

### CFG Definition

|                                                                   |                                                 |
|-------------------------------------------------------------------|-------------------------------------------------|
| Context-Free Grammar:                                             | $G = (V, T, S, P)$                              |
| V: Set of variables                                               | $\{S\}$                                         |
| T: Set of terminal symbols                                        | $\{a, b\}$                                      |
| S: Start variable                                                 | S                                               |
| P: Set of productions                                             | $\{S \rightarrow aSb, S \rightarrow \epsilon\}$ |
| ONLY ONE variable $\rightarrow$ String of variables and terminals |                                                 |

### Union of Two Languages

|                      |                                                                                                         |
|----------------------|---------------------------------------------------------------------------------------------------------|
| Example:             | $L = \{0^n 1^n   n \geq 0\} \cup \{1^n 0^n   n \geq 0\}$                                                |
| Break problem in two | $S^1 \rightarrow 0S^1 1   \epsilon$<br>$S^2 \rightarrow 1S^2 0   \epsilon$                              |
| Merge                | $S \rightarrow S^1   S^2$<br>$S^1 \rightarrow 0S^1 1   \epsilon$<br>$S^2 \rightarrow 1S^2 0   \epsilon$ |

### Simplifications of CFG

|                                                     |                                                 |                                                                            |                                                        |
|-----------------------------------------------------|-------------------------------------------------|----------------------------------------------------------------------------|--------------------------------------------------------|
| Substitution<br>( $B \rightarrow y^1$ )             | $A \rightarrow xBz$<br>$B \rightarrow y^1$      | $A \rightarrow xBz   xy^1z$<br>$A \rightarrow xBBz$<br>$B \rightarrow y^1$ | $A \rightarrow xBBz   xBy^1z$<br>$  xy^1Bz   xy^1y^1z$ |
| Removing $\epsilon$<br>( $B \rightarrow \epsilon$ ) | $A \rightarrow xBz$<br>$B \rightarrow \epsilon$ | $A \rightarrow xBz   xz$                                                   |                                                        |
| Unit<br>Production<br>( $A \rightarrow B$ )         | $A \rightarrow B$<br>$B \rightarrow bb$         | $A \rightarrow bb$<br>$B \rightarrow bb$                                   |                                                        |
| Useless<br>Productions                              | $A \rightarrow aA$<br>(infinite)                | $\therefore$ remove                                                        |                                                        |
|                                                     | Unreachable from S                              | $\therefore$ remove                                                        |                                                        |

Step 1: Remove Nullable Variables  
Step 2: Remove Unit-Production  
Step 3: Remove Useless Variables

### DFA to CFG

1. Create variable  $R^i$  for every state  $q^i$
2. Create rule  $R^i \rightarrow aR^j$  for every transition  $\delta(q^i, a) \rightarrow q^j$
3. For accept states  $q^i$  create rule  $R^i \rightarrow \epsilon$
4. For initial state  $q^0$  make  $R^0$  the start variable

### Conversion to Chomsky Normal Form

- Step 0: If start symbol (S) is on the right hand side, change start symbol  $S^0 \rightarrow S$
- Step 1: Remove Nullable variables ( $A \rightarrow \epsilon$ ) and Unit productions ( $A \rightarrow B$ )
- Step 2: For every symbol  $a$  add  $T^a \rightarrow a$
- Step 3: Replace  $A \rightarrow C^1 C^2 \dots C^n$  with  
 $A \rightarrow C^1 V^1$   
 $V^1 \rightarrow C^2 V^2$   
 $\dots$   
 $V^{n-2} \rightarrow C^{n-1} C^n$

Chomsky form only has productions in forms  
 $A \rightarrow BC$   
 $A \rightarrow a$

### Greibach Normal Form

All Productions have form:  $A \rightarrow aV^1 V^2 \dots V^k : k \geq 0$

Example

|                      |                            |
|----------------------|----------------------------|
| $S \rightarrow abSb$ | $S \rightarrow aT^b S T^b$ |
| $S \rightarrow aa$   | $S \rightarrow aT^a$       |
|                      | $T^a \rightarrow a$        |
|                      | $T^b \rightarrow b$        |

### PDA

Transitions:  $a, b \rightarrow c$  means when input is a, remove  $b$  from stack and add  $c$

If the automaton attempts to pop from empty stack then it halts and rejects input.

A string is accepted if there is a computation such that:  
 All the input is consumed  
 The last state is an accepting state

### PDA Formalities

|                                 |                                              |
|---------------------------------|----------------------------------------------|
| PDA Representation              | $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ |
| Q: States                       | $\{q_0, q_1, q_2\}$                          |
| $\Sigma$ : Input Alphabet       | $\{a, b\}$                                   |
| $\Gamma$ : Stack Alphabet       | $\{a, b, \$\}$                               |
| $\delta$ : Transition Functions | $\delta(q, a, w1) = \{(-q2, w2)\}$           |
| $q_0$ : Initial State           | $q_0$                                        |
| $z$ : Stack Start Symbol        | $\$$                                         |
| F: Accept States                | $\{q_2\}$                                    |

### CFG to PDA

Start with PDA of  $q_0 \xrightarrow{\epsilon, \epsilon \rightarrow S} q_1 \xrightarrow{\epsilon, \$ \rightarrow \$} q_2$

For each CFG production  $A \rightarrow w$  add  $\epsilon, A \rightarrow w$

For each CFG terminal  $a$  add  $a, a \rightarrow \epsilon$

### "Easy" PDA to CFG

For the pair of transitions:  
 $(p \xrightarrow{a, \epsilon \rightarrow t} r) (s \xrightarrow{b, t \rightarrow \epsilon} q)$

Add the production:  $A^{pq} \rightarrow aA^{rs}b$

For each state  $p$  add:  $A^{pp} \rightarrow \epsilon$

For each state-triple  $(p, q, r)$  add:  $A^{pr} \rightarrow A^{pq}A^{qr}$

For initial state and accept state:  
 $\rightarrow \textcircled{0} \ \& \ \square$

Add the production:  $S \rightarrow A^{0a}$

### Easy PDAs:

- Have only 1 accept state
- When accepting a string, the stack is empty (only initial symbol)
- Each transition pushes or pops

### PDA to "Easy" PDA

1. The PDA has a single accept state
- Create new accept state and make  $\epsilon, \epsilon \rightarrow \epsilon$  transitions from old accept states to the new accept state

### PDA to "Easy" PDA (cont)

2. Use new initial stack symbol #      New initial state, that transitions to a new state with  $\epsilon, \epsilon \rightarrow @$  (auxiliary symbol) that transitions to the old initial state with  $\epsilon, \epsilon \rightarrow \$$

3. On acceptance the stack contains only stack symbol #      Old accept state transitions to new to new accept state with  $\epsilon, @ \rightarrow \epsilon$ ,  $\alpha \delta$  self loops with  $\epsilon, x \rightarrow \epsilon$  where  $\forall x \in \Gamma - \{ @, \$ \}$

4. Transitions can't push and pop      Replace any  $q \rightarrow \sigma, a \rightarrow b \rightarrow q$  with  $q \rightarrow \sigma, a \rightarrow \epsilon \rightarrow r \rightarrow \epsilon, \epsilon \rightarrow b \rightarrow q$

5. 4. Transitions can't neither push nor pop      Replace any  $q \rightarrow \sigma, \epsilon \rightarrow \epsilon \rightarrow q$  with  $q \rightarrow \sigma, \epsilon \rightarrow \hat{\epsilon} \rightarrow r \rightarrow \epsilon, \hat{\epsilon} \rightarrow \epsilon \rightarrow q$



By Vipera  
[cheatography.com/vipera/](https://cheatography.com/vipera/)

Published 12th December, 2020.  
Last updated 12th December, 2020.  
Page 2 of 2.

Sponsored by [CrosswordCheats.com](https://CrosswordCheats.com)  
Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>