

Normalization of Context-free Grammars

Prof. (Dr.) K.R. Chowdhary

Former Professor & Head, Department of Computer Sc. & Engineering
MBM Engineering College, Jodhpur

K.R. Chowdhary

Theory of Computation

Automata, Formal Languages, Computation and Complexity

Focuses on pedagogy in its writing, that represents a refreshing approach

Ensures comprehensive and enjoyable learning

Undergone a rigorous classroom testing



© 2025



Get 20% off with this code: **SPRAUT**

Available on Springer Nature Link



[link.springer.com/book/
9789819762347](https://link.springer.com/book/9789819762347)

Please note that promotional coupons are only valid for English-language Springer, Apress, and Palgrave Macmillan books & eBooks and are redeemable on link.springer.com only. Titles affected by fixed book price laws, forthcoming titles and titles temporarily not available on Springer Nature Link are excluded from promotions, as are reference works, handbooks, encyclopedias, subscriptions, or bulk purchases. The currency in which your order will be invoiced depends on the billing address associated with the payment method used, not necessarily your home currency. Regional VAT/tax may apply. Promotional prices may change due to exchange rates. Promotions are valid for individual customers only. Booksellers, book distributors, and institutions such as libraries and corporations please visit springernature.com/contact-us. Promotions do not work in combination with other discounts or gift cards.

Introduction to Simplification of Grammars

- Parsing becomes easier if we add full generality in the grammar G .
- In simplification of G , we remove 1) null-productions (also called ε -productions), 2) unit-productions (chain-rules), and 3) non-reachable symbols and corresponding productions.
- But, the generating power of the grammar remains the same.
- A variable symbol is **useful** if it appears in some derivation, otherwise it is **useless**.
- if $\varepsilon \notin L(G)$, then all the ε -productions can be removed from the grammar.
- **Normal Forms:** Is some restrictions in the productions rules, for the right hand side of a production, so that all the productions are in some standard form. Two types normal forms exists:
 - CNF: Chomsky Normal Form Grammar:** All the productions are like: $A \rightarrow BC$, $A \rightarrow a$, where $A, B, C \in V$ and $a \in \Sigma$.
 - GNF: Greibach Normal Form Grammar:** All the productions are like $A \rightarrow a\alpha$, where $a \in \Sigma$ and $\alpha \in V^*$.

ϵ -productions, useless productions

- Productions of the form $A \rightarrow \epsilon$ are called null productions. All these productions can be eliminated from grammar if $\epsilon \notin L(G)$. Hence, in this case the generating power of G remains unchanged.
- If $A \Rightarrow^* \epsilon$ then variable A is called *nullable*.
- If $B \rightarrow X_1 X_2 \dots X_n$, then X_i can be dropped in this production if $\exists X_i \rightarrow \epsilon$. Such X_i 's are called nullable. But if $B \rightarrow \epsilon$, then X_i 's cannot be eliminated.
- **Useless symbols and productions:** Given the derivation $S \Rightarrow^* \alpha X \beta \Rightarrow^* w \in \Sigma^*$, where $\alpha, \beta \in (V \cup \Sigma)^*$, here X is useful because it appears in some derivation.

Example

$S \rightarrow aSb | \epsilon$, $B \rightarrow bB$, has B useless symbol and $B \rightarrow bB$ as useless production. This is because B is not reachable from S , neither B terminates to null or terminal.

Example

$S \rightarrow A$, $A \rightarrow aA | \epsilon$, $B \rightarrow bA$, has B as useless symbol and $B \rightarrow bA$ as useless production.

Theorem

Theorem

If $G = (V, \Sigma, S, P)$ with $L(G) \neq \emptyset$, then there is an equivalent grammar $G' = \{V', \Sigma, S, P'\}$, such that G' does not contain useless symbols and productions, and $L(G) = L(G')$.

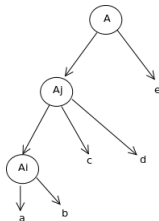
Proof.

We follow the following algorithm to systematically construct G' :

1. For $A_i \rightarrow u$, where $u \in \Sigma^*$, move A_i to V' , and $A_i \rightarrow u$ to P' .
2. $V = V - \{A_i\}$, $P = P - \{A_i \rightarrow u\}$.
3. while $\exists A_j \rightarrow X_1 X_2 \dots X_n \in P$, where $X_i \in V'$, or $X_i \in \Sigma$ do
 - a. $V' = V' \cup \{A_j\}$,
 - b. $P' = P' \cup \{A_j \rightarrow X_1 X_2 \dots X_n\}$,
 - c. $V = V - \{A_j\}$, $P = P - \{A_j \rightarrow X_1 X_2 \dots X_n\}$enddo

The tree for above is shown in next slide. The grammar constructed as G' does not contains null productions as the same were removed in advance, it does not contain the useless symbols and productions. □

Theorem continued ...



The figure shows that first of all we construct the bottom most subtree, A_i , then A_j , and then A , till finally to S . This way only reachable symbols from S , and corresponding productions are move to grammar G' . Therefore, no null or useless productions or symbols have been moved into the grammar G' . Thus generating power of G and G' are same.

Removing unit productions

- Productions of the form $A \rightarrow B$, where $A, B \in V$ are called unit productions.
- Having removed useless symbols and productions, and ϵ -productions, we remove the unit-productions.
- **Example:** Given productions $A \rightarrow B, B \rightarrow bB|c$, it can be substituted by single production:

$$A \rightarrow bB|c. \quad \square$$

- If there is sequence of unit productions, and it gives a look of chain like: $A \Rightarrow^* B$, then all the unit productions can be removed systematically. But, if there is situation like: $A \Rightarrow^* B$, due to $A \Rightarrow BC \Rightarrow B$, and $C \rightarrow \epsilon$, then $A \Rightarrow^* B$ is not a chain.

CNF Theorem

Theorem

If there is grammar $G = (V, \Sigma, S, P)$, there exists an equivalent grammar $G' = (V', \Sigma', S, P')$ without unit-productions, ε -productions, and useless symbols and productions, and is CNF.

Proof.

- The ε -productions, unit-productions, and useless-productions and symbols can be removed using the methods discussed earlier. This does not effect the generating power of the grammar.
- The grammar available now has following format: $A \rightarrow X_1 X_2 \dots X_n$. For $n = 1$, the right hand side of a production has single symbol. This will be terminal only, as all the unit productions have been removed.
- For $n \geq 2$, there is $X_i \in (V \cup \Sigma)$. If X_i is terminal, say a , then substitute a by C_i and introduce a new production $C_i \rightarrow a$.



- All the productions are now of the form: $A \rightarrow a$ or $A \rightarrow C_1 C_2 \dots C_n$. All the productions like $A \rightarrow C_1 C_2 \dots C_n$ are reduced to $A \rightarrow BC$ as follows: ($A \rightarrow C_1 C_2$ is already CNF). For $n > 2$, modify the productions as follows:

$$A \rightarrow C_1 D_1$$

$$D_1 \rightarrow C_2 D_2$$

...

$$D_{n-3} \rightarrow C_{n-2} D_{n-2}$$

$$D_{n-2} \rightarrow C_{n-1} D_n.$$

This proves the theorem. \square



Chowdhary, K.R. (2025). Regular Languages. In: Theory of Computation. Springer, Singapore.

https://doi.org/10.1007/978-981-97-6234-7_7