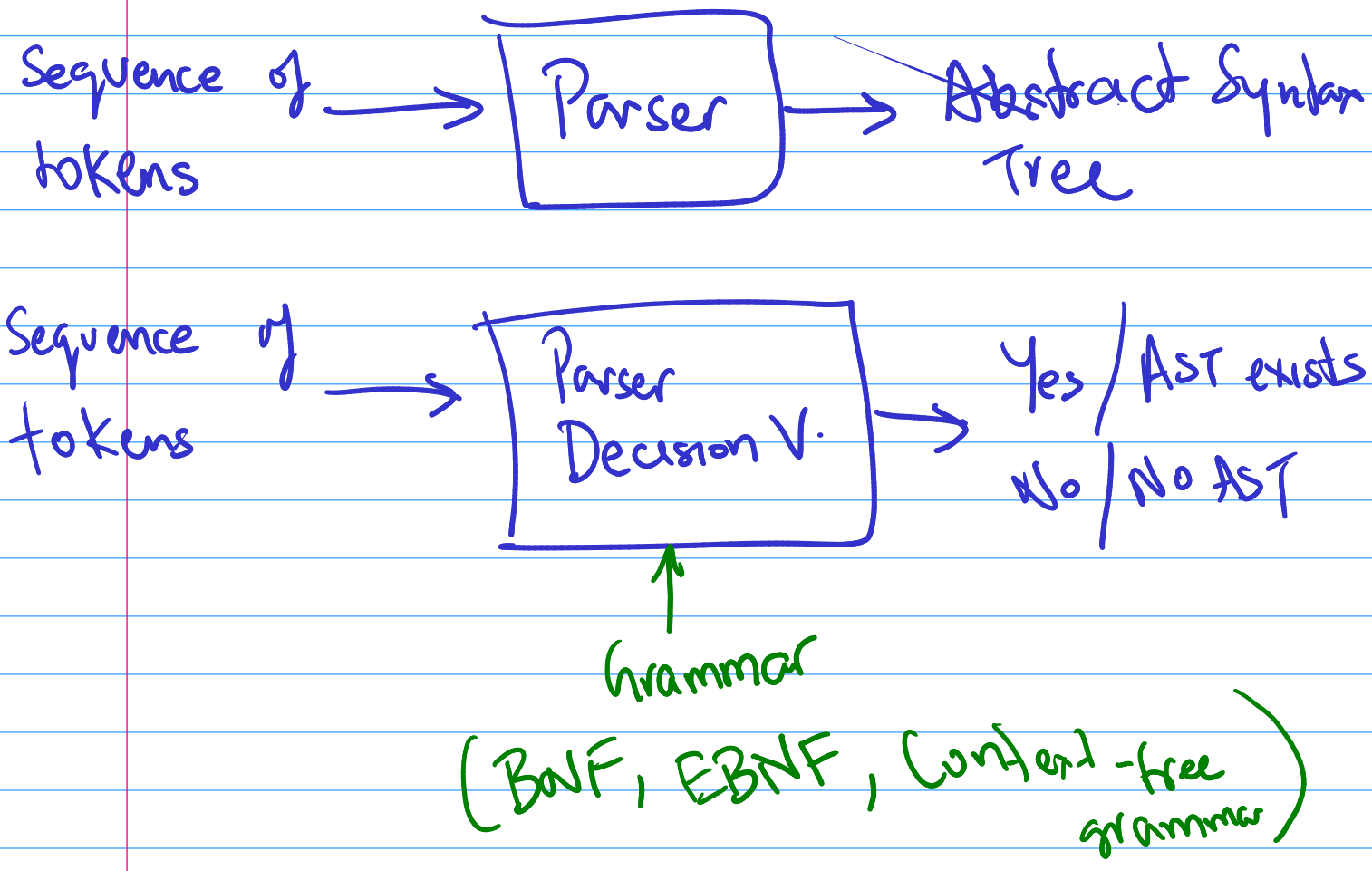
- Lexical Analyzers & Parser Generators

Algorithmics



Grammar: Terminal symbols | Nonterminal symbols | Rules

actual words
cat, dog, apple
sat, on, the,
...

noun, preposition
subject phrase,
object phrase, ...

conceptual categories

Expr ::= Expr '+' Expr

Non-terminal symbol

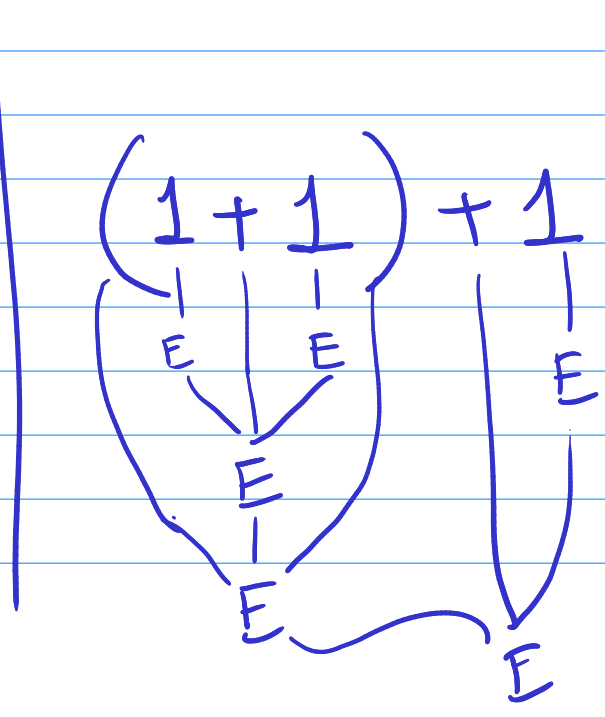
Sequence of terminal & non-terminal symbols

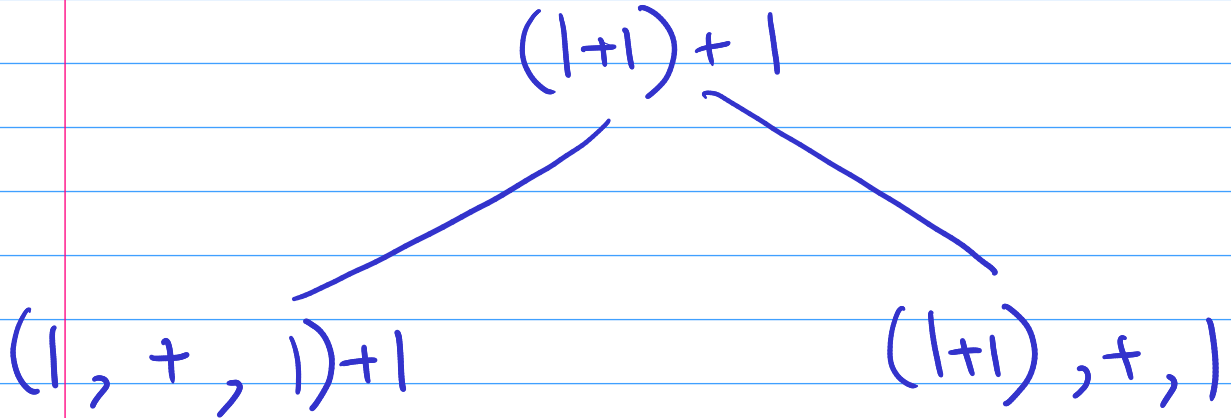
E ::= E + E

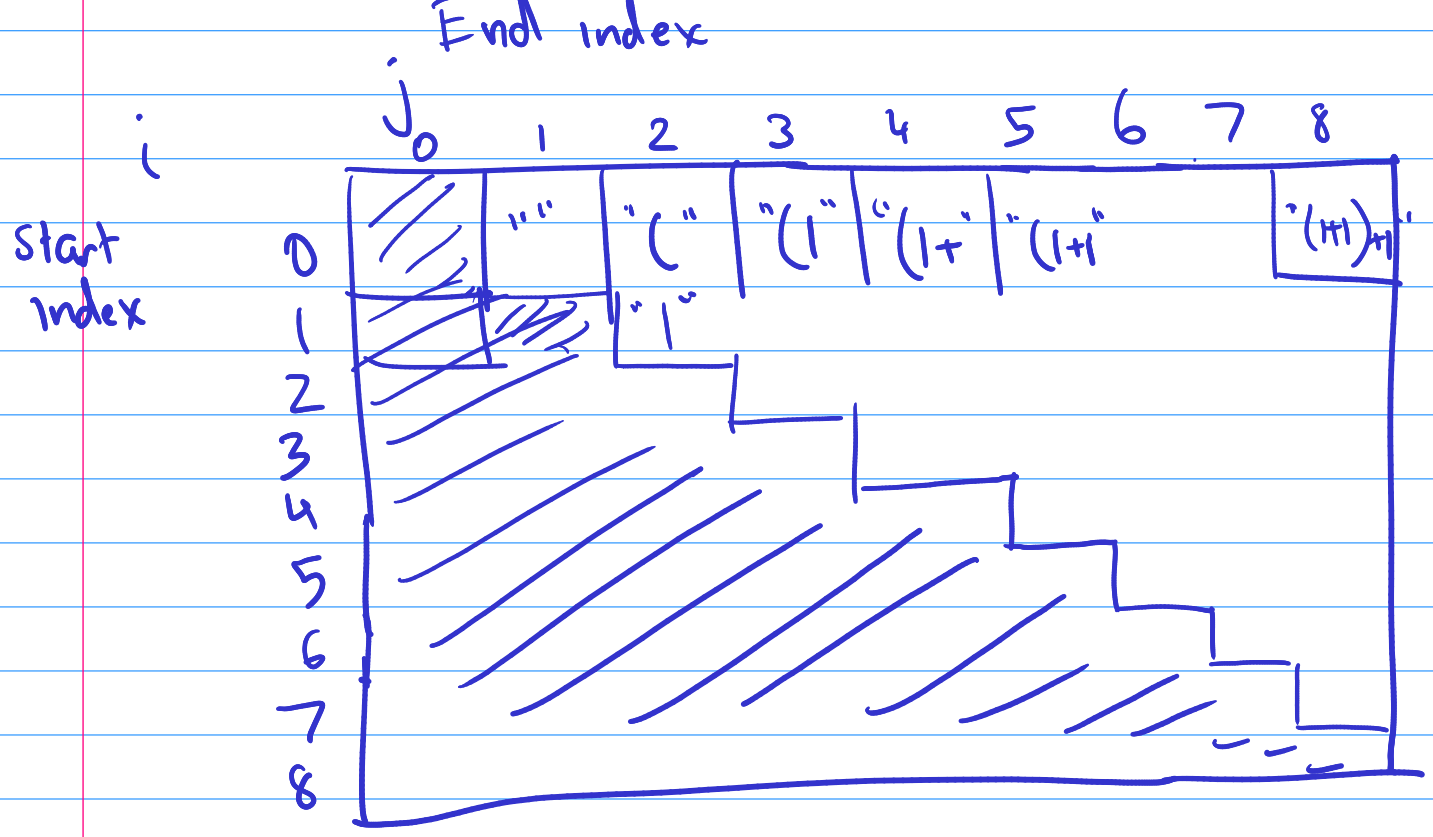
E ::= (E)

E ::= 1

E ::= 0







- For each entry in the table,
for each non-terminal symbol,
for each rule,
check if the rule can potentially be applied & recurse.

- String of length n

- Rule with k symbols on the right

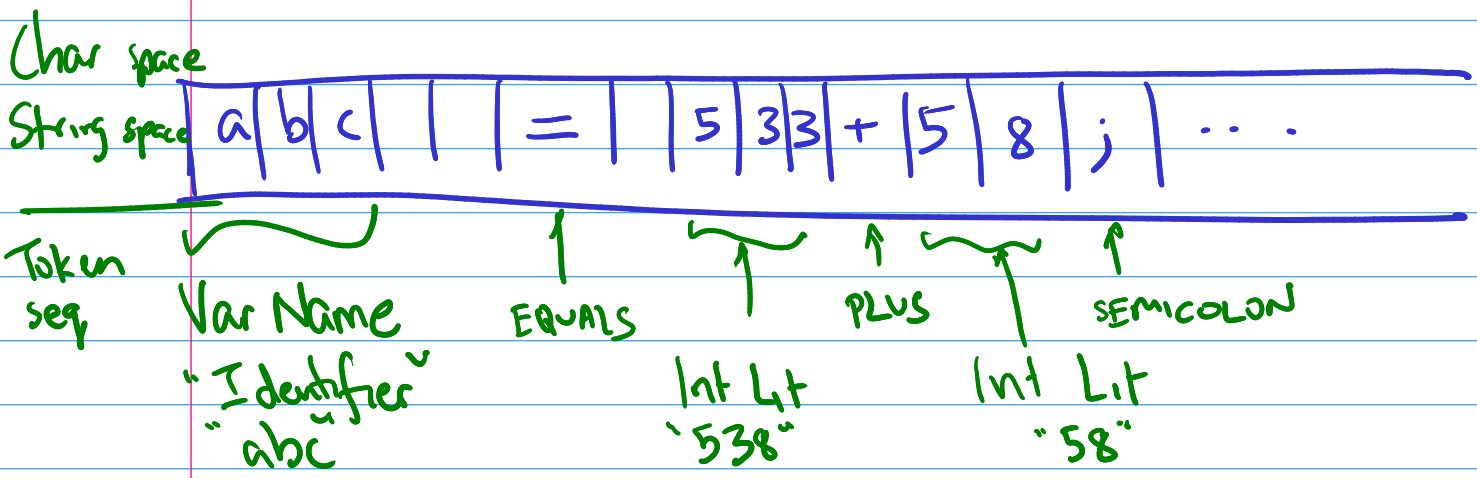
$O(n^{k-1})$ ways of splitting the string

$E ::= E + E \longrightarrow$ (Chomsky Normal Form) $E ::= E\ R$, $R ::= +\ E$

CYK algorithm

Cocke, Younger, Kasami

- Lexical analyzer

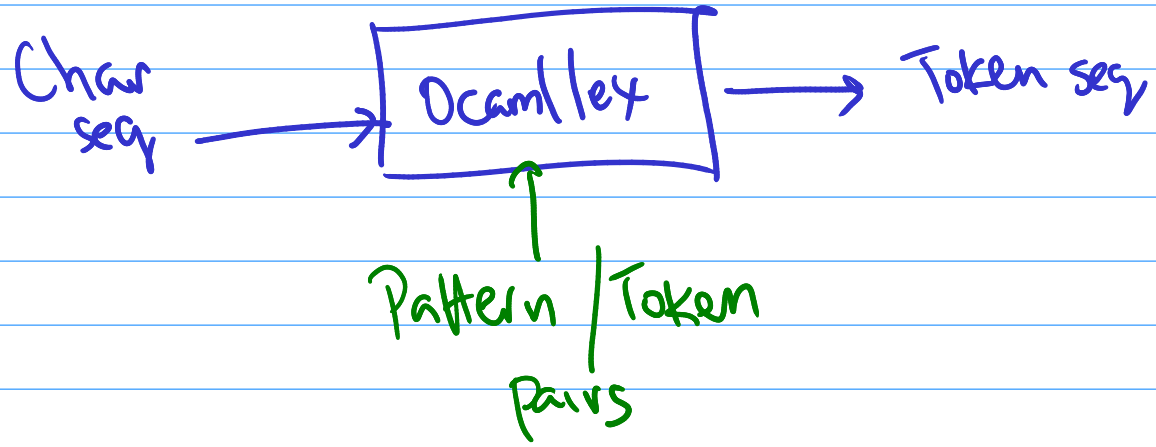


- Set of patterns $Pattern_1 \rightarrow Token_1$

$Pattern_2 \rightarrow Token_2$

$Pattern_3 \rightarrow Token_3$

- Longest match principle



"", "hi", "hihi", "hihihi", ...

hi

$(hi)^*$

"bye" "hibye" "hihibye" "hihibyebye" ...

$(hi)^* \cdot (bye)^*$

... "hi" "hello" "hihello" "hellohello"

$hello^* \; hi^* \; hello^*$

"hi hello hi hello"

$(hello^* \; hi^*)^*$ $(hello \mid hi)^*$

[a-zA-Z0-9]•[a-zA-Z0-9]*•'@'•_____'.'_____

(a|b|c|...|z|

A|B|C|...|Z|

0|1|2|...|9)

[a-zA-Z0-9]•[a-zA-Z0-9]*

(com|org|edu)

$p^+ = p \cdot p^*$ $p^* = \text{"}\text{"} \mid p^+$

Regular Expressions

"Constant string"

p^* (Kleene-*)

p^+ (Kleene-+)

$p_1 \cdot p_2$

$p_1 \mid p_2$

empty