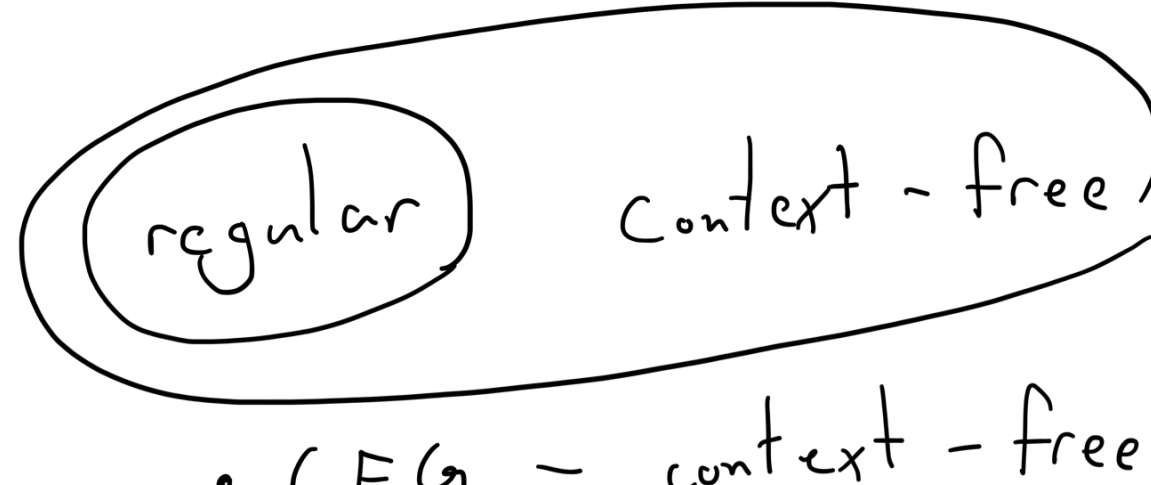
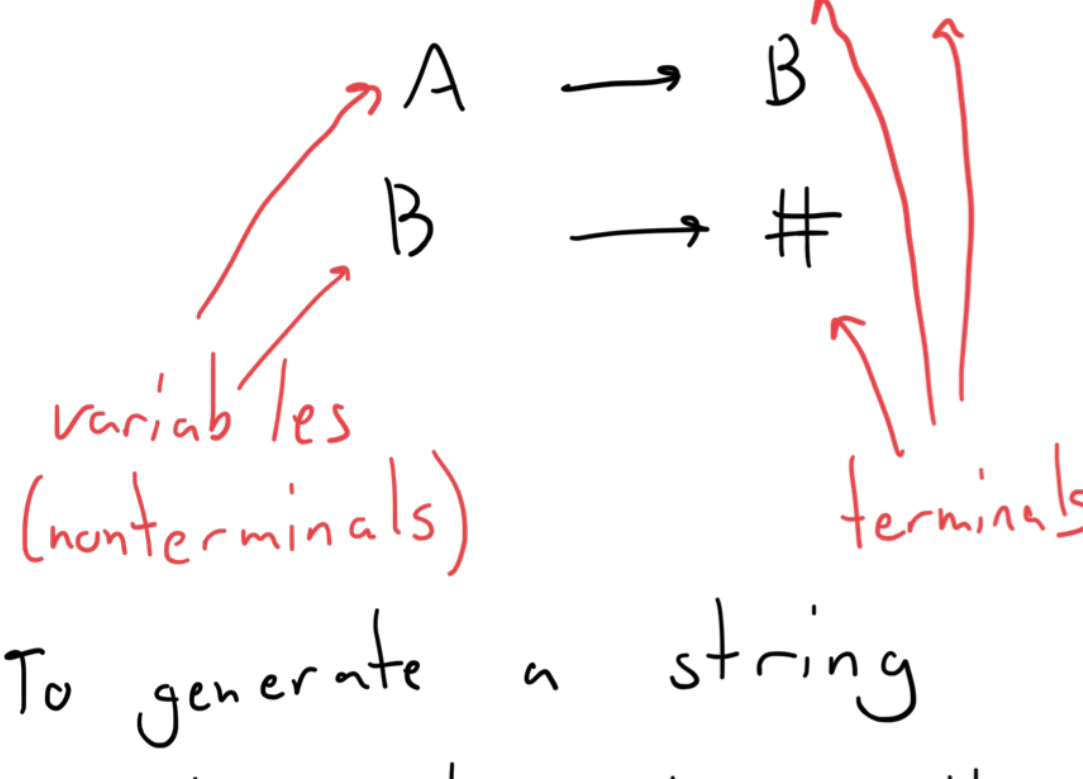


context-free languages & grammars



CFG - context-free grammar

Grammar G_1 :



G_1 is a collection of substitution rules

$000\#111 \in L(G_1)$

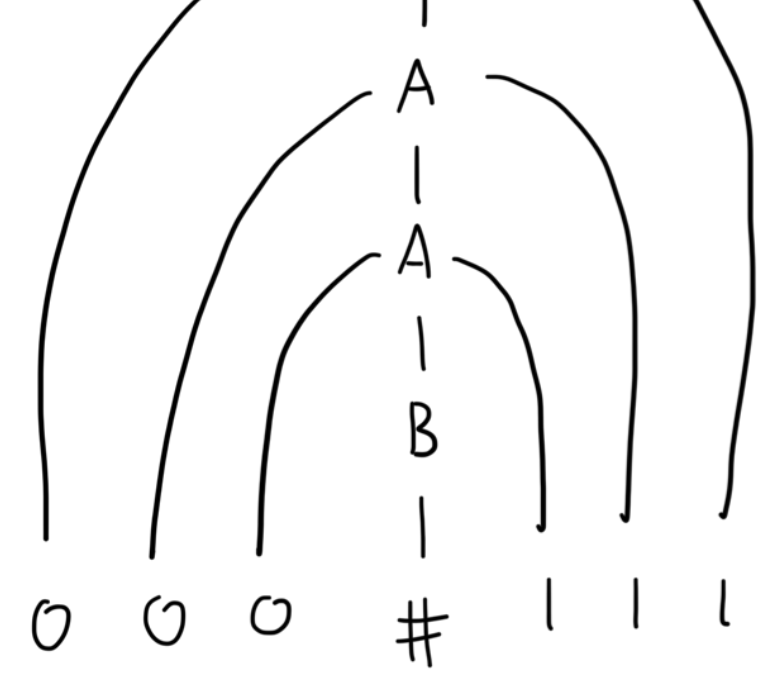
$A \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 000A111 \Rightarrow 000\#111$

To generate a string

1. Write down the start variable (the upper-left most var)
2. Replace a variable with a production.
3. Repeat step 2 until only terminals remain

$A^* \Rightarrow 000\#111$
↑ yields / derives

Parse tree for $000\#111$



Formal Definition A CFG is a 4-tuple

- (V, Σ, R, S) where
1. V is the finite set of variables
 2. Σ is the alphabet (finite set of terminals s.t. $\Sigma \cap V = \emptyset$)
 3. R is the set of rules, each rule being a variable and a string of variables and terminals.
 4. $S \in V$ is the start variable.

Ex 1

$L_1 = (0 \cup 1)^*$

CFG:

$S \rightarrow S0 \mid S1 \mid \epsilon$

$\begin{pmatrix} S \rightarrow S0 \\ S \rightarrow S1 \\ S \rightarrow \epsilon \end{pmatrix}$

or
shorthand notation

$S \Rightarrow S1 \Rightarrow S01 \Rightarrow S001 \Rightarrow \epsilon 001 \Rightarrow 001$

Ex 2

$A = \{0^n 1^n \mid n \geq 0\} \cup \{1^n 0^n \mid n \geq 0\}$

$S \rightarrow S_1 \mid S_2$

$S_1 \rightarrow 0S_11 \mid \epsilon$

$S_2 \rightarrow 1S_20 \mid \epsilon$

Ex 3

$B = \{0^n 1^m \mid n, m \geq 0\}$

$= 0^* 1^*$

$S \rightarrow 0S \mid P$

$P \rightarrow 1P \mid \epsilon$

Operations on CFGs

Let S_1 & S_2 be the start variables of CFGs L_1 & L_2 .

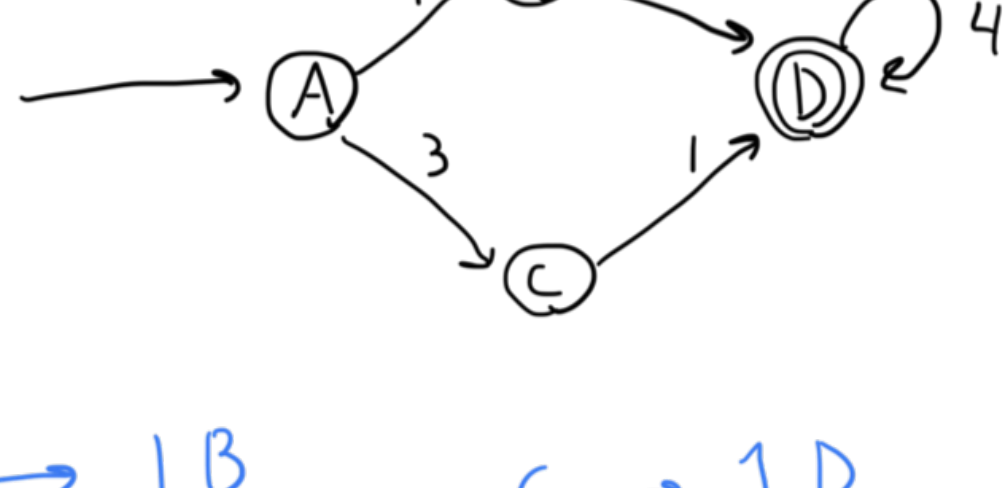
$L_1 \cup L_2 \Rightarrow S \rightarrow S_1 \mid S_2$

$L_1 \circ L_2 \Rightarrow S \rightarrow S_1 S_2$

$L_1^* \Rightarrow S \rightarrow \epsilon \mid S_1 S$

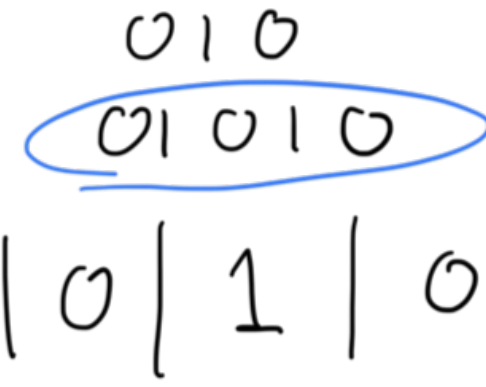
Theorem: Every regular language is context-free

- Make a variable for each state.
- Make a rule for each edge
- Add an epsilon rule for each accept state.



$A \rightarrow 1B$
 $A \rightarrow 3C$
 $B \rightarrow 2B$
 $B \rightarrow 3D$
 $C \rightarrow 1D$
 $D \rightarrow 4D$
 $D \rightarrow \epsilon$

$P = \{ \text{palindromes over } \{0,1\}^* \}$



$S \rightarrow \epsilon \mid 0 \mid 1 \mid 0S0 \mid 1S1$

$S \Rightarrow 0S0 \Rightarrow 01S10 \Rightarrow 01010$
↑ palindrome ↑ palindrome ↑ palindrome ↑ palindrome

$W = \{ w \mid w \in \{0,1\}^* \text{ and the number of 0's equals the number of 1's} \}$

$S \rightarrow 0A \mid 1B \mid \epsilon$

$A \rightarrow 1S \mid 0AA$

$B \rightarrow 0S \mid 1BB$

yield strings with equal 0's & 1's

yields strings with one more 1 than 0's

yields strings with one more 0 than 1's

$S \Rightarrow 0A \Rightarrow 00AA \Rightarrow 000AAA \Rightarrow 0001SAA \Rightarrow 00010AAA \Rightarrow \dots \Rightarrow 000101S1S1S \Rightarrow \dots \Rightarrow 00010111$

an equivalent grammar

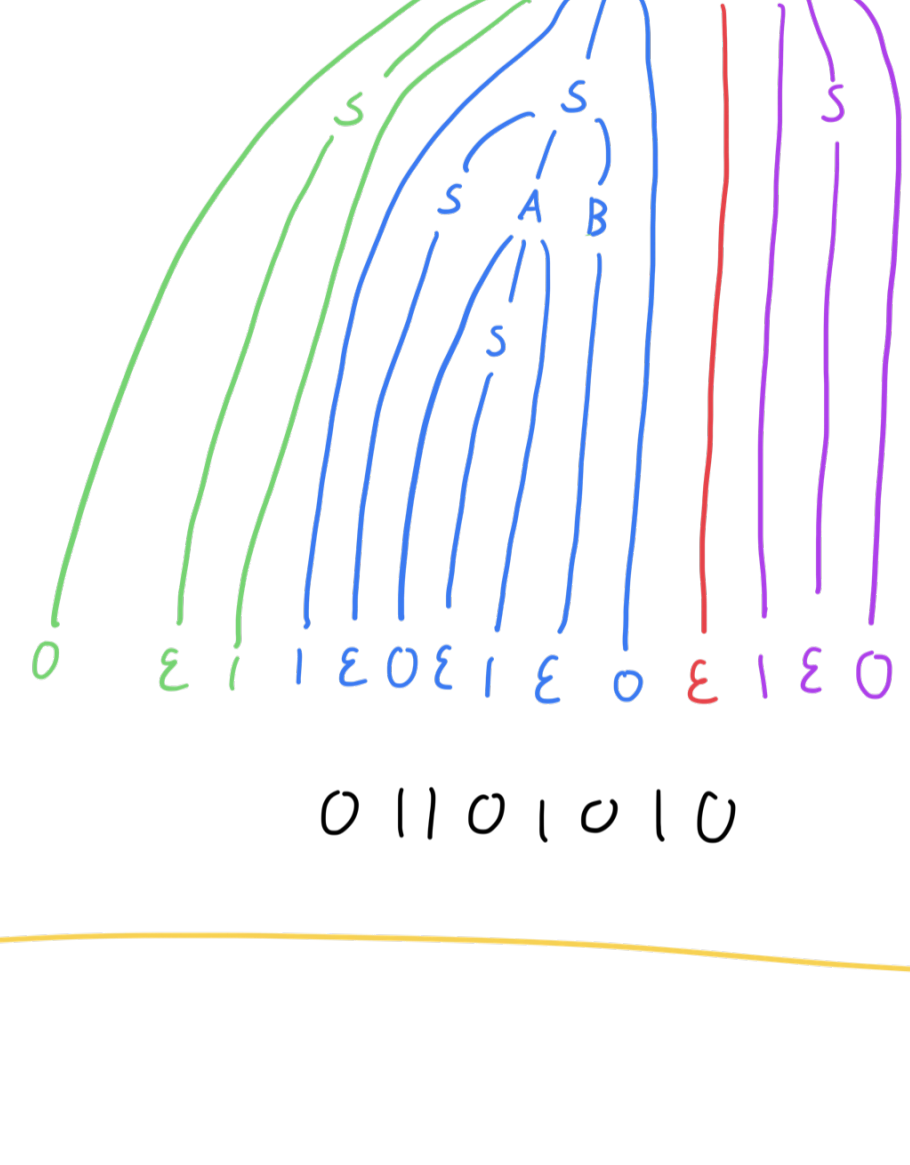
$S := SAB \mid \epsilon$

$A := 0S1 \mid \epsilon$

$B := 1S0 \mid \epsilon$

No substitution that results in more 0's than 1's or vice versa is possible

$(AB)^* = ABABAB \dots$
Every AB can go to ϵ or any string of A's & B's.
For example:



$E \rightarrow E + E \mid E \times E \mid (E) \mid a$

ambiguous
• 2 or more parse trees exist for some input

