$G = \langle V, T, S, P \rangle$ Context-free grammar

V set of non-terminal symbols

T " " terminal symbols (TOKENS)

$S \in V$ initial symbol

P set of rules "productions"

Context-free rules: $A \to \gamma$

$A \in V \quad \gamma \in (V \cup T)^*$

$E \to E + E \mid E * E \mid E - E \mid E / E \mid (E) \mid \underline{id} \mid$

$\underline{num}$

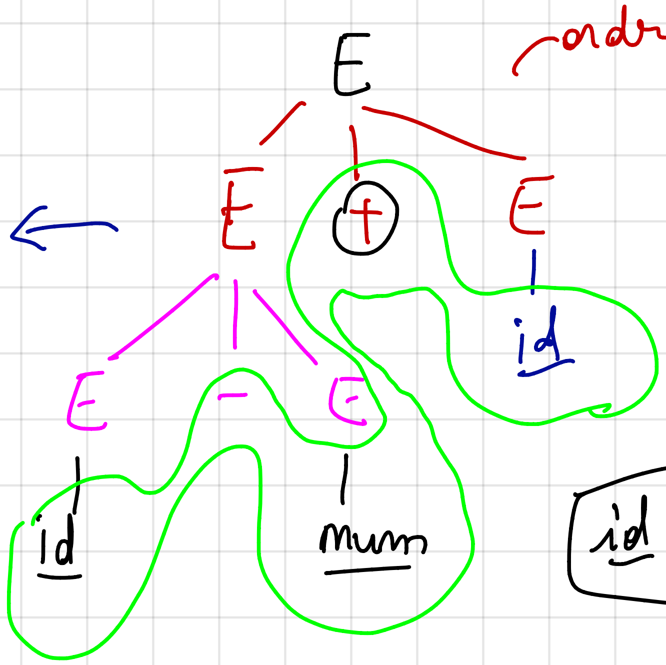
or $\to$ $E \to E + E$

$E \to E * E$

$V = \{E\}$ E initial

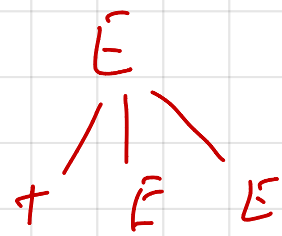
$T = \{\underline{id}, \underline{num}, +, -, *, /, (,)\};$

Parse Tree

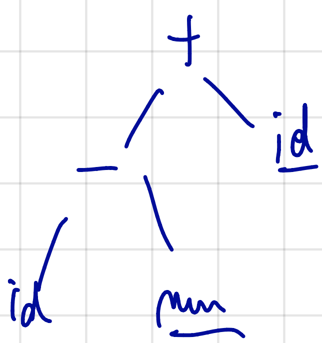


ordered tree

$\neq$



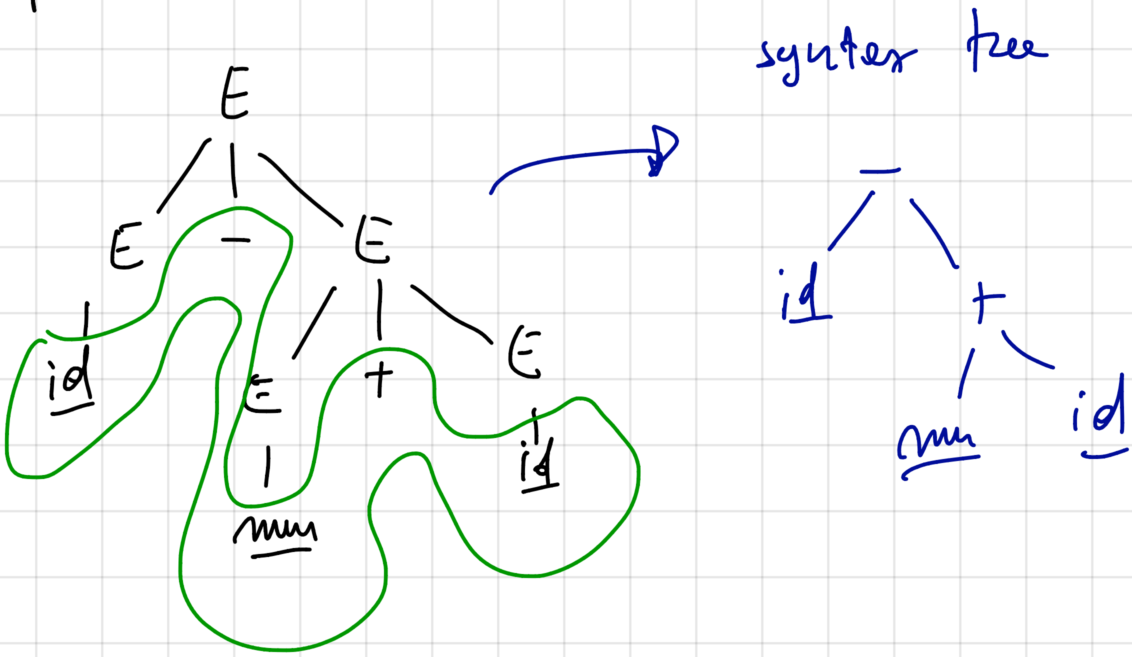
Syntax tree



$\boxed{\underline{id} - \underline{num} + \underline{id}}$

Parsing problem : given id - num + id as a stream of TOKENS

what is the parse tree?



There is at least a string for which the grammar can produce more than one parse-tree

$\Rightarrow$ THE grammar is called AMBIGUOUS

id

Let's change the grammar

$V = \{E, T, F\}$

initial

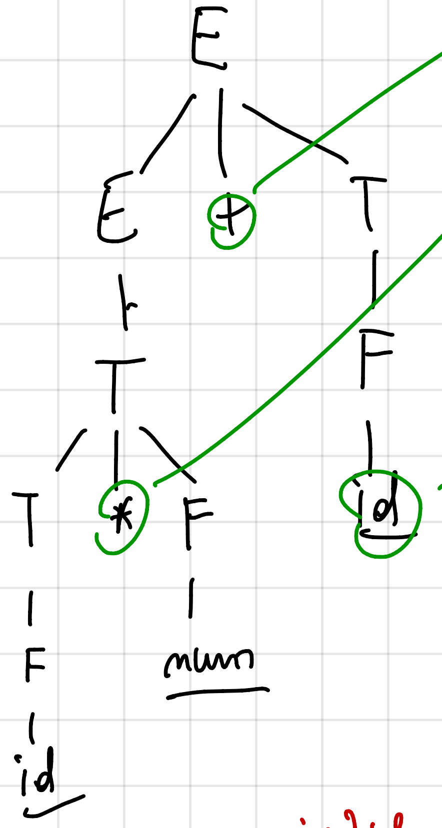
$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow (E) \mid id \mid num$

id * num + id

This grammar is not ambiguous



change gr

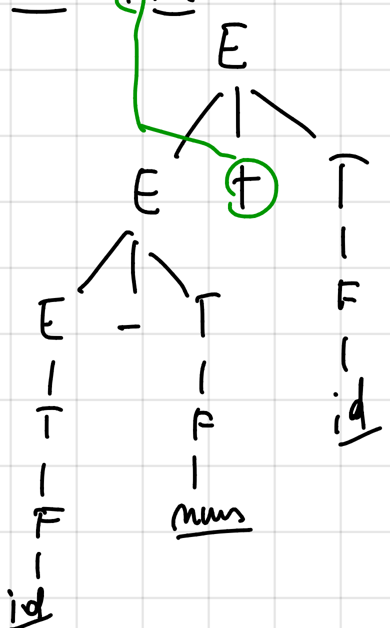
initial

$E \rightarrow E + T \mid E - T \mid T$

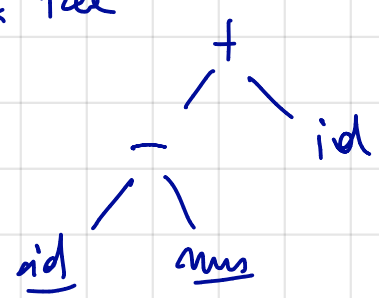
$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow (E) \mid id \mid num$

id - num + id



syntax tree



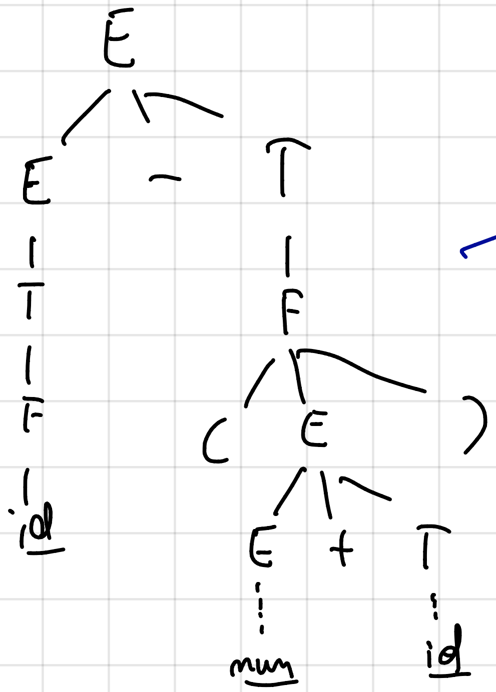
$\underline{id} - (\underline{num} + \underline{id})$

initial

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow (E) \mid \underline{id} \mid \underline{num}$



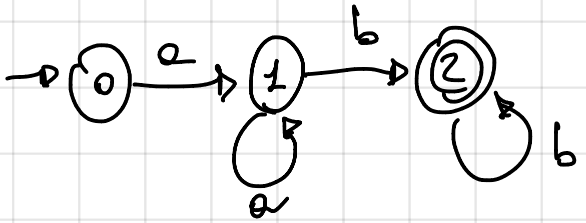
Syntax tree

EXPRESSIVE POWER of FSA

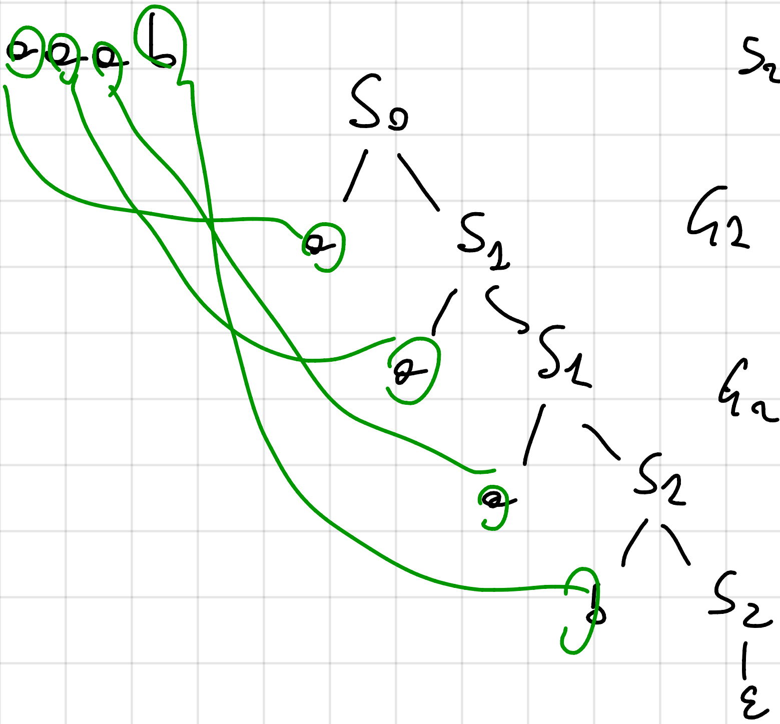
Regular grammars

$A \to \varepsilon$ $b \in T$
$A \to b$ $A, B \in V$
$A \to bB$

$\mathcal{L} = \{a^n b^m \mid n, m > 0\}$



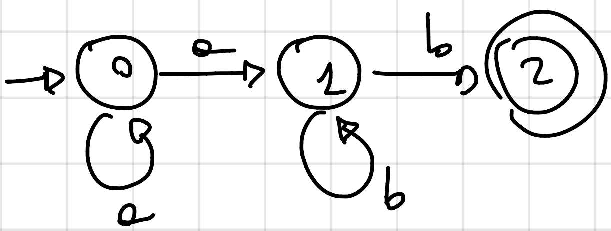
$S_0 \to a S_1$
$S_1 \to a S_1 \mid b S_2$
$S_2 \to \varepsilon$
$S_2 \to b S_2$



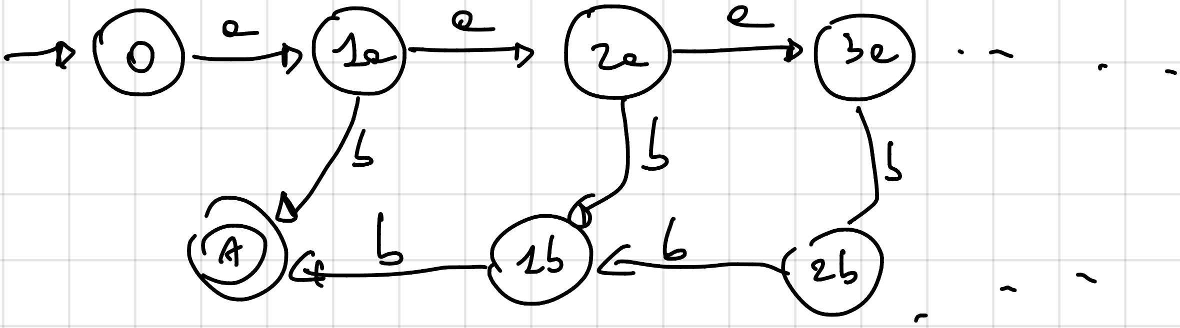
$G_1 \mid S \to aS \mid b$ unambiguous

$G_2 \left\{ \begin{array}{l} S \to a S_1 \mid aS \mid b \\ S_1 \to a S_1 \mid b \end{array} \right.$ ambiguous

$\mathcal{L}_1 = \{a^n b^n \mid n > 0\}$



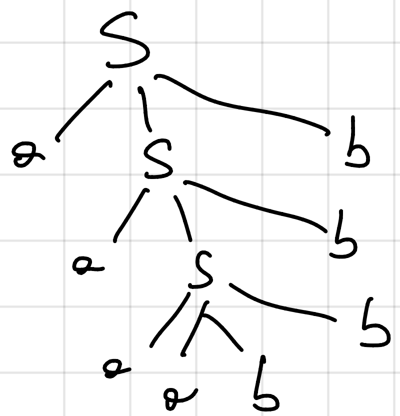
$\to \{a^n b^m \mid n, m > 0\}$



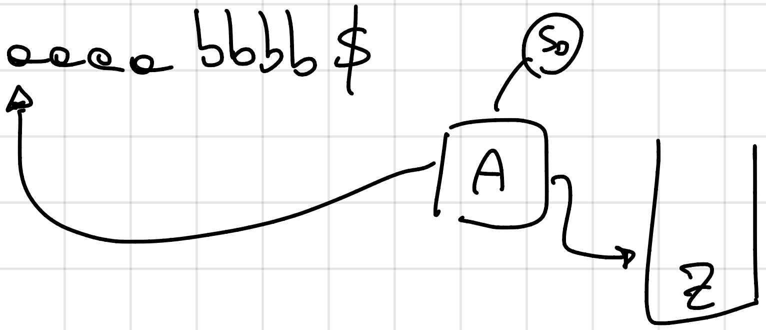
Context - free - grammar for $\mathcal{L}_1 = \{a^n b^n \mid n > 0\}$

$S \to aSb \mid ab$

aaaa bbbb



Pushdown automaton for $\mathcal{L}_1 = \{a^n b^n \mid n > 0\}$



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

$V = \{Z, A\}$

ACCEPT iff stack is empty and input is empty

S_0 initial state

input sym / stack symb

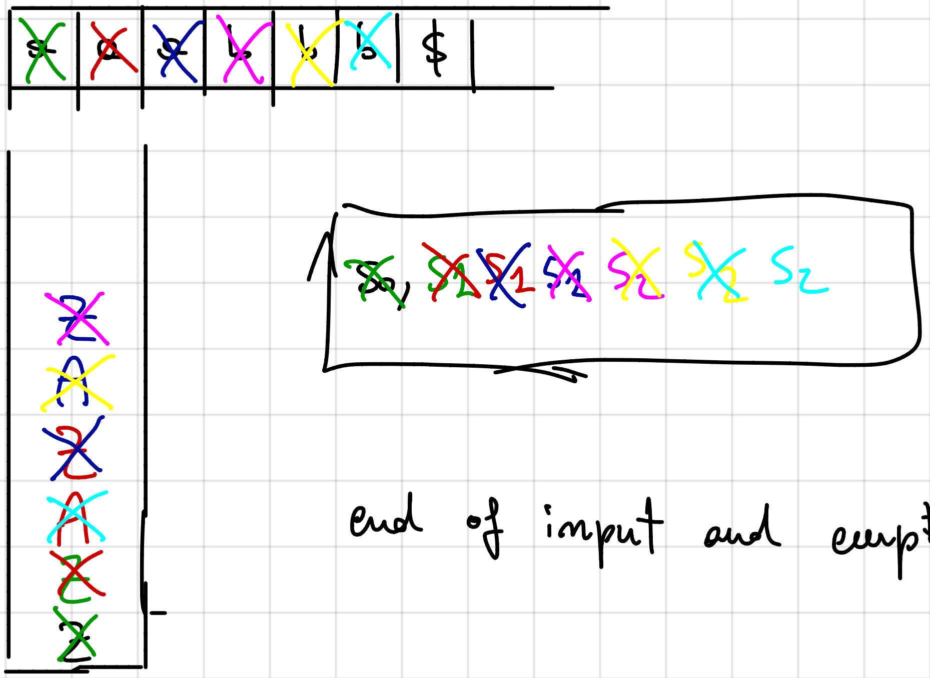
	a	b	\$
Z	S_1, Z	—	—
A	—	—	—

S_1

	a	b	\$
Z	S_2, AZ	S_2, ε	—
A	—	—	—

S_2

	a	b	\$
Z	—	—	—
A	—	S_2, ε	



end of input and empty stack → ACCEPT!

$L_2 = \{a^n b^m c^{2n} \mid n > 0,\ m \geq 0\}$ accept by empty stack / input

s_0

	a	b	c	\$
Z	s_1, AZ	–	–	–

s_1

	a	b	c	\$
Z	s_1, AAZ	s_2, Z	s_3, ε	–
A	–	–	–	–

s_2

	a	b	c	\$
Z	–	s_2, Z	s_3, ε	–
A	–	–	–	–

s_3

	a	b	c	\$
Z	–	–	–	–
A	–	–	s_3, ε	–

aabbbcccc $(s_0, aabbbcccc, Z) \to (s_1, abbbcccc, AZ)$

$\to (s_1, bbbcccc, AAAZ) \to (s_2, bbcccc, AAAZ) \to (s_2, bcccc, AAAZ)$

$\to (s_2, cccc, AAAZ) \to (s_3, ccc, AAA) \to (s_3, cc, AA)$

$\to (s_3, c, A) \to (s_3, \varepsilon, \varepsilon)$ ACCEPT