Example

Given a context-free grammar with the following production rules, find the nullable variables:

$S \rightarrow ABC$

$A \rightarrow B \mid a$

$B \rightarrow C \mid b \mid \lambda$

$C \rightarrow AB \mid D$

$D \rightarrow Cd$

$N_0 = \{B\}$

$N_1 = \{B, A\}$

$N_2 = \{B, A, C\}$

$N_3 = \{B, A, C, S\}$

Example (continued)

$S \rightarrow ABC$
$A \rightarrow B \mid a$
$B \rightarrow C \mid b \mid \lambda$
$C \rightarrow AB \mid D$
$D \rightarrow Cd$

$S \rightarrow ABC$
$S \rightarrow ABC \mid BC \mid AC \mid AB \mid A \mid B \mid C \mid \varepsilon$

$C \rightarrow AB \mid D$
$C \rightarrow AB \mid A \mid B \mid D$

$D \rightarrow Cd$
$D \rightarrow Cd \mid d$

$N = \{A, B, C, S\}$

Example (continued)

$S \rightarrow ABC \mid AB \mid AC \mid BC \mid A \mid B \mid C \mid \varepsilon$
$A \rightarrow B \mid a$
$B \rightarrow C \mid b$
$C \rightarrow AB \mid A \mid B \mid D$
$D \rightarrow Cd \mid d$

Note that we have gotten rid of all λ-productions. However, other beneficial changes can still be made.

Chomsky Normal Form

There are other ways to limit the form a grammar can have.

A context-free grammar in Chomsky Normal Form (CNF) has all of its rules restricted so that there are no more than two symbols, either one terminal or two variables, on the right-hand side of a production rule.

This seems very restrictive, but actually every context-free grammar can be converted into Chomsky Normal Form.

Chomsky Normal Form

Definition 6.4: A context-free grammar is in Chomsky Normal Form (CNF) if every production is one of these two types:

$A \rightarrow BC$

$A \rightarrow a$

where A, B, and C are variables and a is a terminal symbol.

Chomsky normal form

For languages that include the empty string λ, the rule $S \to \lambda$ may also be allowed, where S is the start symbol, as long as S does not occur on the right-hand side of any rule

Chomsky Normal Form

Theorem 6.6: Any context-free grammar G = (V, T, S, P) with $\lambda \notin L(G)$ has an equivalent grammar G' = (V', T', S, P') in Chomsky Normal Form.

(Actually, for languages that include the empty string λ, the rule $S \rightarrow \lambda$ may also be allowed, where S is the start symbol, as long as S does not occur on the right-hand side of any rule.)

Examples:

$$S \to AS$$
$$S \to a$$
$$A \to SA$$
$$A \to b$$

Chomsky
Normal Form

$$S \to AS$$
$$S \to AAS$$
$$A \to SA$$
$$A \to aa$$

Not Chomsky
Normal Form

$$S \to ABa$$

$$A \to aab$$

$$B \to Ac$$

Not Chomsky

Normal Form

Chomsky Normal Form

- Step 1:
  - Remove λ-Productions
- Step 2:
  - Remove Unit Productions
- Step 3:
  - Remove useless symbols

Chomsky Normal Form: Proof by construction

Given a CFG grammar $G = (V, T, S, P)$, to convert it to Chomsky Normal Form:

1. Eliminate λ-productions and unit-productions from G, producing a CFG $G' = (V, T, S, P')$, such that
   $L(G') = L(G) - \{\lambda\}$.

2. Convert G' into $G'' = (V'', T, S, P'')$ so that every production is either of the form
   $A \rightarrow B_1 B_2 \ldots B_k$
   (where $k \geq 2$ and each B_i is a variable in V''),
   or of the form
   $A \rightarrow a$

Chomsky Normal Form

Basically, what you are doing in step 2 is restricting the right sides of productions to be either single terminals or strings of two or more variables.

What we don't want is strings of length > 2 that have one or more terminals in them. If we have strings like this, for every terminal *a* appearing in such a string:

1. Add a new variable, X_a and
   add a new production, $X_a \rightarrow a$
2. Replace *a* by X_a in all the productions where it appears (except those in the form $A \rightarrow a$).

Chomsky Normal Form (continued)

3. Convert G'' into G''' = (V''', T, S, P'''). To do this, replace each production having more than two variables on the right by an equivalent set of productions, each one having exactly two variables on the right. (Create new variables as necessary to accomplish this.)

For example:
the production $A \rightarrow BCD$ would be replaced with

$A \rightarrow BZ_1$
$Z_1 \rightarrow CD$

Done!

Introduce variables for terminals: T_a, T_b, T_c

$$S \to ABa$$
$$A \to aab$$
$$B \to Ac$$

$$S \to ABT_a$$
$$A \to T_aT_aT_b$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

Introduce intermediate variable: V_1

$$S \to ABT_a$$
$$A \to T_aT_aT_b$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

$\Rightarrow$

$$S \to AV_1$$
$$V_1 \to BT_a$$
$$A \to T_aT_aT_b$$
$$B \to AT_c$$
$$T_a \to a$$
$$T_b \to b$$
$$T_c \to c$$

Introduce intermediate variable: V_2

$$
\begin{aligned}
S &\to AV_1 \\
V_1 &\to BT_a \\
A &\to T_aT_aT_b \\
B &\to AT_c \\
T_a &\to a \\
T_b &\to b \\
T_c &\to c
\end{aligned}
$$

$$
\begin{aligned}
S &\to AV_1 \\
V_1 &\to BT_a \\
A &\to T_aV_2 \\
V_2 &\to T_aT_b \\
B &\to AT_c \\
T_a &\to a \\
T_b &\to b \\
T_c &\to c
\end{aligned}
$$

Final grammar in Chomsky Normal Form:

Initial grammar

$$S \to ABa$$

$$A \to aab$$

$$B \to Ac$$

$$S \to AV_1$$

$$V_1 \to BT_a$$

$$A \to T_a V_2$$

$$V_2 \to T_a T_b$$

$$B \to AT_c$$

$$T_a \to a$$

$$T_b \to b$$

$$T_c \to c$$

Replace any production $A \to C_1 C_2 \cdots C_n$

with $A \to C_1 V_1$

$$V_1 \to C_2 V_2$$

$$\cdots$$

$$V_{n-2} \to C_{n-1} C_n$$

New intermediate variables: $V_1, V_2, \ldots, V_{n-2}$

- Example:
  - $S \rightarrow AB$
  - $A \rightarrow aAA \mid \lambda$
  - $B \rightarrow bBB \mid \lambda$

- A and B are nullable since $A \rightarrow \varepsilon$ and $B \rightarrow \varepsilon$
- S is nullable since $S \rightarrow AB$ and A and B are nullable

  - Remove $A \rightarrow \lambda$ and $B \rightarrow \lambda$
  - Our final grammar looks like:
    - $S \rightarrow AB \mid A \mid B \mid \varepsilon$
    - $A \rightarrow aAA \mid aA \mid a$
    - $B \rightarrow bBB \mid bB \mid b$

- Removing unit transitions:
  - $S \rightarrow AB \mid aAA \mid aA \mid a \mid bBB \mid bB \mid b \mid \varepsilon$
  - $A \rightarrow aAA \mid aA \mid a$
  - $B \rightarrow bBB \mid bB \mid b$
- Note that S, A, and B are all useful.

- Define new productions: $X_a \rightarrow a$ and $X_b \rightarrow b$ and replace instance of a with X_a, similarly for b
    - $S \rightarrow AB \mid aAA \mid aA \mid a \mid bBB \mid bB \mid b \mid \varepsilon$
    - $A \rightarrow aAA \mid aA \mid a$
    - $B \rightarrow bBB \mid bB \mid b$
  - New:
    - $S \rightarrow AB \mid X_a AA \mid X_a A \mid a \mid X_b BB \mid X_b B \mid b \mid \varepsilon$
    - $A \rightarrow X_a AA \mid X_a A \mid a$
    - $B \rightarrow X_b BB \mid X_b B \mid b$
    - $X_a \rightarrow a$
    - $X_b \rightarrow b$

- $S \rightarrow AB \mid \underline{X_a\ AA} \mid X_a\ A \mid a \mid \underline{X_b\ BB} \mid X_b\ B \mid b \mid \varepsilon$
- $A \rightarrow \underline{X_a\ AA} \mid X_a\ A \mid a$
- $B \rightarrow \underline{X_b\ BB} \mid X_b\ B \mid b$
- $X_a \rightarrow a$
- $X_b \rightarrow b$

- Add productions
  - $Y_1 \rightarrow AA$
  - $Y_2 \rightarrow BB$

- Our final grammar
  - $S \rightarrow AB \mid X_a\ Y_1 \mid X_a\ A \mid a \mid X_b\ Y_2 \mid X_b\ B \mid b \mid \varepsilon$
  - $A \rightarrow X_a\ Y_1 \mid X_a\ A \mid a$
  - $B \rightarrow X_b\ Y_2 \mid X_b\ B \mid b$
  - $Y_1 \rightarrow AA$
  - $Y_2 \rightarrow BB$
  - $X_a \rightarrow a$
  - $X_b \rightarrow b$

Example

Original grammar:

$S \rightarrow AB \mid ab$

$A \rightarrow ABAB \mid BA$

$B \rightarrow ab \mid b$

After step 2:

$S \rightarrow AB \mid X_a X_b$

$X_a \rightarrow a$

$X_b \rightarrow b$

$A \rightarrow ABAB \mid BA$

$B \rightarrow X_a X_b \mid b$

Example

After step 2:

$S \rightarrow AB \mid X_a X_b$

$X_a \rightarrow a$

$X_b \rightarrow b$

$A \rightarrow ABAB \mid BA$

$B \rightarrow X_a X_b \mid b$

After step 3:

$S \rightarrow AB \mid X_a X_b$

$X_a \rightarrow a$

$X_b \rightarrow b$

$A \rightarrow AY_1 \mid BA$

$Y_1 \rightarrow BY_2$

$Y_2 \rightarrow AB$

$B \rightarrow X_a X_b \mid b$

Example

If you recognize that
$A \rightarrow ABAB$
has two copies of the same pair of variables, you could substitute the following instead:
(but the first procedure works equally well)

After step 3:

$S \rightarrow AB \mid X_a X_b$

$X_a \rightarrow a$

$X_b \rightarrow b$

$A \rightarrow Y_1 Y_1 \mid BA$

$Y_1 \rightarrow AB$

$B \rightarrow X_a X_b \mid b$

Proof (concluded)

This constitutes a proof by construction that any CFG can be converted to CNF.

Later, this will be used to prove that there are languages which are not context-free.

Example

G:

S => AS | BABC

A => A1 | 0A1 | 01

B => 0B | 0

C => 1C | 1

G in CNF:

X0 => 0
X1 => 1
S => AS | BY1
Y1 => AY2
Y2 => BC
A => AX1 | X0Y3 | X0X1
Y3 => AX1
B => X0B | 0
C => X1C | 1

All productions are of the form: A=>BC or A=>a

Example

$P:$

$S \to aB$
$S \to bA$
$A \to aS$
$A \to bAA$
$A \to a$
$B \to bS$
$B \to aBB$
$B \to b$

$S \to C_a B$
$S \to C_b A$
$A \to C_a S$
$A \to C_b D_1$
$A \to a$
$B \to C_b S$
$B \to C_a D_2$
$B \to b$
$C_a \to a$
$C_b \to b$
$D_1 \to AA$
$D_2 \to BB$

$S \to C_a B$

$S \to C_b A$

$A \to C_a S$

$A \to C_b D_1$

$A \to a$

$B \to C_b S$

$B \to C_a D_2$

$B \to b$

$C_a \to a$

$C_b \to b$

$D_1 \to AA$

$D_2 \to BB$

P: $S \to C_a B \mid C_b A$

$A \to C_a S \mid C_b D_1 \mid a$

$B \to C_b S \mid C_a D_2 \mid b$

$C_a \to a$

$C_b \to b$

$D_1 \to AA$

$D_2 \to BB$

Greibach Normal Form

Greibach Normal Form is similar to Chomsky Normal Form, except that every production is of the form $A \to ax$, where a is a terminal symbol and x is a string of zero or more variables.
Note that GNF puts a limit on where terminals and variables can appear – restrictions on their relative positions – rather than on the number of symbols on the right-hand side of the production rules.

Greibach Normal Form

Definition 6.5: A context-free grammar is said to be in Greibach Normal Form if all productions have the form

$$A \rightarrow ax$$

where $a \in T$ and $x \in V^*$

Examples:

$$S \to cAB$$

$$A \to aA \mid bB \mid b$$

$$B \to b$$

Greibach
Normal Form

$$S \to abSb$$

$$S \to aa$$

Not Greibach
Normal Form

Greibach Normal Form

Example:

Convert the following grammar into GNF:

$$S \rightarrow abSb \mid aa$$

Introduce new variables A and B to stand for a and b respectively, and substitute:

$$S \rightarrow aBSB \mid aA$$
$$A \rightarrow a$$
$$B \rightarrow b$$

Greibach Normal Form

Theorem 6.7: Any context-free grammar G = (V, T, S, P) with $\lambda \notin$ L(G) has an equivalent grammar G' = (V', T', S, P') in Greibach Normal Form.

It is hard to prove this, and it is hard to construct an easy-to implement algorithm for performing the conversion.