

# Closure properties of Context-free languages and Grammars

Prof. (Dr.) K.R. Chowdhary

*Email: kr.chowdhary@iitj.ac.in*

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

Saturday 19<sup>th</sup> July, 2025

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](https://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](https://springernature.com/contact-us). Promotions do not work in combination with other discounts or gift cards.

# Closure properties of CNF

- **Intersection of two CFLs:** Let  $G_1, G_2$  be two context-free grammars.

$G_1 :$

$S \rightarrow AB, S \rightarrow A, A \rightarrow 0A1$

$B \rightarrow 0B, B \rightarrow 0$

$\therefore L(G_1) = \{0^n 1^n 0^+\}$

$\therefore L_1 \cap L_2 = 0^n 1^n 0^n \notin CFL \text{ for } n \geq 1.$

$G_2 :$

$S \rightarrow BA, S \rightarrow A, A \rightarrow 1A0$

$A \rightarrow 10, B \rightarrow 0B, B \rightarrow 0$

$\therefore L(G_2) = \{0^+ 1^n 0^n\}$

- **Union of two CFLs:** For  $L_1 = (G_1)$  and  $L_2 = (G_2)$ ,  $L_1 \cup L_2 \in CFL$ .

$S \rightarrow S_1 | S_2$ , and  $V_1 \cap V_2 = \emptyset$

$P = P_1 \cup P_2 \cup \{S \rightarrow S_1 | S_2\}$ ,  $V = V_1 \cup V_2 \cup \{S\}$ ,  $\Sigma = \Sigma_1 \cup \Sigma_2$ .

- **Concatenation of two CFLs:** For  $L_1 = (G_1)$  and  $L_2 = (G_2)$ ,  $L_1 \circ L_2 \in CFL$ .

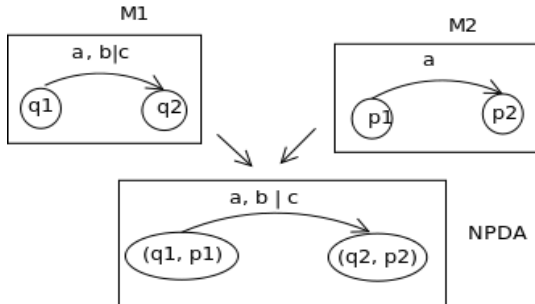
$S \rightarrow S_1 \circ S_2$ , and  $V_1 \cap V_2 = \emptyset$

$P = P_1 \cup P_2 \cup \{S \rightarrow S_1 \circ S_2\}$ ,  $V = V_1 \cup V_2 \cup \{S\}$ ,  $\Sigma = \Sigma_1 \cup \Sigma_2$ .

- **Kleene star of two CFLs:** For  $L_1 = (G_1)$  and  $L_2 = (G_2)$ ,  $L_1^* \in CFL$ , where  $S \rightarrow S_1 S | \epsilon$ ,  $V_1 \cap V_2 = \emptyset$ .

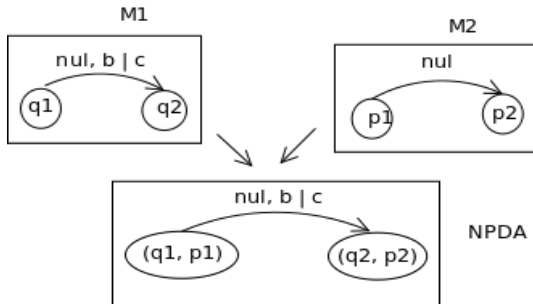
# Closure properties of CFLs

- **CFL**  $\cap$  **Reg.** lang  $\in$  **CFL**
- Let  $M_1$  is NPDA accepting CF language  $L_1$  by final state, and  $M_2$  be a FA accepting  $L_2$ . The PDA recognizing  $L_1 \cap L_2$  simulates  $P$  and  $M$  simultaneously, like cross-product of two FA.
- We construct new NPDA  $M$  for  $L_1 \cap L_2$  to simulate  $M_1$  and  $M_2$  in parallel.



# Closure properties of CFLs

- **CFL  $\cap$  Reg. lang  $\in$  CFL ...**



- **Simulating start state:** For  $q_0 \in M_1, p_0 \in M_2$  there is  $(q_0, p_0) \in M$
- **Simulating final state:** For  $q_1 \in F_1$ , and  $p_1, p_2 \in F_2$  there is  $(q_1, p_1), (q_1, p_2) \in F$ .

- **Membership problem:** For CFG  $G_1$ , find if  $w \in L(G)$  ?

The membership algorithm is: Parser. That is, if we are able to obtain a parse-tree for given word  $w$ , then  $w \in L(G)$  else not.

- **Empty Language:** Is  $L(G) = \emptyset$ ?

Algorithm:

1. Remove useless symbols
2. Check if start symbol is useless? If yes, then  $L(G) = \emptyset$  else not.

- **Infinite Language Problem:** Is  $L = L(G)$  an infinite language?

Algorithm:

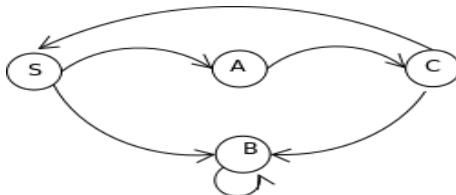
1. remove useless symbols
2. remove null and unit productions
3. create dependency graph for variables
4. if there is a loop in the dependency graph, then  $L$  is infinite language else not.

- **Infinite Language Problem:** Is  $L = L(G)$  an infinite language? ...

Let the grammar be:

$S \rightarrow AB, A \rightarrow aCb|a, B \rightarrow bB|bb$

$C \rightarrow cBS$



Since there is a loop in the dependency graph, the language is infinite. The derivation is  $S \Rightarrow^* (acbb)^i S(bbb)^i$ .



Chowdhary, K.R. (2025). Context-Free Grammars and Languages. In: Theory of Computation. Springer, Singapore.

[https://doi.org/10.1007/978-981-97-6234-7\\_6](https://doi.org/10.1007/978-981-97-6234-7_6)