ε — "Empty string"

Start ::= A | B (or)

A ::= ε | aA

B ::= ε | bB

① Produces strings of either all as or all bs

② Is ambiguous

Start ::= A

A ::= ε | aB

B ::= bA

① 0 or more ab-s

ab ab ab ... ab

② Unambiguous

Start ::= A_2 | A_3

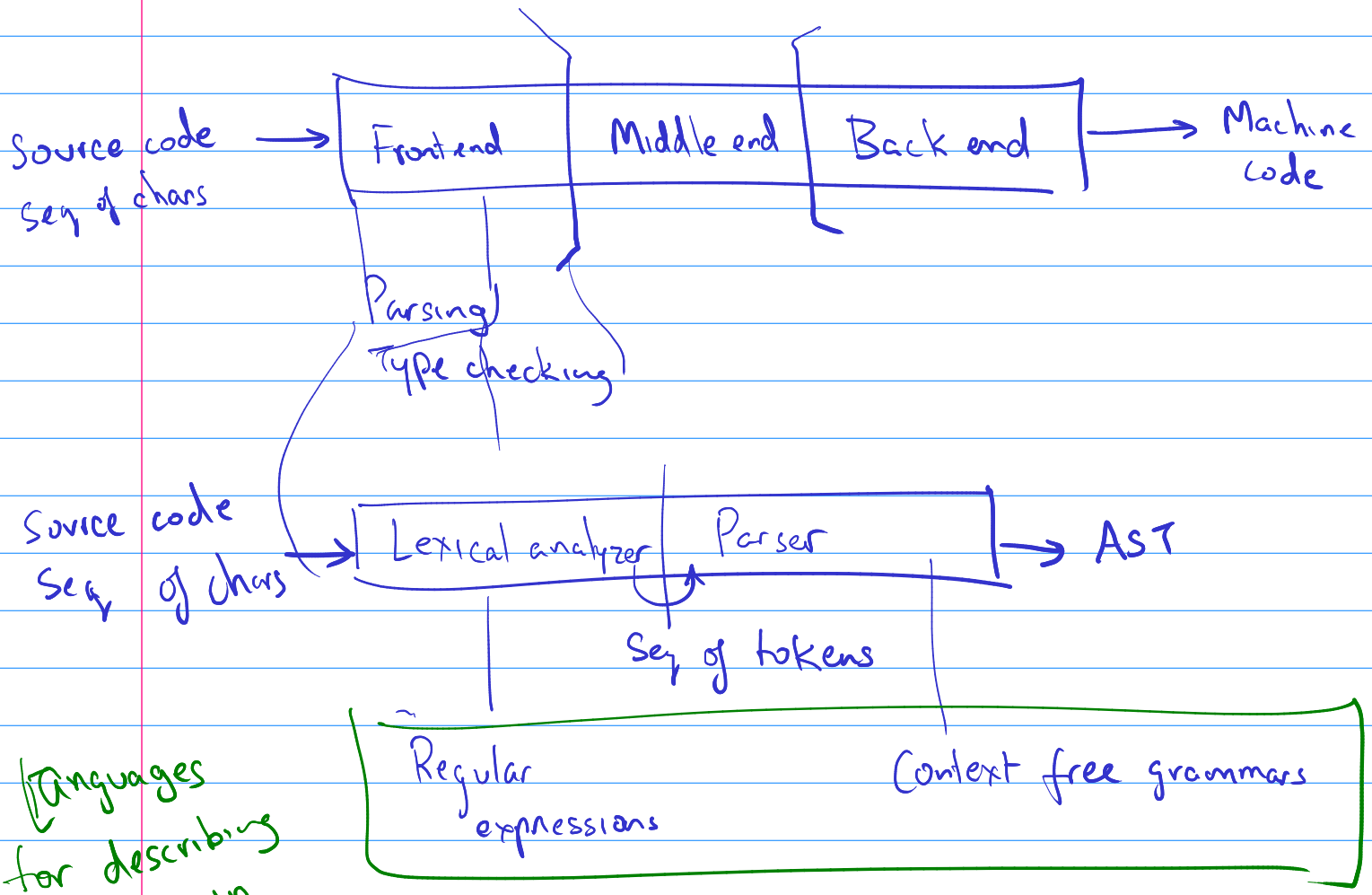
A_2 ::= ε | aa A_2

A_3 ::= aaa A_3 | ε

① $\underbrace{aaaa \cdots a}_{n}$ where n is either a multiple of 2 or a multiple of 3.

② Ambiguous for strings whose lengths are a multiple of 6.

Question: Can you rewrite this grammar in a way that is unambiguous?



Q: What algorithms do these tools even use?

How do they work?

Regular Expressions

Goal: Context-free grammar describing REs.

$$r ::= \bot \mid \varepsilon \mid a \mid r + r \mid rr \mid r^*$$

- $r + r$: Union / or / Disjunction
- $\bot$: Turnstile "Bottom"
- a: Any concrete individual character
- rr: Concatenation and
- r^*: Kleene-* Stephen Cole Kleene

Example ①

Hi

Hi Hi

Hi Hi Hi

ε

$(Hi)^*$ — Any number of Hi-s is a legal greeting

0 or more

$(Hi + Hello)^*$ — Any number of His & Hellos intermixed

0 or more

Hi Hi ✓

ε ✓

Hi Hello Hi ✓

Help ✗

$(Hi)^* \cdot (Bye)^*$ — Some # (≥ 0) of His followed by some # (≥ 0) of Bye's.

Hi Hi Bye Bye Bye

$$r ::= \checkmark 1 \mid \checkmark \varepsilon \mid \checkmark a \mid \checkmark r_1 + r_2 \mid \checkmark r_1 r_2 \mid \checkmark r^*$$

Disjunction: A string matches $r_1 + r_2$ iff

if either matches r_1 or

it matches r_2

170 — v 103 — || ??? — + (...) ∪

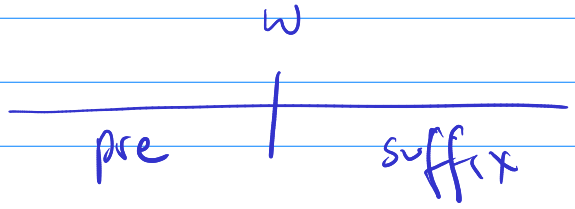
grep — |

- A string w matches a iff $w = a$

- A string w matches ε iff $w = \varepsilon$.

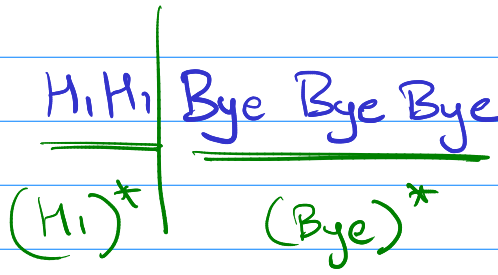
- The regex $\perp$ matches nothing at all.

Concatenation: A string w matches $r_1 r_2$ iff there is some way to split w



such that pre matches r_1 & suff matches r_2.

$(Hi)^*$ $(Bye)^*$



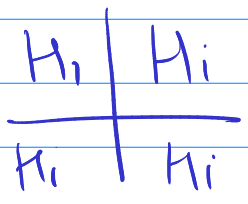
Kleene *: A string w matches r^* iff

either $w = \varepsilon$, or

there is some way of splitting w into pieces



so that each substring/piece matches r.

$(Hi)^*$ 

$(Bye)^*$ 

$\perp^*$

Regex	Math
$r_1 + \perp = r_1 = \perp + r_1$	$x + 0 = x = 0 + x$
$r_1 \cdot \varepsilon = r_1 = \varepsilon \cdot r_1$	$x \cdot 1 = x = 1 \cdot x$
$r_1 + \varepsilon$	
$r \cdot \perp = \perp = \perp \cdot r$	$x \cdot 0 = 0 = 0 \cdot x$

Alphabet — Set containing all the letters that we recognize

$\Sigma = \{A, B, C, \ldots, Z\}$

Σ = All Ascii characters

$\Sigma = \{0, 1\}$ Binary alphabet

$\Sigma = \{1\}$ Unary alphabet

$\Sigma = \{A, B, C, D, \cdots, Z\}$

$A + B + C + D + \cdots + Z$ —— matches any string of length 1.

matches the alphabet

$r + r = r$

Algorithmic Question

Given regex r & string w, does w match r?