Info retreival in Info science

- recall (completeness)
- precision (Optimization)

* How good/to measure Google IR is?

completeness → no body Knows

recall precision

recall: X $\frac{x}{\infty} \to \infty$

precision ↓ Can be measured ↓ subjective person to person as it is manually checked

* How to recognize an Object

---

30 Aug Friday

10:50am PL Umrigar

* Wednesday programming project!!

* Expression check /a+ab/

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: $V1 = \underbrace{42}_{①}\ \underbrace{+}_{②}\ \underbrace{++a}_{③};$

$V2 = \underbrace{42}_{①}\ \underbrace{++}_{②}\ \underbrace{+a}_{③};$

①, ②, ③ — prefix tokens

Math star munch

int k = i/*p; → this will get commented

* 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 → 22/7, 3/40 ....
characters, strings

some lang. allow array literal, maps

* Whitespace Usage in Different Programming Languages

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

* Regex Not enough to specify syntax
  - syntax tree for GCD program

  ∴ CGF is used to specify syntax

* CFG (T, N, R, S) — Start symbol belonging

terminal symbols "int"

non-terminal symbols "phrase" "int i"

set of production rules having rules

LHS

RHS — defines this

- 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 yeild
- the language defined by grammer is the set of all terminal sentential forms derived from the start symbol:

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

* 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 grammer -> sentence having multiple parse trees

to handle -> transform the grammer

how?

* Associative and Precedence < left / right

Eg: see slide

lowest

1) + and -
2) left associative * and /
3) Highest precedence right associative ** or ^ (for exponentiation)

Highest

2:20 DAA Patrick

* Computational Complexity

- for loop inside a for loop → $i<n$ _ $j<n$

  then → $O(n^2)$

- quicksort better than merge sort heap sort

  bit faster > uses additional memory >

midterm Qs

← in assembly long.

- for loop inside a for loop → $i<n$ _ $j<n^2$

  then → $O(n^3)$

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

  → $O(n^2)$

* Table in slides

* Little O and Little omega loose upper & lower bounds

* Recursive code

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

If you see problem dropping by

const factor → log (no. of levels)

const amount → linear no. of levels

* Substitution mt^d for solving recurrences

Big O complexity to be found out by recursion tree

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

Q. $T(n) = T(n/3) + T(2n/3) + \cancel{T(n)}\, cn$

$\hookrightarrow O(n \log n)$

* The Master Mt^d

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

↖ master f^n

Stooge sort

3 cases (See slide but not that imp)

formula ✗

theory ✓

* Stooge sort uses high, low

worst sorting algo

1000 elements → 0.13 sec

→ 0.0.0 (bubble sort)

10k elements → 80sec lol

quick sort 0.0 sec