6 Simplification of CFGs; Normal Forms

[Linz ch. 6]

For the study of CFL's, we put their CFG's into **normal forms**, e.g.

- **Chomsky Normal Form (CNF)**, and
- **Greibach Normal Form (GNF).**

As a preliminary, we "simplify" the given CFG in 2 ways:

(1) Removing **$\boldsymbol{\lambda}$-productions**, and

(2) removing **unit productions**.

Definition: Given a CFG G:

(1) A **$\boldsymbol{\lambda}$-production** of G has the form $A \to \lambda$

(2) A **unit production** of G has the form $A \to B$

where A, B are non-terminals of G.

Theorem : (**$\boldsymbol{\lambda}$ and unit prod. elimination**)

For any CFG, G, there is a CFG, $\widehat{G}$, with no $\boldsymbol{\lambda}$- or unit prods, s.t.

$$L(\widehat{G}) = L(G) \backslash \{\lambda\}.$$

$\boldsymbol{Proof}$: Note: we cannot just **delete** such productions from G.
Think of the grammars for:

- $\{a^n b^n \mid n \geq 0\}$: $S \rightarrow \lambda \mid aSb$
- $WN_{[\,]}$: $S \rightarrow \lambda \mid [\,S\,] \mid SS$

Instead we **repeatedly** do the following:

(a) If $A \rightarrow \lambda$ and $B \rightarrow xAy$ are in G (and $xy \neq \lambda$),
add $B \rightarrow xy$ to G.

(b) (i) If $A \rightarrow B$ $(A \neq B)$ and $B \rightarrow x$ are in G,
add $A \rightarrow x$ to G.

(ii) If $A \rightarrow A$ is in G, just **remove** it!

We get a grammar G', where $L(G') = L(G)$.

Now remove all λ- and **unit** prods.
We get a grammar $\widehat{G}$ where $L(\widehat{G}) = L(G) \backslash \{\lambda\}$. $\square$

Examples:

(1) G is $S \to \lambda \mid aSb$, $\quad L(G) = \{a^n b^n \mid n \geq 0\}$.

Then G' has productions:

$$S \to \lambda$$

$$S \to aSb$$

$$S \to ab.$$

So $L(G') = L(G)$.

Then $\widehat{G}$ has productions:

$$\boxed{\begin{array}{l} S \to aSb \\ S \to ab \end{array}}$$

So

$$L(\widehat{G}) = L(G) \backslash \{\lambda\} = \{a^n b^n \mid n > 0\}.$$

(2) $\boldsymbol{G}$ is $\boldsymbol{WN}_{[\,]}$, with productions

$$S \to \lambda$$

$$S \to [\,S\,]$$

$$S \to SS$$

To get $\boldsymbol{G'}$, **add**

$$S \to [\,]$$

$$S \to S$$

(Added a unit production!)

To get $\widehat{G}$, **remove** $\boldsymbol{\lambda}$ and unit productions:

$$S \to [\,S\,]$$

$$S \to SS$$

$$S \to [\,]$$

i.e.

$$\boxed{S \to [\,S\,] \mid SS \mid [\,]}$$

So

$$L(\widehat{G}) \;=\; L(G)\backslash\{\lambda\}.$$

Note:

The language L and $L\backslash\{\lambda\}$ have "essentially" the same structure, since:

(a) If L is generated by a CFG G, then
$L\backslash\{\lambda\}$ is generated by $\widehat{G}$.

(b) If $L\backslash\{\lambda\}$ is generated by a CFG G, with productions P, then L is generated by G **plus** a **new start** terminal S_0
plus new prods $S_0 \to S \mid \lambda$.

Hence

$$L \text{ is a CFL} \iff L\backslash\{\lambda\} \text{ is a CFL}.$$

Q. Why do we want to eliminate **λ**- and **unit productions**?

A. With unit productions, there can be **loops** in derivations.

With **λ**-productions, long strings of non-terminals can be generated and then erased!

Without λ- or unit productions, every step in a derivation makes "**progress**" towards a terminal string.