Decidability Property of Regular Languages:

↓

Algo ?

- Emptiness problem is decidable

↳ FA is not accepting any string $L = \{ \_, \_, \_ \}$

$L = \Phi$

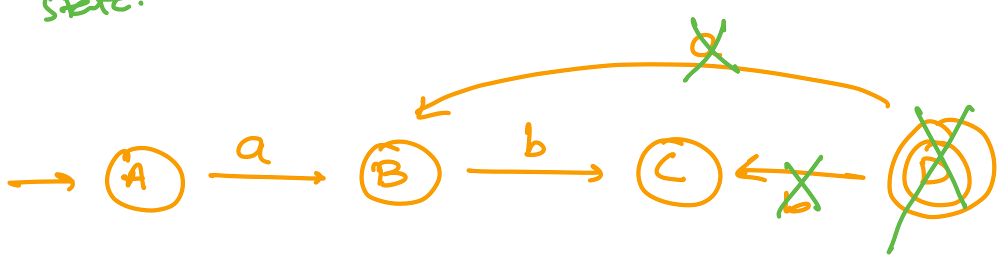
when can u say that FA will accept at least 1 string?

Algo:

1. Select all states which are not reachable from IS.

   Delete all unreachable state and also delete the transitions corresponding to them.

2. In the remaining FA, see if there is at least 1 final state.



$L = \Phi$?

atleast 1 final state $L \neq \Phi \rightarrow$

no final state $L = \Phi \rightarrow$

} Emptiness is decidable

- Infiniteness problem is Decidable

$\Sigma = \{a, b\}$

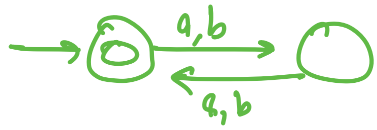
$L_1 =$ length of strings should be 2 $= \{aa, ab, ba, bb\}$

$L_2 =$ atleast 2 $= \{aa, ab, ba, bb \cdots\cdots\}$

Q: FA given, you need to tell finite language create or infinite language create ?

At least 2 length:

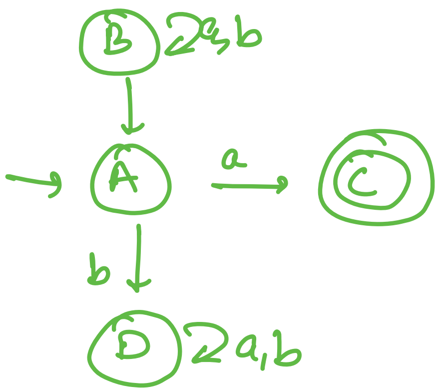
Even length strings:



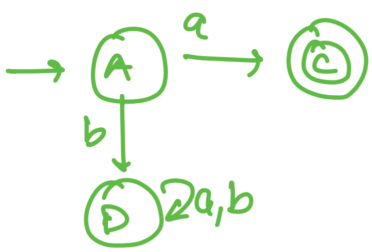
If language is ∞ then definitely there will be a loop

→ reachable from IS ✓
from loop reach to FS ✓

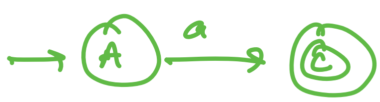
Eg:



Step 1: Remove all states which are unreachable from IS and also transitions corresponding to them.



Step 2: Delete all states & transitions from which u can't reach to final state.



Step 3: In remaining FA, atleast 1 loop → True: Infinite

→ false: Finite

Equality Problem is Decidable ?

$FA_1 = FA_2$?

If strings accepted by 2 FA are same then FA are said to be equal.

$L_1 = L_2$?

strings in this language L_1 = strings in language L_2.

Q: L_1, L_2

Is there any algo which can tell if $L_1 = L_2$ OR $L_1 \neq L_2$?

RL:	L_1	L_2
DFA:	DFA_1	DFA_2
minimized DFA:	DFA m_1	DFA m_2

Equal ?

Yes → $L_1 = L_2$

No → $L_1 \neq L_2$

Membership problem is Decidable

string w , language L

$w \in L$?

$L = \{\epsilon, aa, ab, ba, bb, aabb \dots\}$

$w =$ "aba" ?

RL $\quad L \longrightarrow$ FA

FA provide w, you end final state $\longrightarrow$ Accepted Present in language $w \in L$

false $\downarrow$

$w \notin L$

Regular Expression

$\uparrow$ Represent

Regular Languages

Generator $\swarrow$ $\qquad$ $\searrow$ Acceptor

Regular Grammar $\qquad$ FA

Grammar:

Grammar is to generate the entire language

Grammar is represented as:

(V, T, P, S) — $S \to$ Start symbol, $P \to$ Productions, $V \to$ variables, $T \to$ Terminals

Eg: Start Symbol

Productions P

$$S \to aSb$$
$$S \to aB$$
$$B \to b$$

$V = \{S, B\}$

$T = \{a, b\}$

$a^n b^n \mid n \geq 1$

$S \rightarrow aSb$

$S \rightarrow aaSbb$

$S \rightarrow aaaSbbb$

$S \rightarrow aaaaBbbb$

$S \rightarrow aaaabbbb$

a^4b^4

Sentential form / Sequential Form

$S \rightarrow aSb$

$S \rightarrow aaBb$

$S \rightarrow aabb$

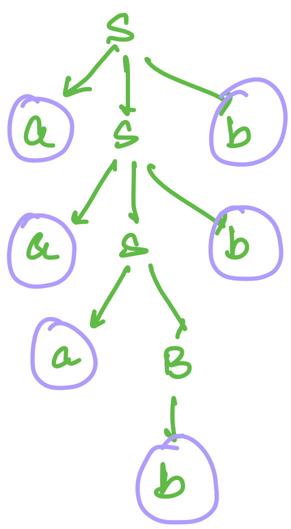
a^2b^2

$S \rightarrow aB$

$\rightarrow ab$

a^1b^1

Derivation Tree / Parse Tree



a^3b^3

Multiple Grammar:

$S \rightarrow aSb$

$S \rightarrow ab$

$S \rightarrow aSb \mid ab$

Q: Grammar which generates strings of length 2.

$\Sigma = \{a, b\}$

<u>Way 1:</u>

$L = \{aa, ab, ba, bb\}$

$S \rightarrow aa \mid ab \mid ba \mid bb$

Way 2:

RE: $\underbrace{(a+b)}_{A}\ \underbrace{(a+b)}_{A}$

$S \rightarrow AA$

$A \rightarrow a \mid b$

Q: $a^n \mid n \geq 0$

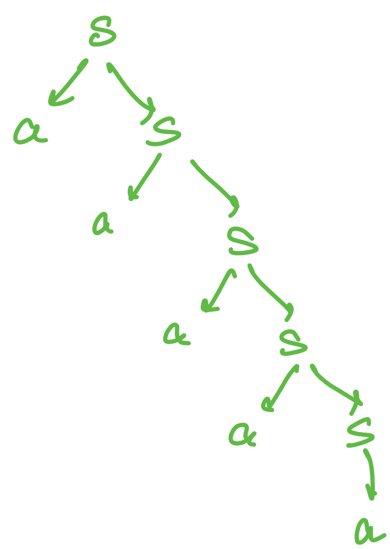
$L = \{a^0, a^1, a^2, a^3 \ldots\}$

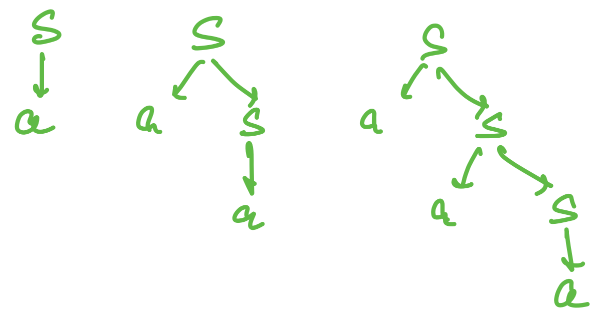
$L = \{\varepsilon, a, aa, aaa \ldots\}$

RE: a^*

a^*

$S \rightarrow aS \mid a$



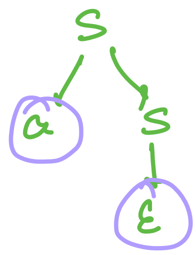


$S \rightarrow aS \mid \varepsilon$

ε:

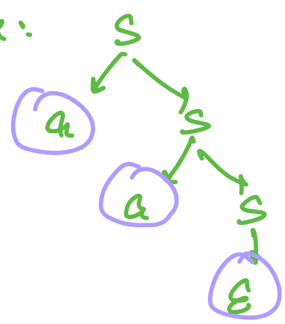
ε

a:



$a \cdot \varepsilon = a$

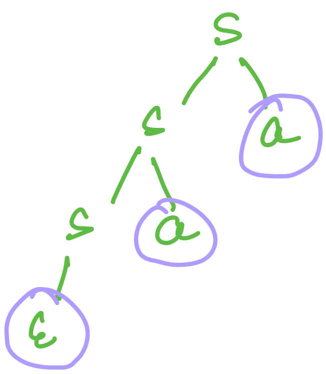
aa:



$aa\varepsilon = aa$

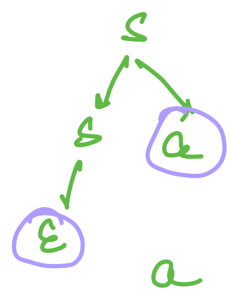
OR

$S \to Sa \mid \varepsilon$



$\varepsilon aa = aa$

ε



a

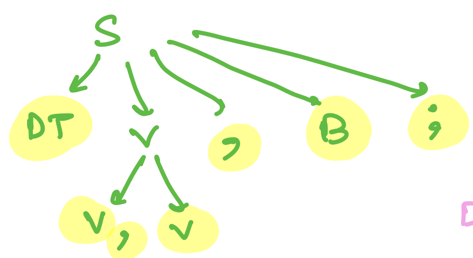
```
int main ()
{
    int a,b;
    a=10;
    b=20;
    :
}
```

$DT \to int \mid float \mid double$

$S \to DT \quad V \ , \ B \ ;$

$V \to V , V$

$B \to C$



DT V, V, B ;

Q: $(a+b)^*$ ⟶ set of all strings over a, b

Length 0 : ε

Length 1 : $(a+b)$

Length 2 : $(a+b)\ (a+b)$

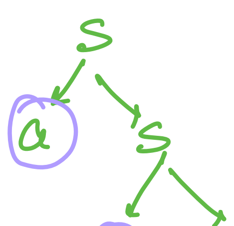
Length 3 : $(a+b)\ (a+b)\ (a+b)$

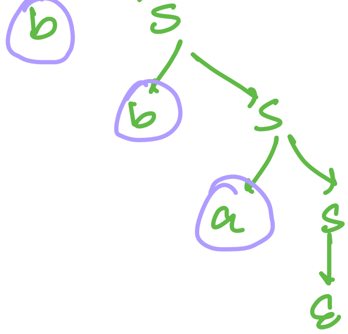
$S \to aS \mid bS \mid \varepsilon$

$(a+b)^+$

$S \to aS \mid bS \mid a \mid b$

abba





Q: Strings of length atleast 2

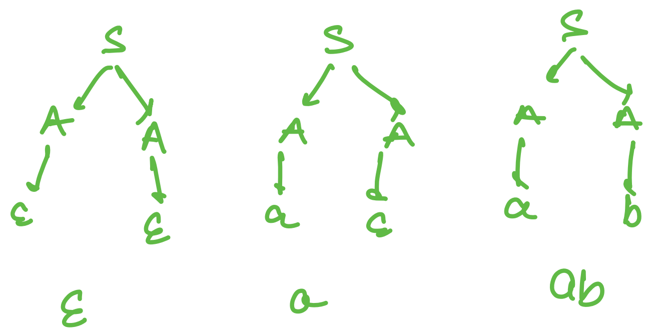
RE: $\underbrace{(a+b)}_{A}\underbrace{(a+b)}_{A}\underbrace{(a+b)^*}_{B}$

$S \to AAB$
$A \to a \mid b$
$B \to aB \mid bB \mid \varepsilon$

Q: length atmost 2

RE: $\underbrace{(a+b+\varepsilon)}_{A}\underbrace{(a+b+\varepsilon)}_{A}$

$S \to AA$
$A \to a \mid b \mid \varepsilon$



Q: starts & ends with different symbols.

RE: $a\underbrace{(a+b)^*}_{A}b + b\underbrace{(a+b)^*}_{A}a$

$S \to aAb \mid bAa$
$A \to aA \mid bA \mid \varepsilon$

Q: starts & ends with same symbols.

RE: $a\underbrace{(a+b)^*}_{A}a + b\underbrace{(a+b)^*}_{A}b + a + b$

$S \to aAa \mid bAb \mid a \mid b$

$A \to aA \mid bA \mid \varepsilon$

Q: $a^n b^n \mid n \geq 1$

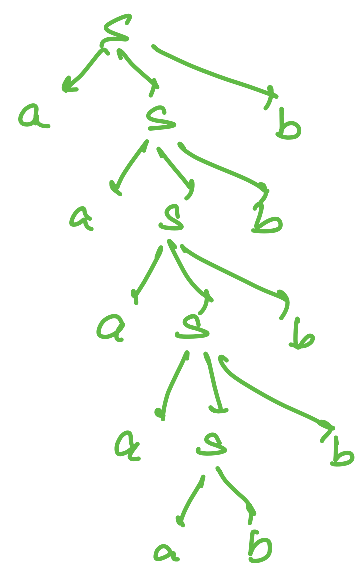
$\underline{\qquad\qquad}$ $\underline{\qquad\qquad}$

n a's n b's

$\underline{a}$ $\underline{a}$ $\underline{a}$ $\underline{a}$ $\underline{a}\underline{b}$ $\underline{b}$ $\underline{b}$ $\underline{b}$ $\underline{b}$

$a^5 b^5$

$S \to aSb \mid ab$



Q: $a^n b^n \mid n \geq 0$

$S \to aSb \mid \varepsilon$

Q: Palindrome

nitin

$\Sigma = \{a, b\}$

even

odd

$w w^R$

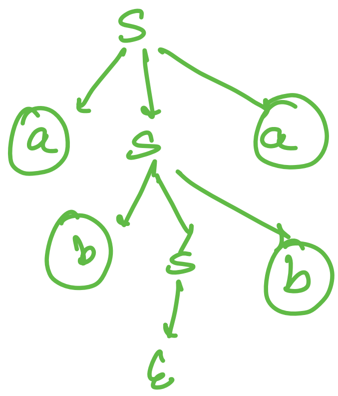
$w a w^R$ OR $w b w^R$

abb bba

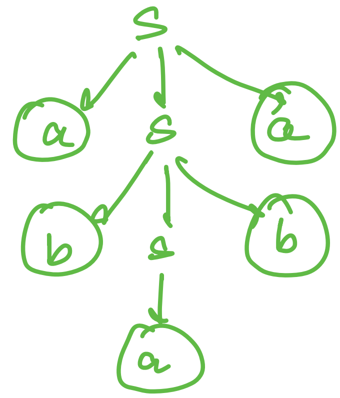
$S \rightarrow aSa \mid bSb \mid a \mid b$

$S \rightarrow aSa \mid bSb \mid \varepsilon$

abba



ab a ba



$S \rightarrow aSa \mid bSb \mid \varepsilon \mid a \mid b$

Q: Even length strings

$((a+b)(a+b))^*$ — each $(a+b)$ is A; $(a+b)(a+b)$ is B

$S \rightarrow BS \mid \varepsilon$

$B \rightarrow AA$

$A \rightarrow a \mid b$

a^*

$S \rightarrow aS \mid \varepsilon$

a^+

$S \rightarrow aS \mid a$

Q: $a^n b^m \mid n, m \geq 1$

RE: $a^+ b^+$: $\underbrace{aa^*}_{A} \underbrace{bb^*}_{B}$

$S \rightarrow AB$

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$

OR

$S \rightarrow aS \mid Sb \mid ab$

OR

$S \rightarrow aAbB$

$A \rightarrow aA \mid \varepsilon$

$B \rightarrow bB \mid \varepsilon$

Q: $\underbrace{a^n b^n}_{A} \underbrace{c^m}_{B} \mid n, m \geq 1$

$S \rightarrow AB$

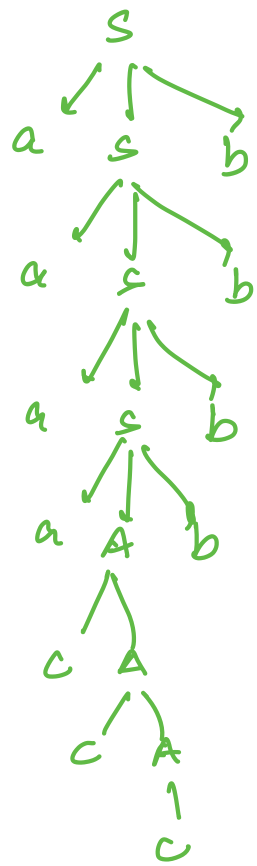
$A \to aAb \mid ab \longrightarrow a^n b^n$

$B \to cB \mid c \longrightarrow c^m$

Q: $a^n c^m b^n \mid n, m \geq 1$

$S \to aSb \mid aAb$

$A \to cA \mid c$

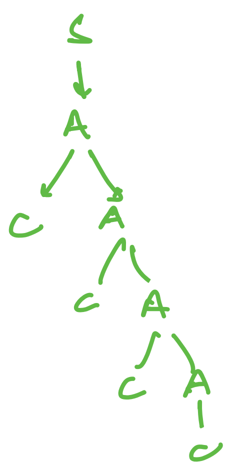


$a^4 c^3 b^4$

$S \to aSb \mid A$

$A \to cA \mid c$

$a^n c^m b^n \mid n \geq 0, m \geq 1$



$\underline{\underline{c^4}}$

$a^n c^m b^n \mid n, m \geq 0$

$S \to aSb \mid A$

$A \to cA \mid \varepsilon$

Q: $\underbrace{a^n b^n}_{A} \underbrace{c^m d^m}_{B} \mid n, m \geq 1$

$S \to AB$

$A \to aAb \mid ab$

$B \to cBd \mid cd$

Q: $a^n b^{2n} \mid n \geq 1$

$a^n (bb)^n$

$S \rightarrow aSbb \mid abb$

$a^n b^{2n} \mid n \geq 0$

$S \rightarrow aSbb \mid \varepsilon$

Q: $a^n \underbrace{b^m c^m}_{A} d^n \mid n, m \geq 1$

$S \rightarrow aSd \mid aAd$

$A \rightarrow bAc \mid bc$

Q: $a^{m+n} b^m c^n \mid n, m \geq 1$

$a^n \underbrace{a^m b^m}_{A} c^n$

$S \rightarrow aSc \mid aAc$

$A \rightarrow aAb \mid ab$

Q: $a^n b^m c^{n+m} \mid n, m \geq 1$

$a^n \underbrace{b^m c^m}_{A} c^n$

$S \rightarrow aSc \mid aAc$

$A \rightarrow bAc \mid bc$

Last year mid sem

Q:

$L = \{0^i 1^j 0^k \mid j > i+k\}$

$i \geq 0 \quad k \geq 0$

$j \geq 1$

$$\underbrace{0^i 1^i}_{A}\ \underbrace{1^+}_{B}\ \underbrace{1^k 0^k}_{C}$$

$S \rightarrow ABC$

$A \rightarrow 0A1 \mid \varepsilon$

$B \rightarrow 1B \mid 1$

$C \rightarrow 1C0 \mid \varepsilon$

Chomsky's Classification of Grammar:

1. Regular Grammar (Type 3)
2. Context free Grammar (Type 2)
3. Context sensitive Grammar (Type 1)
4. Recursively Enumerable Grammar (Type 0)

more relaxed ↓ | more restricted ↑

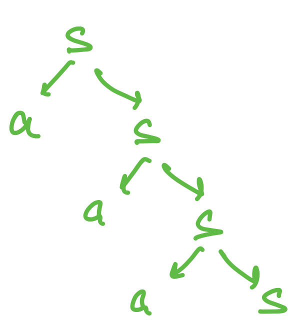
Type 3: Regular Grammar

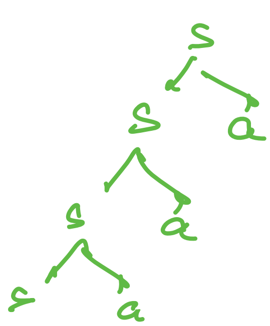
Grammar has all the production rules of the form:

$A \rightarrow \alpha B \mid \beta$	$A \rightarrow B\alpha \mid \beta$
$A, B \in V$	$A, B \in V$
$\alpha, \beta \in T^*$	$\alpha, \beta \in T^*$

$\underbrace{S}_{A} \rightarrow \underbrace{a}_{\alpha}\underbrace{S}_{B} \mid \underbrace{a}_{\beta}$ $\qquad$ $\underbrace{S}_{A} \rightarrow \underbrace{S}_{B}\underbrace{a}_{\alpha} \mid \underbrace{a}_{\beta}$

Right Linear Grammar





Left Linear Grammar