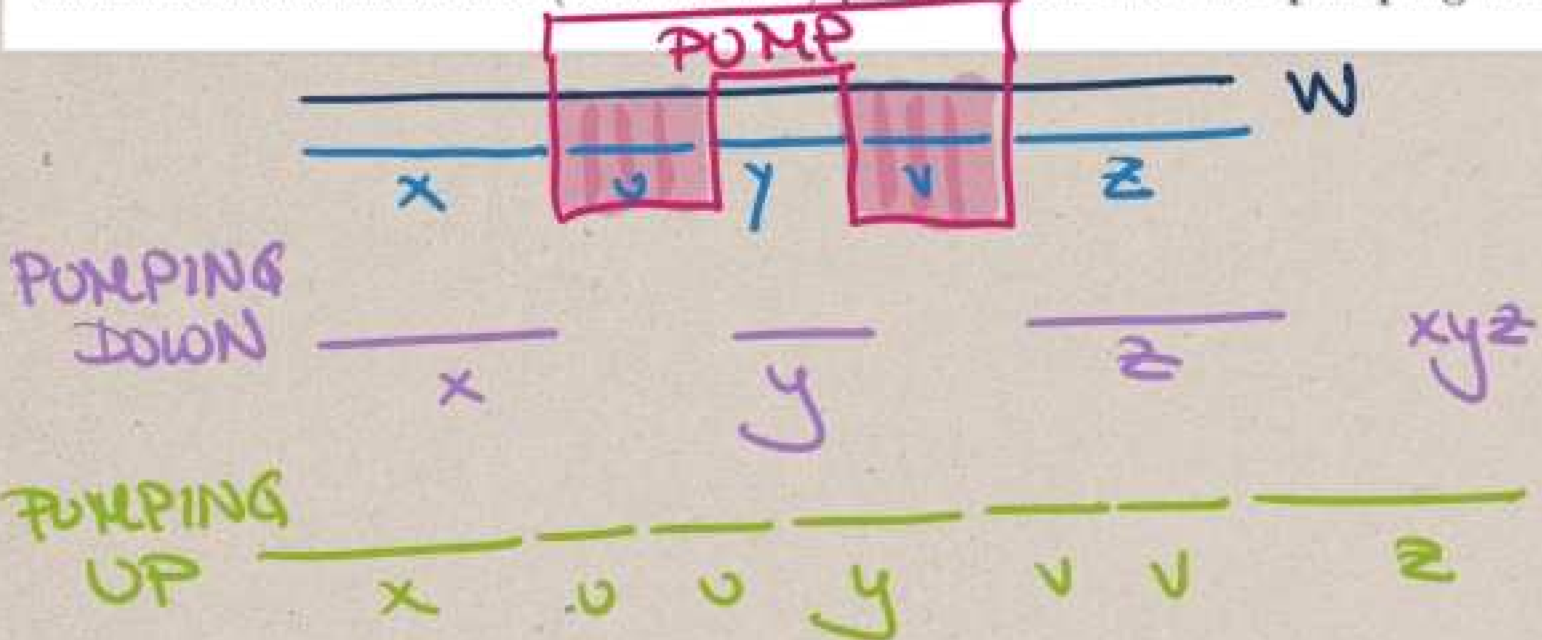
Step 4 Iteratively remove all problems of Type C.

This produces the right answer.

q.e.d.

§ 3.3 The context-free pumping lemma

Definition 3.9. Let $L \subseteq \mathbb{W}$ be a language. We say that L *satisfies the (context-free) pumping lemma with pumping number* n if for every word $w \in L$ such that $|w| \geq n$ there are words u, v, x, y, z such that $w = xuyvz$, $|uv| > 0$, $|uyv| \leq n$ and for all $k \in \mathbb{N}$, we have that $xu^k yv^k z \in L$. We say that L *satisfies the (context-free) pumping lemma* if there is some n such that it satisfies the (context-free) pumping lemma with pumping number n.



$xyz = xu^0yv^0z$

$xuuyvvz = xu^2yv^2z$

REMINDER Regular Pumping lemma

Definition 2.10. Let $L \subseteq \mathbb{W}$ be a language. We say that L *satisfies the (regular) pumping lemma with pumping number* n if for every word $w \in L$ such that $|w| \geq n$ there are words x, y, z such that $w = xyz$, $|y| > 0$, $|xy| \leq n$ and for all $k \in \mathbb{N}$, we have that $xy^k z \in L$. We say that L *satisfies the (regular) pumping lemma* if there is some n such that it satisfies the (regular) pumping lemma with pumping number n.

If a language L satisfies the pumping lemma and we have written $w = xyz$ as in the definition, then $xz = xy^0z$, xy^2z, xy^3z, etc. are all in L. We call the transition from $w = xyz$ to xz *pumping down* and the transition to xy^kz (for $k > 1$) *pumping up*.

Theorem 2.11 (The regular pumping lemma). For every regular language L, there is a number n such that L satisfies the regular pumping lemma with pumping number n.

CONTEXT-FREE

for all $w \in L$ $|w| \geqslant n$, there are x, y, z, u, v s.t.
$w = xuyvz$
$|uyv| \leq n$, $|uv| > 0$
& f.a. k $xu^kyv^kz \in L$.

for all $w \in L$ s.t. $|w| \geqslant n$, there are x, y, z s.t.
$w = xyz$, $|xy| \leq n$, $|y| > 0$
& for all k
$xy^kz \in L$

Observations

1. The pump now has two parts.
2. The bound $|vyv| \leq n$ does not give information about <u>WHERE</u> the pump is, since we have no bound on the size of x.
3. Regular PL $\Longrightarrow$ CF PL

If $w = rst$ with $\boxed{|rs| \leq n}$ $\boxed{|s| > 0}$,

let
$$
\begin{aligned}
x &:= \varepsilon \\
u &:= \varepsilon \\
y &:= r \\
v &:= s \\
z &:= t
\end{aligned}
$$

$$|uv| = |\varepsilon s| = |s| > 0$$

$$|uyv| = |\varepsilon r s| = |rs| \leq n$$

Therefore there are uncountably many languages satisfying CF PL, so it cannot characterize the class of c-f languages.

Theorem For every context-free language L there is an n s.t. L satisfies the context-free pumping lemma with pumping # n.

Proof in Lecture XIII.

Note In Example (5c) on ES#1, this language was shown to be type 1, so this proves that there are languages which are type 1, but not type 2.

Application of the theorem

The language $L := \{a^n b^n c^n ; n > 0\}$ is not context-free.

[Suppose it is, so by Thm, there is a pumping number N.

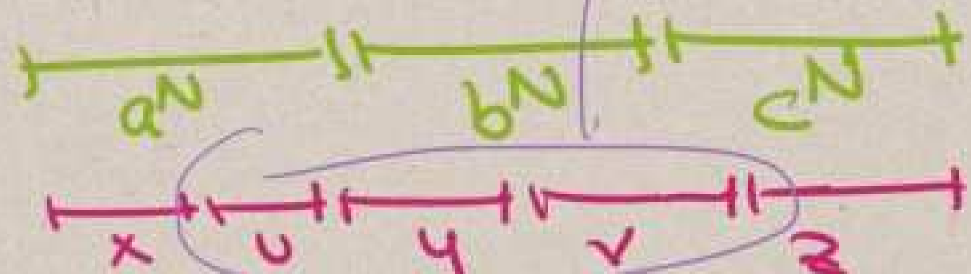
Choose $w = a^N b^N c^N \in L$.

$w = xuyvz$

This situation is impossible: uyv cannot contain as, bs, & cs!!

Since $|uyv| \leq N$, there are five cases:

???



1. entirely as,
2. entirely bs,
3. entirely cs,
4. as & bs.
5. bs & cs.

So, in each case, pumping down will create an imbalance.]