Chapter 13

Grammatical Format

13.1 Regular Grammars

We previously saw that

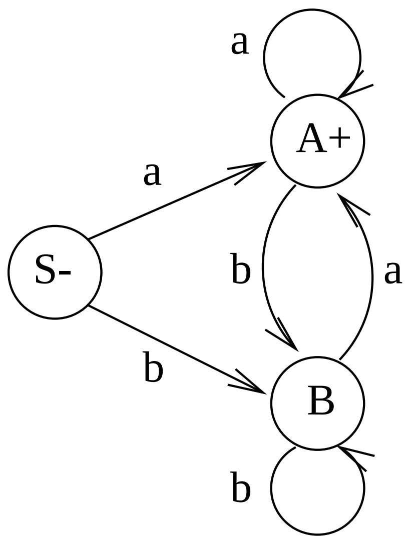
- CFG's can generate some regular languages.
- CFG's can generate some nonregular languages.

We will see that

- all regular languages can be generated by CFG's.
- some nonregular languages cannot be generated by CFG's.

Can turn FA into a CFG as follows:

Example: $L =$ all words ending in a.
FA:



Definition: The *path development* of a word processed on a machine:

- Start in starting state S.
- For each state visited, print out the input letters used thus far and the current state.

The word *ababba* has following path development on the FA:

$$\begin{array}{l} S \\ aA \\ abB \\ abaA \\ ababB \\ ababbB \\ ababbaA \\ ababba \end{array}$$

Now we define the following productions:

$$S \rightarrow aA \mid bB$$

$$\begin{aligned} A &\rightarrow aA \mid bB \mid \Lambda \\ B &\rightarrow aA \mid bB \end{aligned}$$

Note that:

- The CFG has a production

  $$X \rightarrow cY$$

  if and only if in the FA, there is an arc from state X to state Y labeled with c.

- The CFG has a production

  $$X \rightarrow \Lambda$$

  if and only if state X in the FA is a final state.

Derivation of *ababba* using the CFG:

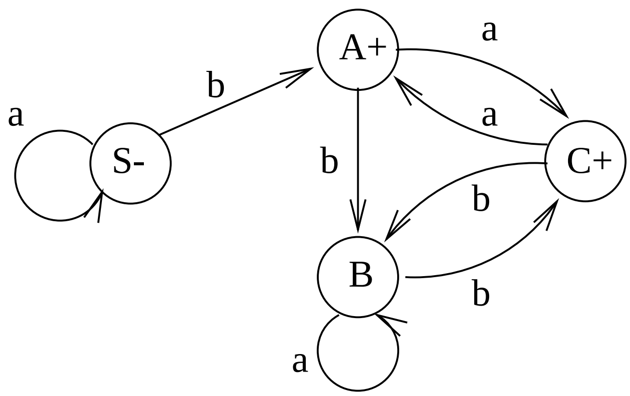
$$\begin{aligned} S &\Rightarrow aA \\ &\Rightarrow abB \\ &\Rightarrow abaA \\ &\Rightarrow ababB \\ &\Rightarrow ababbB \\ &\Rightarrow ababbaA \\ &\Rightarrow ababba \end{aligned}$$

There is a one-to-one correspondence between path developments on the FA and derivations in the CFG; i.e., we can use the pigeonhole principle.

The derivation of the word *ababba* using the CFG is exactly the same as the path development given above.

Theorem 21 *All regular languages are CFL's.*

Example:
FA:



productions:

$$\begin{aligned} S &\rightarrow aS \mid bA \\ A &\rightarrow aC \mid bB \mid \Lambda \\ B &\rightarrow aB \mid bC \\ C &\rightarrow aA \mid bB \mid \Lambda \end{aligned}$$

Consider a CFG $G = (\Sigma, \Omega, R, S)$, where

- Σ is the set of terminals
- Ω is the set of nonterminals, and $S \in \Omega$ is the starting nonterminal
- $R \subset \Omega \times (\Sigma + \Omega)^*$ is the set of productions, where a production $(N, \mathcal{U}) \in R$ with $N \in \Omega$ and $\mathcal{U} \in (\Sigma + \Omega)^*$ is written as

$$N \rightarrow \mathcal{U}$$

Definition: For a given CFG $G = (\Sigma, \Omega, R, S)$, W is a *semiword* if $W \in \Sigma^*\Omega$; i.e., W is a string of terminals (maybe none) cancatenated with exactly one nonterminal (on the right).

Example: $aabaN$ is a semiword if N is a nonterminal and a and b are terminals.

Definition: $G = (\Sigma, \Omega, R, S)$ is a *regular grammar* if $(N, \mathcal{U}) \in R$ implies $\mathcal{U} \in (\Sigma^*\Omega) + \Sigma^*$; i.e., each production has one of the following two forms:

1. nonterminal $\rightarrow$ semiword
2. nonterminal $\rightarrow$ word

where "word" $\in \Sigma^*$ is a string of terminals, possibly Λ.

Theorem 22 *If a CFG is a regular grammar, then the language generated by this CFG is regular.*

Proof.

- will prove theorem by showing that there is a TG that accepts the language generated by the CFG.
- Suppose CFG is as follows:

$$\begin{array}{rcl} N_1 & \rightarrow & w_1 M_1 \\ N_2 & \rightarrow & w_2 M_2 \\ & \vdots & \\ N_n & \rightarrow & w_n M_n \\ \\ N_{n+1} & \rightarrow & w_{n+1} \\ N_{n+2} & \rightarrow & w_{n+2} \\ & \vdots & \\ N_{n+m} & \rightarrow & w_{n+m} \end{array}$$

  where N_i and M_i are nonterminals (not necessarily distinct) and $w_i \in \Sigma^*$ are strings of terminals.
- Thus, $w_i M_i$ is a semiword.
- At least one of the $N_i = S$. Assume that $N_1 = S$.
- Create a state of the TG for each nonterminal N_i and for each nonterminal M_j.

- Also create a state +.
- Make the state for nonterminal S the initial state of the transition graph.
- Draw an arc labeled with w_i from state N_i to state M_i if and only if there is a production $N_i \rightarrow w_i M_i$.
- Draw an arc labeled with w_i from state N_i to state + if and only if there is a production $N_i \rightarrow w_i$.
- Thus, we have created a TG.
- By considering the path developments of words accepted by the TG, we can show that there is a one-to-one correspondence between words accepted by TG and words in CFL.
- Thus, these are the same language.
- Kleene's Theorem implies that the language has a regular expression.
- Thus, language is regular.

∎

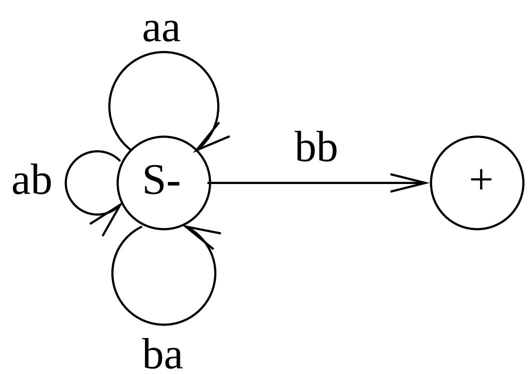
Remarks:

- all regular languages can be generated by some regular grammars (Theorem 21)
- all regular grammars generate some regular language.
- a regular language may have many CFG's that generate it, where some of the CFG's may not be regular grammars.

Example: CFG
productions:

$$S \rightarrow aaS \mid abS \mid baS \mid bb$$

Corresponding TG:



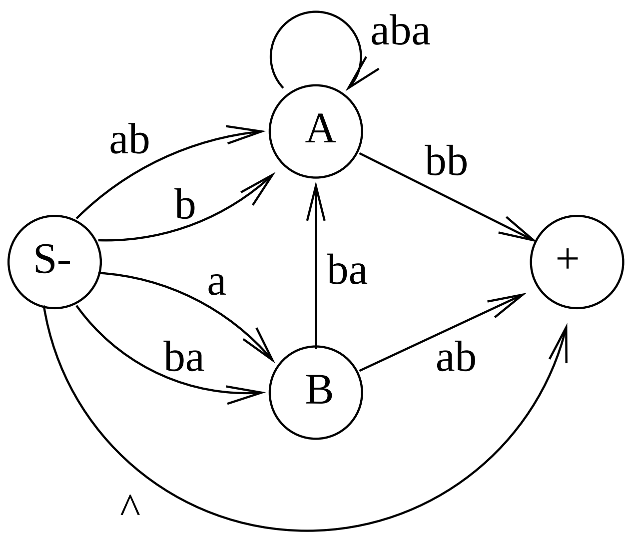
Below is another CFG that is not a regular grammar for the same language:

$$
\begin{aligned}
S &\rightarrow AaS \mid AbS \mid bAS \mid bb \\
A &\rightarrow a
\end{aligned}
$$

Example: CFG
productions:

$$\begin{aligned} S &\rightarrow aB \mid bA \mid abA \mid baB \mid \Lambda \\ A &\rightarrow abaA \mid bb \\ B &\rightarrow baA \mid ab \end{aligned}$$

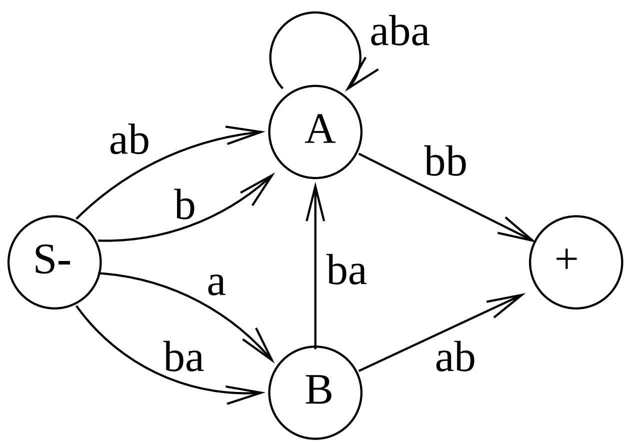
Corresponding TG:



Example: CFG
productions:

$$\begin{aligned} S &\rightarrow aB \mid bA \mid abA \mid baB \\ A &\rightarrow abaA \mid bb \\ B &\rightarrow baA \mid ab \end{aligned}$$

Corresponding TG (note that CFG does not generate Λ):



Definition: A production $(N, \mathcal{U}) \in R$ is a Λ-*production* if $\mathcal{U} = \Lambda$, i.e., the production is $N \to \Lambda$.

If CFG does not contain a Λ-production, then $\Lambda \notin$ CFL.

However, CFG may have Λ-production and $\Lambda \notin$ CFL.

Example: productions:

$$\begin{array}{rcl} S & \to & aX \\ X & \to & \Lambda \end{array}$$

13.2 Chomsky Normal Form

13.2.1 Λ Productions and Nullable Nonterminals

Recall we previously defined Λ-*production*:

$$N \to \Lambda$$

where N is some nonterminal.

Note that

- If some CFL contains the word Λ, then the CFG must have a Λ-production.
- However, if a CFG has a Λ-production, then the CFL does not necessarily contain Λ; e.g.,

  $$\begin{array}{rcl} S & \to & aX \\ X & \to & \Lambda \end{array}$$

  which defines the CFL $\{a\}$.

Definition: For a given CFG with Ω as its set of nonterminals and Σ as its set of terminals, a *working string* $W \in (\Sigma + \Omega)^*$ is any string of nonterminals and/or terminals that can be generated from the CFG starting from any nonterminal.

Example: CFG:

$$\begin{aligned} S &\to a \mid Xb \mid aYa \\ X &\to Y \mid \Lambda \\ Y &\to X \mid a \end{aligned}$$

Then in the derivation

$$\begin{aligned} S &\Rightarrow aYa \\ &\Rightarrow aXa \\ &\Rightarrow aa \end{aligned}$$

we have that aYa, aXa, and aa are all working strings.

Definition: For a given CFG having a nonterminal X and W a possible working string, we use the notation

$$X \stackrel{*}{\Rightarrow} W$$

if there is some derivation in the CFG starting from X that can result in the working string W.

Example: CFG:

$$\begin{aligned} S &\to a \mid Xb \mid aYa \\ X &\to Y \mid \Lambda \\ Y &\to X \mid a \end{aligned}$$

Since we have the following derivation

$$\begin{aligned} S &\Rightarrow aYa \\ &\Rightarrow aXa \\ &\Rightarrow aa \end{aligned}$$

we can write $S \stackrel{*}{\Rightarrow} aYa$ and $S \stackrel{*}{\Rightarrow} aXa$ and $S \stackrel{*}{\Rightarrow} aa$.

Definition: In a given CFG, a nonterminal X is *nullable* if

1. There is a production $X \to \Lambda$, or

2. $X \stackrel{*}{\Rightarrow} \Lambda$; i.e., there is a derivation that starts at X and leads to Λ:

$$X \Rightarrow \cdots \Rightarrow \Lambda$$

Example: CFG:

$$\begin{aligned} S &\rightarrow a \mid Xb \mid aYa \\ X &\rightarrow Y \mid \Lambda \\ Y &\rightarrow X \mid a \end{aligned}$$

has nullable nonterminals X, Y.

Example: CFG:

$$\begin{aligned} S &\rightarrow X \mid XY \mid Z \\ X &\rightarrow Z \mid \Lambda \\ Y &\rightarrow Wa \mid a \\ Z &\rightarrow WX \mid aZ \mid Zb \\ W &\rightarrow XYZ \mid bXa \mid \Lambda \end{aligned}$$

has nullable nonterminals S, X, Z, W.

Definition: For any language L, define the language L_0 as follows:

1. if $\Lambda \notin L$, then L_0 is the entire language L, i.e., $L_0 = L$.
2. if $\Lambda \in L$, then L_0 is the language $L - \{\Lambda\}$; i.e., if we let $T = \{\Lambda\}$, then $L_0 = L \cap T'$, so L_0 is all words in L except Λ.

Theorem 23 *If L is a CFL generated by a CFG G_1 that includes Λ-productions, then there is another CFG G_2 with no Λ-productions that generates L_0.*

Basic Idea.

- We give constructive algorithm to convert CFG G_1 with Λ-productions into equivalent CFG G_2 with no Λ-productions:

  1. Delete all Λ-productions.

2. For each production

$$X \to \text{something}$$

with at least one nullable nonterminal on the right-hand side, do the following for *each possible nonempty subset* of nullable nonterminals on the RHS:

   (a) create a new production

$$X \to \text{new something}$$

   where the new RHS is the same as the old RHS except with the entire current subset of nullable nonterminals removed.

   (b) do not create the production

$$X \to \Lambda$$

Example: CFG G_1

$$\begin{aligned} S &\to a \mid Xb \mid aYa \\ X &\to Y \mid \Lambda \\ Y &\to X \mid a \end{aligned}$$

has nullable nonterminals X, Y.

We create new productions:

Original Production	New Production
$S \to Xb$	$S \to b$
$S \to aYa$	$S \to aa$
$X \to Y$	Nothing
$Y \to X$	Nothing

New CFG G_2:

$$\begin{aligned} S &\to a \mid Xb \mid aYa \mid b \mid aa \\ X &\to Y \\ Y &\to X \mid a \end{aligned}$$

Example: CFG G_1

$$
\begin{array}{rcl}
S & \rightarrow & X \mid XY \mid Z \\
X & \rightarrow & Z \mid \Lambda \\
Y & \rightarrow & Wa \mid a \\
Z & \rightarrow & WX \mid aZ \mid Zb \\
W & \rightarrow & XYZ \mid bXa \mid \Lambda
\end{array}
$$

has nullable nonterminals S, X, Z, W.

We create new productions:

Original Production	New Production
$S \rightarrow X$	Nothing
$S \rightarrow XY$	$S \rightarrow Y$
$S \rightarrow Z$	Nothing
$X \rightarrow Z$	Nothing
$Y \rightarrow Wa$	$Y \rightarrow a$
$Z \rightarrow WX$	$Z \rightarrow W$ and $Z \rightarrow X$
$Z \rightarrow aZ$	$Z \rightarrow a$
$Z \rightarrow Zb$	$Z \rightarrow b$
$W \rightarrow XYZ$	$W \rightarrow YZ$, $W \rightarrow XY$, and $W \rightarrow Y$
$W \rightarrow bXa$	$W \rightarrow ba$

New CFG G_2:

$$
\begin{array}{rcl}
S & \rightarrow & X \mid XY \mid Z \mid Y \\
X & \rightarrow & Z \\
Y & \rightarrow & Wa \mid a \\
Z & \rightarrow & WX \mid aZ \mid Zb \mid W \mid X \mid a \mid b \\
W & \rightarrow & XYZ \mid bXa \mid YZ \mid XY \mid Y \mid ba
\end{array}
$$

- We need to show two things:
  1. all non-Λ words generated using original CFG G_1 can be generated using new CFG G_2.
  2. all words generated using new CFG G_2 can be generated using original CFG G_1.

- First we show that all non-Λ words generated using original CFG G_1 can be generated using new CFG G_2.
  - Suppose our CFG G_1 included the productions $A \to bBb$ and $B \to \Lambda$.
  - Suppose we had the following derivation of a word:

$$\begin{array}{rll} S & \Rightarrow & \ldots \\ & \Rightarrow & baAaAa \\ & \Rightarrow & babBbaAa \qquad \text{from } A \to bBb \\ & \Rightarrow & \ldots \\ & \Rightarrow & babBbaabAa \\ & \Rightarrow & bbabbaabAa \qquad \text{from } B \to \Lambda \\ & \Rightarrow & \ldots \end{array}$$

  - There would have been no difference if we had applied the production $A \to bb$ rather than $A \to bBb$ in the third line.
  - More generally, we can see that any non-Λ word generated using original CFG G_1 can be generated using new CFG G_2.
- Now show that all words generated using new CFG G_2 can be generated using original CFG G_1.
  - Note that each new production is just a combination of old productions (e.g., $X \to aYa$ and $Y \to \Lambda$).
  - Can show that any derivation using G_2 has a corresponding derivation using G_1 that possibly uses a Λ-production.
  - Hence, all words generated using new CFG G_2 can be generated using original CFG G_1.

■

13.2.2 Unit Productions

Definition: A production $(N, \mathcal{U}) \in R$ is a *unit production* if $\mathcal{U} \in \Omega$; i.e., the production is of the form

$$\text{one nonterminal} \to \text{one nonterminal}$$

Theorem 24 *If a language L is generated by a CFG G_1 that has no Λ-productions, then there is also a CFG G_2 for L with no Λ-productions and no unit productions.*

Basic Idea.

- Use the following rules to create new CFG:
- For each pair of nonterminals A and B such that there is a production

  $$A \rightarrow B$$

  or a chain of productions (unit derivation)

  $$A \stackrel{*}{\Rightarrow} B,$$

  introduce the following new productions:

  - if the non-unit productions from B are

    $$B \rightarrow s_1 \mid s_2 \mid \ \ldots \ \mid s_n$$

    where the $s_i \in (\Sigma + \Omega)^*$ are strings of terminals and nonterminals, then create the new productions

    $$A \rightarrow s_1 \mid s_2 \mid \ \ldots \ \mid s_n$$

  - Do the same for all such pairs A and B simultaneously.
  - Remove all unit productions.
- Can show that G_1 and G_2 generate the same language.

∎

Example: CFG G_1:

$$\begin{array}{rcl} S & \rightarrow & X \mid Y \mid bb \\ X & \rightarrow & Z \mid aXY \\ Y & \rightarrow & Xa \mid a \\ Z & \rightarrow & YX \mid S \mid Zb \end{array}$$

has unit productions and unit derivations

$$
\begin{aligned}
S &\rightarrow X \\
S &\rightarrow Y \\
X &\rightarrow Z \\
Z &\rightarrow S \\
S &\Rightarrow X \Rightarrow Z \\
X &\Rightarrow Z \Rightarrow S \\
Z &\Rightarrow S \Rightarrow X \\
Z &\Rightarrow S \Rightarrow Y \\
X &\Rightarrow Z \Rightarrow S \Rightarrow Y
\end{aligned}
$$

We create new productions:

Original Unit Production (Derivation)	New Productions
$S \rightarrow X$	$S \rightarrow aXY$
$S \rightarrow Y$	$S \rightarrow Xa \mid a$
$S \Rightarrow X \Rightarrow Z$	$S \rightarrow YX \mid Zb$
$X \rightarrow Z$	$X \rightarrow YX \mid Zb$
$X \Rightarrow Z \Rightarrow S$	$X \rightarrow bb$
$X \Rightarrow Z \Rightarrow S \Rightarrow Y$	$X \rightarrow Xa \mid a$
$Z \rightarrow S$	$Z \rightarrow bb$
$Z \Rightarrow S \Rightarrow X$	$Z \rightarrow aXY$
$Z \Rightarrow S \Rightarrow Y$	$Z \rightarrow Xa \mid a$

New CFG G_2:

$$
\begin{aligned}
S &\rightarrow bb \mid aXY \mid Xa \mid a \mid YX \mid Zb \\
X &\rightarrow aXY \mid YX \mid Zb \mid bb \mid Xa \mid a \\
Y &\rightarrow Xa \mid a \\
Z &\rightarrow YX \mid Zb \mid bb \mid aXY \mid Xa \mid a
\end{aligned}
$$

Theorem 25 *Consider a CFG* $G_1 = (\Sigma, \Omega_1, R_1, S_1)$*, which generates language* $L_1 = L(G_1)$*. Then there exists another CFG* $G_2 = (\Sigma, \Omega_2, R_2, S_2)$ *such that*

- $L(G_2) = L(G_1) - \{\Lambda\}$ *and*
- $(N, u) \in R_2$ *implies* $u \in \Omega^+ + \Sigma$*;*

i.e., G_2 generates all non-Λ strings of L_1 and each production in G_2 is of one of two basic forms:

1. *Nonterminal $\rightarrow$ string of only Nonterminals*
2. *Nonterminal $\rightarrow$ one terminal*

Basic Idea. We will give a constructive proof:

- Assume that the nonterminals in the CFG G_1 are $S, X_1, X_2, \ldots, X_n$.
- Assume that the terminals in the CFG G_1 are a and b.
- Introduce two new nonterminals A and B.
- Introduce two new productions:

$$\begin{aligned} A &\rightarrow a \\ B &\rightarrow b \end{aligned}$$

- For each original production involving terminals,
  - replace each a with the nonterminal A
  - replace each b with the nonterminal B

  Example: Original production in G_1:

$$X_5 \rightarrow X_1 aba X_3 bb X_2$$

  becomes new production in G_2:

$$X_5 \rightarrow X_1 ABA X_3 BB X_2$$

  which is string of only Nonterminals.

  Example: Production in G_1:

$$X_2 \rightarrow abaab$$

  becomes new production in G_2:

$$X_2 \rightarrow ABAAB$$

  which is string of only nonterminals.

- So now all original productions have been transformed into new productions that have only nonterminals on the RHS.
- Also, we have two new productions for A and B.
- Note that any derivation starting from S to produce the word

$$ababba$$

  will now follow same sequence of (new) productions to derive the string

$$ABABBA$$

  starting from S.
- Then apply $A \to a$ and $B \to b$ the proper number of times to get the word $ababba$.
- Hence, any word generated by the original CFG G_1 can be generated by the new CFG G_2.
- Need to show that any word generated by the new CFG G_2 can also be generated by the original CFG G_1.
  - Consider new CFG G_2 without the two productions $A \to a$ and $B \to b$.
  - Applying the new productions numerous times will result in a string of A's and B's.
  - Applying corresponding original productions in same order will result in the same string with a's and b's.
  - Then change string of A's and B's into a's and b's using $A \to a$ and $B \to b$.
  - Thus, every word generated using new CFG G_2 can also be generated using original CFG G_1.

■

Example: CFG G_1:

$$\begin{aligned} S &\to abSba \mid bX_1aX_1 \mid X_2 \mid bb \\ X_1 &\to aa \mid aSX_1b \\ X_2 &\to X_1a \mid a \end{aligned}$$

can be transformed into new CFG G_2:

$$\begin{aligned} S &\to ABSBA \mid BX_1AX_1 \mid X_2 \mid BB \\ X_1 &\to AA \mid ASX_1B \\ X_2 &\to X_1A \mid A \\ A &\to a \\ B &\to b \end{aligned}$$

13.2.3 Chomsky Normal Form

Definition: A CFG $G = (\Sigma, \Omega, R, S)$ is in *Chomsky Normal Form* (CNF) if $(N, \mathcal{U}) \in R$ implies $\mathcal{U} \in (\Omega\Omega) + \Sigma$; i.e., each of its productions has one of the two forms:

1. Nonterminal $\to$ string of exactly two Nonterminals
2. Nonterminal $\to$ one terminal

Theorem 26 *For any CFL L, the non-Λ words of L can be generated by a CFG in CNF.*

Basic Idea. By construction:

- Let $L_0 = L$ if $\Lambda \notin L$, and $L_0 = L - \{\Lambda\}$ if $\Lambda \in L$.
- By Theorem 23, we know there is a CFG for L_0 that has no Λ-productions.
- By Theorem 24, we know there is a CFG for L_0 that has no unit productions.
- By Theorem 25, we know there is a CFG for L_0 for which each of its productions are of one of two forms:
  1. Nonterminal $\to$ string of only nonterminals
  2. Nonterminal $\to$ one terminal
- So now assume that our CFG for L_0 has the above three properties.

- Do nothing to the productions of the form

  Nonterminal → one terminal

- For each production of the form

  Nonterminal → string of Nonterminals

  we expand it into a collection of productions as follows:

  - Suppose we have the production

    $$X_4 \to X_2X_5X_3X_2X_1$$

  - Replace the production with the new productions

    $$\begin{array}{rcl} X_4 & \to & X_2R_1 \\ R_1 & \to & X_5R_2 \\ R_2 & \to & X_3R_3 \\ R_3 & \to & X_2X_1 \end{array}$$

    where the R_i are new nonterminals.

  - For each transformation of original productions, introduce new nonterminals R_i.

- This transformation creates a new CFG in CNF.

- Now we have to show that the language generated by the new CFG is the same as that generated by the original CFG.

- First show that any word that can be generated by original CFG can also be generated by new CFG:

  - In any derivation of a word using the original CFG, we just replace any production of the form

    $$X_4 \to X_2X_5X_3X_2X_1$$

    with the new productions

    $$\begin{array}{rcl} X_4 & \to & X_2R_1 \\ R_1 & \to & X_5R_2 \\ R_2 & \to & X_3R_3 \\ R_3 & \to & X_2X_1 \end{array}$$

    - This gives us a derivation of the word using the new CFG.

- Now show that any word that can be generated by the new CFG can also be generated by the original CFG:

    - Note that the nonterminal R_3 is only used in the RHS of the production
    $$R_2 \to X_3 R_3$$
    - Thus, that is the only way R_3 would arise.
    - Similarly, the nonterminal R_2 is only used in the RHS of the production
    $$R_1 \to X_5 R_2$$
    - Thus, that is the only way R_2 would arise.
    - We can similarly show the same for all new nonterminals R_i
    - Thus, since we use different R_i's in the expansion of each production, the new nonterminals R_i cannot interact to create new words.

■

Example: CFG

$$\begin{aligned} S &\to abSba \mid bX_1aX_2 \mid bb \\ X_1 &\to aa \mid aSX_1b \\ X_2 &\to X_1a \mid abb \end{aligned}$$

can be transformed into new CFG

$$\begin{aligned} S &\to ABSBA \mid BX_1AX_2 \mid BB \\ X_1 &\to AA \mid ASX_1B \\ X_2 &\to X_1A \mid ABB \\ A &\to a \\ B &\to b \end{aligned}$$

which can then be transformed into a CFG in CNF:

$$\begin{aligned} S &\to AR_1 \\ R_1 &\to BR_2 \end{aligned}$$

$$
\begin{aligned}
R_2 &\rightarrow SR_3 \\
R_3 &\rightarrow BA \\
S &\rightarrow BR_4 \\
R_4 &\rightarrow X_1R_5 \\
R_5 &\rightarrow AX_2 \\
S &\rightarrow BB \\
X_1 &\rightarrow AA \\
X_1 &\rightarrow AR_6 \\
R_6 &\rightarrow SR_7 \\
R_7 &\rightarrow X_1B \\
X_2 &\rightarrow X_1A \\
X_2 &\rightarrow AR_8 \\
R_8 &\rightarrow BB \\
A &\rightarrow a \\
B &\rightarrow b
\end{aligned}
$$

13.3 Leftmost Nonterminals and Derivations

Definition: The *leftmost nonterminal* (LMN) in a working string is the first nonterminal that we encounter when we scan the string from left to right.

Example: In the string $bbabXbaYSbXbY$, the LMN is X.

Definition: If a word w is generated by a CFG by a certain derivation and at each step in the derivation, a rule of production is applied to the leftmost nonterminal in the working string, then this derivation is called a *leftmost derivation* (LMD).

Example: CFG:

$$\begin{aligned} S &\rightarrow baXaS \mid ab \\ X &\rightarrow Xab \mid aa \end{aligned}$$

The following is a LMD:

$$\begin{aligned} S &\Rightarrow baXaS \\ &\Rightarrow baXabaS \\ &\Rightarrow baXababaS \\ &\Rightarrow baaaababaS \\ &\Rightarrow baaaababaab \end{aligned}$$

Example: CFG:

$$\begin{aligned} S &\rightarrow XY \\ X &\rightarrow Yb \mid Xa \mid aa \mid YY \\ Y &\rightarrow XbbX \mid ab \end{aligned}$$

The word *abbaaabbabab* has the following derivation tree:

```
          S
         / \
        /   \
       /     \
      /       \
     /         \
    X         _ Y _
   / \       / / \ \
  Y   b     / |   | \
 / \       X  b   b  X
a   b     / \       / \
         X   a     /   \
        / \       Y     Y
       a   a     / \   / \
                a   b a   b
```

Note that if we walk around the tree starting down the left branch of the root with our left hand always touching the tree, then the order in which we first visit each nonterminal corresponds to the order in which the nonterminals are replaced in LMD.

This is true for any derivation in any CFG

Theorem 27 *Any word that can be generated by a given CFG by some derivation also has a LMD.*