

* How good / ^{to} measure Google IR is?

completeness → no body knows

recall precision

$\frac{x}{\infty} \rightarrow 0$ $\frac{x}{x} \rightarrow 1$

↓ ↓

can be measured

↓

subjective person to person

as it is manually checked

* How to recognize on Object

30 Aug Friday

10:50am PL Umrigar

* Wednesday programming project!!

* Expression check $\frac{a+ab}{a}$

aaab → aa ab

* Lexers AKA Scanner

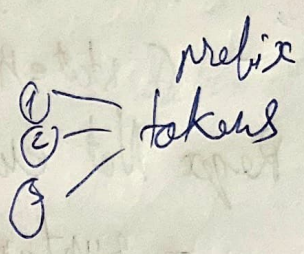
↓ generates

lex & flex using finite automata

* Maximum Munch Rule

"when scanning, recognize the longest prefix of the input which constitutes a token"
↳ even if errors are there

Eg: $V_1 = \overset{(1)}{4} \overset{(2)}{2} \overset{(3)}{+} ++a;$
 $V_2 = \overset{(1)}{4} \overset{(2)}{2} \overset{(3)}{+} ++a;$



Math star munch

int k = i /* p;
↳ this will get connected

* Keywords Vs Reserve Words

Eg: class c { // class as keyword
 f(a) {
 a.class = 22; // class as identifier
 }
}

Keywords allow easy language subsetting.
but complicates implementation

* Literals "direct representation of data value within a program"

Eg: Integers, Floats, Rationals in Lisp like languages
characters, strings

22/7, 3/40, ...

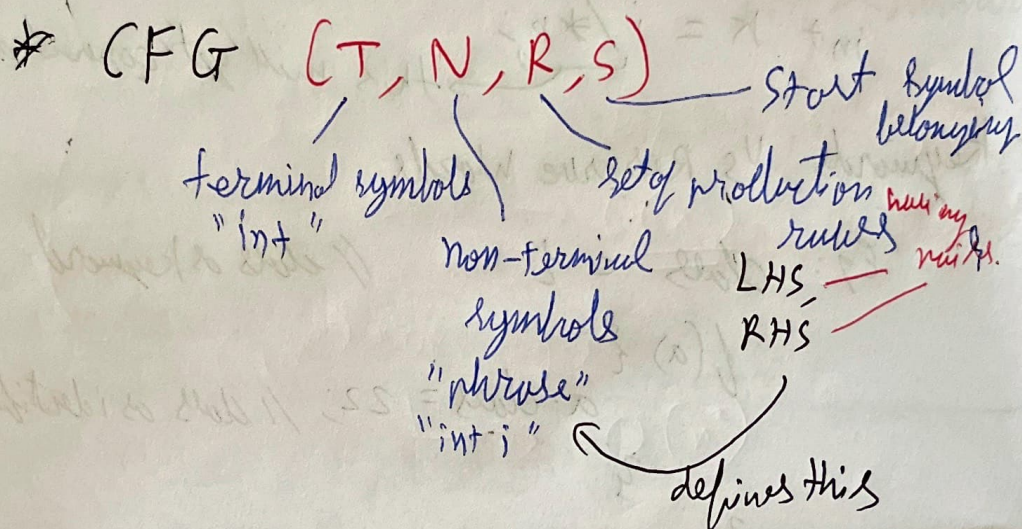
Some lang. allow array literal, maps

* Whitespace Usage in Different Programming Languages 26

- Fortran removed all white space
- Python enforces it as indentation
- C statement bug (see slides)

* Regex Not enough to specify syntax

- syntax tree for GCP program
- \therefore CFG is used to specify syntax



- Continued (see slides)

* Derivations

Eg: slope * x + intercept: (slide)

=> to denote single step in derivation

=>* to denote multiple step in derivation

Hence (see slides)

The intermediate form is called sentential form

- the final sentence form is called a yield²⁷
- the language defined by grammar is the set of all terminal sentential forms derived from the start symbol:

$$L = \{x \in T^* \mid S \Rightarrow^* x\}$$

* Parse Trees - graphical representation of derivation

If $A \Rightarrow \alpha$ is a derivation step, then add each symbol in RHS α as the children of the node corresponding to A .

Eg: parse tree for slope * x + intercept
(see slide)

first one is better
why?

* Ambiguous grammar → sentence having multiple parse trees

to handle → transform the grammar
how?

* Associative and Precedence
 \swarrow left
 \searrow right

Eg: see slide

lowest $\nearrow +$ and $-$
 \searrow left associative * and / (for exponentiation)
 highest \searrow Highest precedence right associative * or ^

* Computational Complexity

- for loop inside a for loop $\rightarrow i < n, j < n$
 then $\rightarrow \underline{O(n^2)}$

- quicksort better than merge sort
bit faster $>$ *uses additional memory* $>$ *heap sort*

miller Rabin

in assembly lang.

- for loop inside a for loop $\rightarrow i < n, j < n^2$
 then $\rightarrow \underline{O(n^3)}$

- $O(n^2) + O(n \log n)$

$\rightarrow \underline{O(n^2)}$

* Table in slides

* Little O and Little Ω loose upper & lower bounds

* Recursive Code

binary, merge sort $O(n \log n)$

If you see problem dropping by

const factor

const amount

$\rightarrow \log(\text{no. of levels})$

\rightarrow linear no. of levels

* Substitution mtd for solving recurrences

Big O complexity to be found out by recursion tree

$$T(n) = 3T(n/4) + \cancel{T(n^2)} \quad (\text{see slide})$$

$$y. T(n) = T(n/3) + T(2n/3) + \cancel{c}n$$

$$\hookrightarrow O(n \log n)$$

* The Master Mtd

$$T(n) = aT(n/b) + f(n)$$

↑ master f

Hooge sort

3 cases (see slide but not that imp)

formulax
theory ✓

* Hooge sort uses high, low
works + sorting only

1000 elements \rightarrow 0.13 sec

\rightarrow 0.0.0 (bubble sort)

10 k elements \rightarrow 80 Sec lol

quicksort 0.0 Sec