

DFA \leftrightarrow NFA \leftrightarrow regular expression \leftrightarrow pattern

regular languages

Pushdown automata

2DFA

Context-free grammar

(CFG)

↑ accepted

Context-free language

(CFL)

$$G = (N, \Sigma, P, S)$$

where

- * N is a finite set of NON terminal symbols
- * Σ is a finite set of terminal symbols disjoint from N
- * P is a finite subset of $N \times (N \cup \Sigma)^*$
- * $S \in N$ is the start symbol.

$$abcabc \in \Sigma^*, a, b, c \in \Sigma$$

1

$$S \rightarrow a \mid Ba$$

$$B \rightarrow Ba \mid a \mid \epsilon$$

$$(S) \rightarrow (a)$$

$$\hookrightarrow Ba \rightarrow Baa \rightarrow Baaa \rightarrow \epsilon aaa \downarrow aaa \rightarrow aaaa$$

Consider $\alpha, \beta \in (N \cup \Sigma)^*$, we say that

β is derivable from α in one step and write

$$\alpha \xrightarrow[G]{1} \beta$$

if β can be obtained from α by replacing some occurrence of a nonterminal A in α with γ , where $A \rightarrow \gamma$ is in P ,

$$\text{if } \exists \alpha_1, \alpha_2 \in (N \cup \Sigma)^*$$

$$A \rightarrow \gamma \in P$$

$$\text{st. } \alpha = \alpha_1 A \alpha_2$$

$$\beta = \alpha_1 \gamma \alpha_2$$

$$S, B \in N$$

$$S \rightarrow B$$

$$B \rightarrow a \mid Ba$$

$$S \rightarrow B \rightarrow Ba \rightarrow Baa \rightarrow aaaa$$

$$abcacbbba \in \Sigma^*$$

$$a, b, c \in \Sigma$$

$$P = \{ S \rightarrow A \mid B \mid C \}$$

not the final string of the string

capital symbols $\in N$

your final string cannot have any of these symbols

Terminal Symbols
 1) Denoted by small letters
 2) Can appear in any step of derivation process

3) Once it appears, it cannot change into anything else.

Non terminal Symbols
 Denoted by capital letters

Not allowed in the final step of derivation

Changes into terminal, nonterminal or a combination of both symbols according to production rules.

$S \rightarrow A \mid \text{ABC}$
 $A \rightarrow a \mid aA \mid \text{ABC}$
 $B \rightarrow b \mid bB$
 $C \rightarrow c \mid cC$

$a^n b^n \rightarrow$ Try to design a CFG to accept this

$S \rightarrow A \rightarrow aA \rightarrow aaA \rightarrow aaaa$
 $\hookrightarrow A \rightarrow ABC \rightarrow aABC$
 $\rightarrow aAbBC$

$aAbBC \rightarrow aabbc$
 $\text{ABC} \rightarrow \text{ABCBC}$
 $aABBCbBC$

$a^n b^n$

$S \rightarrow aSb \mid \epsilon$

palindromes on $\Sigma = \{a, b, c\}$

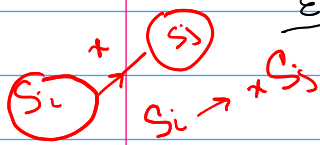
$S \rightarrow aSa \mid bSb \mid cSc \mid a \mid b \mid c \mid \epsilon$

legitimate bracketing

$S \rightarrow SS \mid (S) \mid \epsilon$

000
0000

binary strings divisible by 3.



$$S = S_0$$

$$S_0 \rightarrow 1S_1 \mid 0S_0 \mid \epsilon$$

$$S_1 \rightarrow 1S_0 \mid 0S_2$$

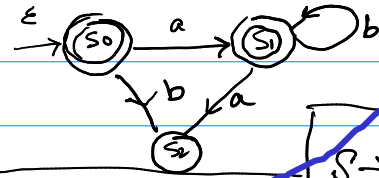
$$S_2 \rightarrow 1S_2 \mid 0S_1$$

10101100

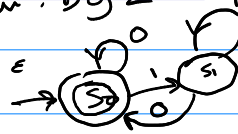
instead of the header moving from left to right and "exhausting" the string, assume that it generates the string as it moves from left to right

CFGs are strictly more powerful than DFAs

All strings with one a followed by b's
 $\epsilon, a, ab, abb, \dots$



div. by 2



$$S_0 \rightarrow \epsilon \mid 0S_0 \mid 1S_1$$

$$S_1 \rightarrow 0S_0 \mid 1S_1$$