

W0 Mk 3
SMM.

9 Jan 2023
Tuesday

[lec:- 1] :-

- Instead of PL focus on PPL.
- OS is ~~also~~ abstracting out hardware details from us !!!

↳ does I/O management.
↳ process " "
↳ memory "

(low-level things are managed by OS)

Coding
Editor ID
application
compiler

↓
so that we can focus
on high level things.

OS

Hardware

→ at every level, there is

some kind of
abstraction, so

that I can do
high level thinking things →
i.e. programming.

- Abstraction is everywhere like driving car w/o knowing engine & many other things.

- why PPL?

↳ To improve ability to develop effective algo &
to use your language.

- If PL min. poor → u'll be weak at expressing
your computation / logic into code.

→ ↑ed ability to learn new languages

→ To allow better choice of PL

① Tiobe statistics : - which shows what are the
popular PL.)

WJMK
= $\frac{c}{3v}$
= $\frac{m}{v}$

→ To understand significance of implementation.

Syntax : structure / valid

Semantics : meaning of program / what does it actually do.

→ To make it easier to design a new language.

If 2 PL have same syntax & semantics, but if their implementation is different, then 2 PL are different.

If algo is same → even though no need to be equal, may not get same performance!!

as may depend on Data structure. (Map or vector)

Implementation detail

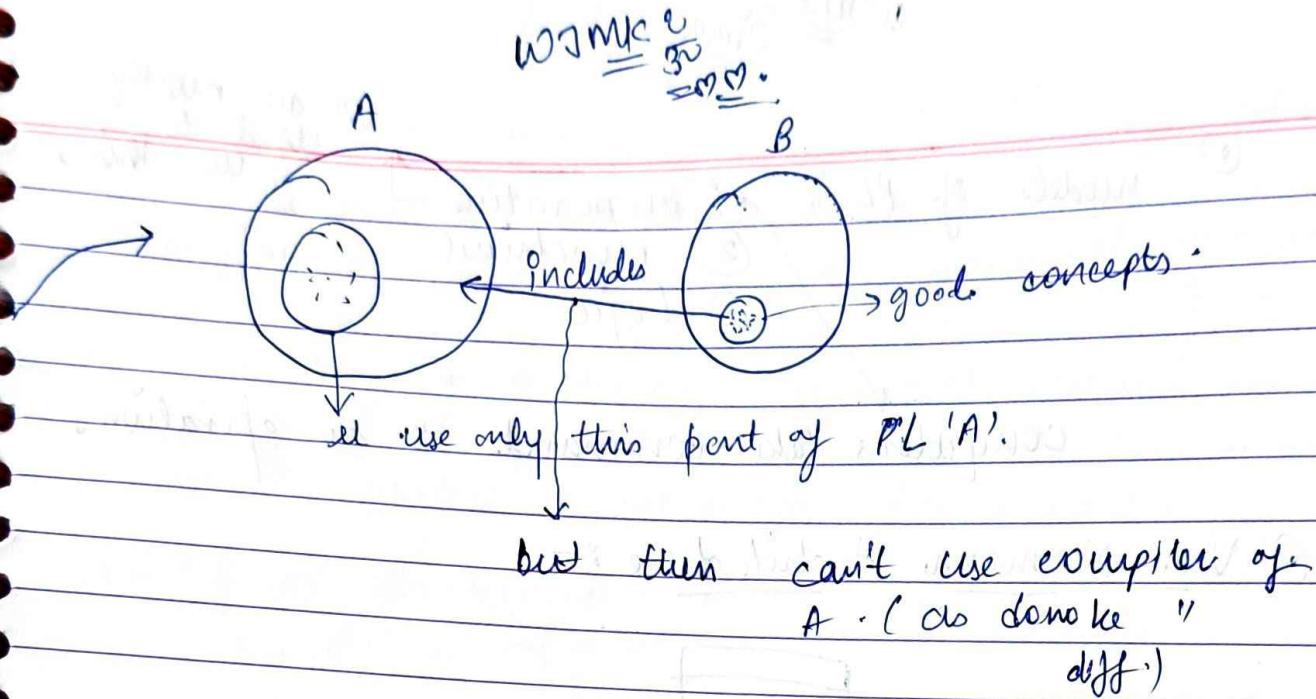
for loop use kya, yaar while loop.

$$y = x() + z() + w();$$

left to right, right to left calculation, answer is different.

so puchha phle which PL is this?

(say $x()$ ki value define se $z()$ changes & $w()$ "



so need to create new PL, new compiler.

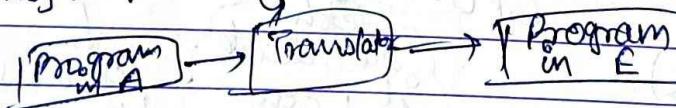
(don't have a compiler, so
donko implement kr ske)

by understanding
implementation of

A & B ka
compiler.

- Source to Source compiler? (S2SC)

↓
Transfer?



productivity: amount of efforts required.

- It depends on task & performance chalke productivity yaa both yaa more yaa

when more
focussed on
creativity

like critical
applications,

(ek voat
yeh mil)

jaaye, then
those things will be
concerned)

WOMK
= 25 mm.

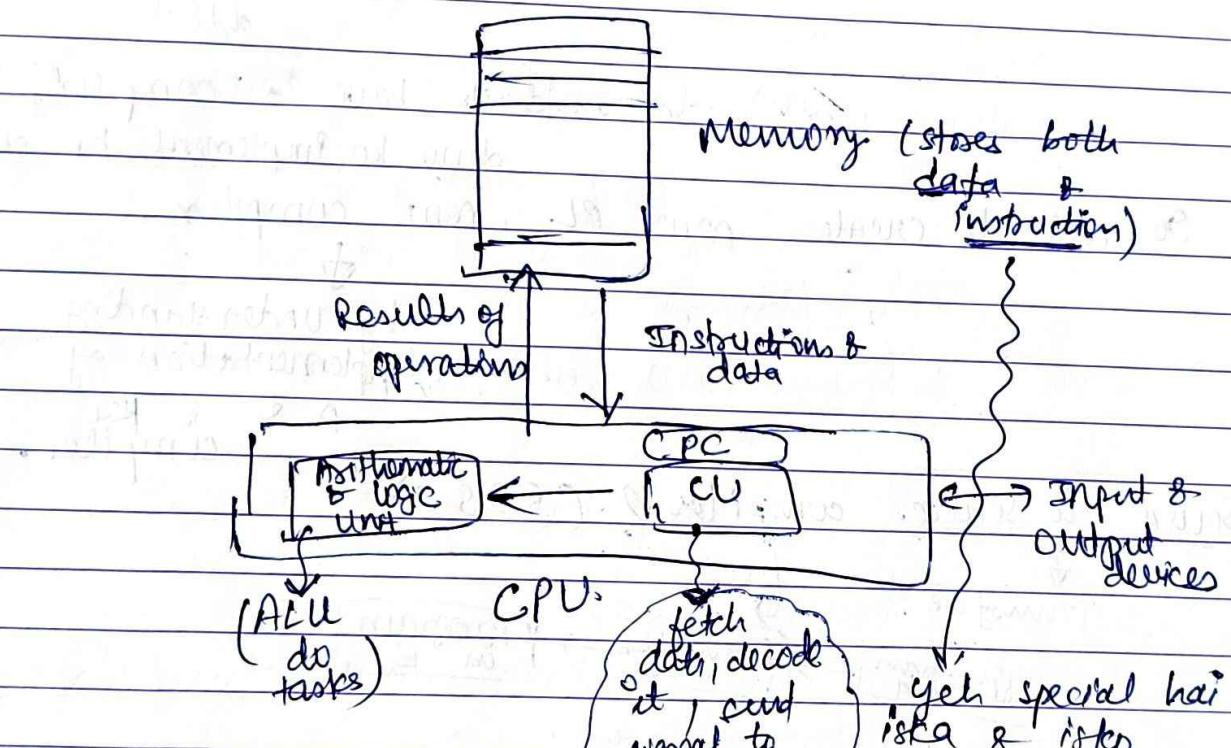
① models of PL :-

- ① Imperative ↑
- ② Functional
- ③ Logic

we are mostly used to use this.

computers take commands & do operation.

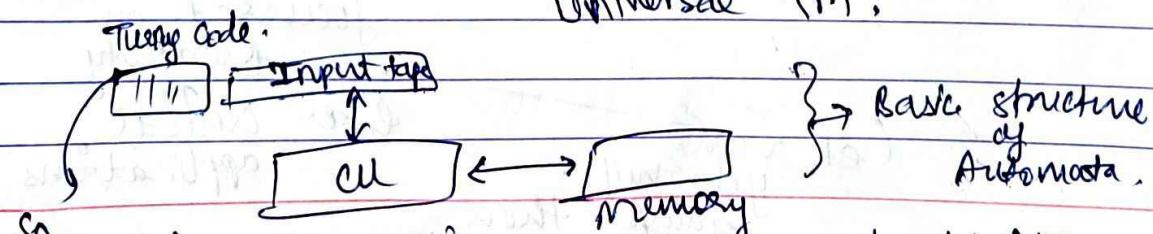
② Van Neumann Architecture :-



butta tha → of w memory stores only data !

Binary string → Turing Code of a Turing Machine

isiko doalte hain in Input tape of Universal TM.



So logic bhi' memory mein store karte hain and yehi' Von Neumann architecture mein hota.

WJMki
—~~for~~—

• imperative languages mimic von Neumann architecture :-

Variables \leftrightarrow memory cells
assignment statements \leftrightarrow data piping b/w memory & CPU
operations & expressions \leftrightarrow CPU execution.

Explicit control of execution flows \leftrightarrow PC

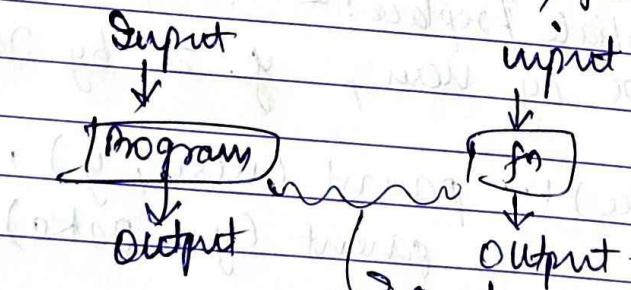
as by default toh
sequential toh hoti hai execution

li^o branching nahi hai, toh sequentially
loga execution.

2) functional :-

Programming is like solving mathematical functions

$$\text{eg : } z = f(y, g(h(x)))$$



as do same things, so express ur program as fⁿ.

of programs ~~will~~ use that's why most recursion.

Program and its subprogram are just implementation of math functions.

3) logic! -

$$WGMK = \frac{3}{5} = 60\%$$

- ① Program expressed as rules in formal logic
 - ② Execution by rule resolution
Example : relationship among people.

{ fact: mother (Joanne, Jake)
father (Vern, Joanne) }.

rule :- grandparent (x, z) :- parent (x, y),
parent (y, z)

if you
given;
can
do nothing

goal : grandparent (num, false)

If x is parent of y ,
and y is parent of z ,
then x is grandparent of z .

Need to prove this using fact & rule.

so Instantiation / replace :-

x by New, y by Z by Jake.

`grandparent (uem, jake) :- parent (uem, y),
parent (y, jake)`

~~let~~ let put $y = 30 \text{ anne}$

`grandparent (V, Jake) :- parent (V, J), parent (J, Jake).`

as both are true

so this is true! (by given rule)

facts → statements which are assumed to be true;

WOMKE

\equiv

$\exists \forall \neg \rightarrow$

or deduced \rightarrow goal.

So, established truthness of statement by using fact & rule.

10 Jan, 2023

wednesday

Lec-2 :-

gcc testa.c & ./a.out
 (compile & execute)

• ./a.out

↳ (execute)

→ ek vaar compile ho gya, fir aage sirf execute kro.

①

float i = 22.3

if (i == 2.3)

{ true; }

else false;

↳ Ans: false!

↳ as don't compare float like this, as float jiyada precision hata kar, last bits may not match, & precision error.

② (IEEE 758 floating pt. representation)

① float i = 22.3;

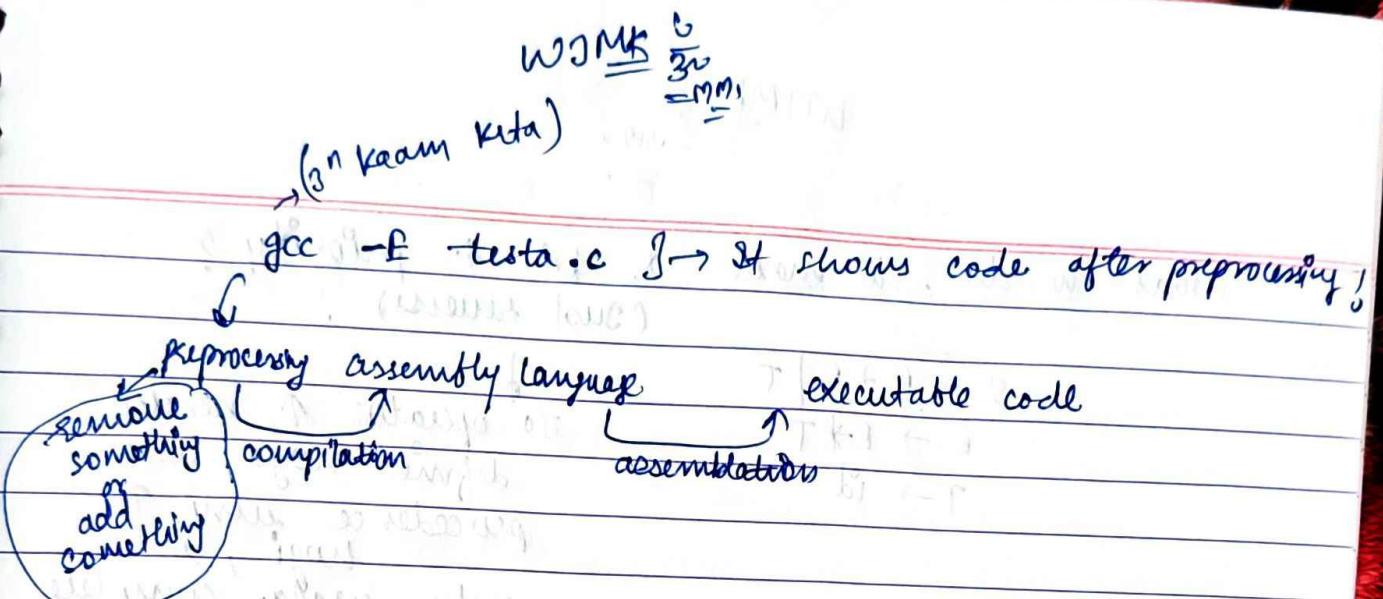
" p = 22.3;

if (i == p) ✓

↳ as both are stored with same precision error.

or $i - 22.3 < \text{min value of float} \rightarrow \text{then false}$

③ int a; (declaration)? ~~(?)~~ (?)



gcc --help → gives all options

- ① extern a; → no error.
↳ telling to compiler that there is variable 'a' used in program.

definition → allocates space, not declaration.

- ② extern a;
printf("%d", &a); → error!
↳ 'a' is undefined reference.

operator precedence → where it is actually defined?
How machine knows?

as every language ki grammar hota
hai, wahan yeh define hota!

a + b + c ; (2 operator with equal
precedence).

↳ then left se start kena yaa right
se → defined in grammar

apar kuchh se kena chabte ho,
tum bracket lgao!

$S \rightarrow E * E \mid T$?
 $E \rightarrow E + T$
 $T \rightarrow id$

→ which one will execute
1st!

→ here add will execute 1st, as E produces
(+) → and only then S will execute!

200 MAS.

WJMK $\frac{e}{S} \frac{S}{M} \frac{M}{e}$
=.

what to do to make * & / higher priority?
(just reverse).

$$S \rightarrow E + E | T$$

$$E \rightarrow E * T$$

$$T \rightarrow id$$

so operator ↑ needs

define hoga, use
precedence $\frac{e}{S} \frac{S}{M} \frac{M}{e}$ ↑
hoga, as

needs wala complete
hoga, tabhi upar wala
hoga.

① Write grammar for precedence of all 4 operators :-

$$S \rightarrow E + E | E - E | T$$

$$E \rightarrow E * T | E : T | T$$

$$T \rightarrow id$$

② garbage collection

↑ clearing unused memory & give to other users,

C does not have it!

↳ so memory utilization more!

W3MK $\equiv \frac{1}{2} \sum m_i^2$,

W3MK $\equiv \frac{1}{2} \sum m_i^2$.

16 Jan 2023

Tuesday

[lec-3] :-

why?

- Ted capacity to express ideas.
- " " " learn new language.
- Improved background for choosing appropriate language.

Evaluation (criteria) and characteristics :-

readability (so relate variable name with its use)

writability

Reliability

↳ int ✓

↳ float comparison (float) → lossing scalability

Simplicity should not be hurdle → sometimes simplicity ke lie multiple features we create for same thing so that creates confusion.

• we should have something common & must be used for all → ORTHOGONALITY. → complexity ↓.

↑
tryada orthogonality also difficult.

for eg : some languages :-

+ only for int,

for float such as

complexity ↑

orthogonality ↓.

WORK

```

int i = 22;
float p = 22.0;
if (i == 22) {
    printf("true");
}

```

⇒ Implicit type conversion

```

else
printf("false");
printf("In %f", i + p); printf("In %f",
return 0;
}

```

⇒ promotion happening
integer promoted to float.

demotion nhी
happening → data loss.

will promote to float
but again demotion happening
so garbage answer!

↓
Issues due to orthogonality,
agar explicit type conversion na krei!

In C → reserved key words, why required?

bcz of w compiler ko

dkat hogi ki wo keyword hai
ya variable name??

Do, find → can be used in loop or can be used as
variable name, then how compiler will
identify?

ek step doge detko
(every input)

int a

MM
extract prega 3rd using regular expression & then look-a-

head, jischi nhī mila, fir compare prega with

WORMS
= 20m.

reserved keywords.

→ difficult to see kaam starting, ending,
instead of bracket, we can use
do, end also!

① grep in UNIX

↳ problematic → naam se kaam nahi pta chl skta.
will show all files & unmein line no. ;
where that string ('char search word') is
present.

grep -rnrw -e "word"

* Errors in program can go undetected when nearly
any combination of primitives is legal.

↳ eg. $\frac{1}{2}$ means allowed
in coding.

int + float → allowed.

↳ but [logical error] → don't get correct/
expected output.

↳ debugging kaise kroge?

↳ thin-2 print statements daalo!

(char line k baad mati)

Support for ABSTRACTION :- (1) Process (2) Data

↳ reduces code.

Expressivity: powerful operations can be accomplished with a
complex statements.

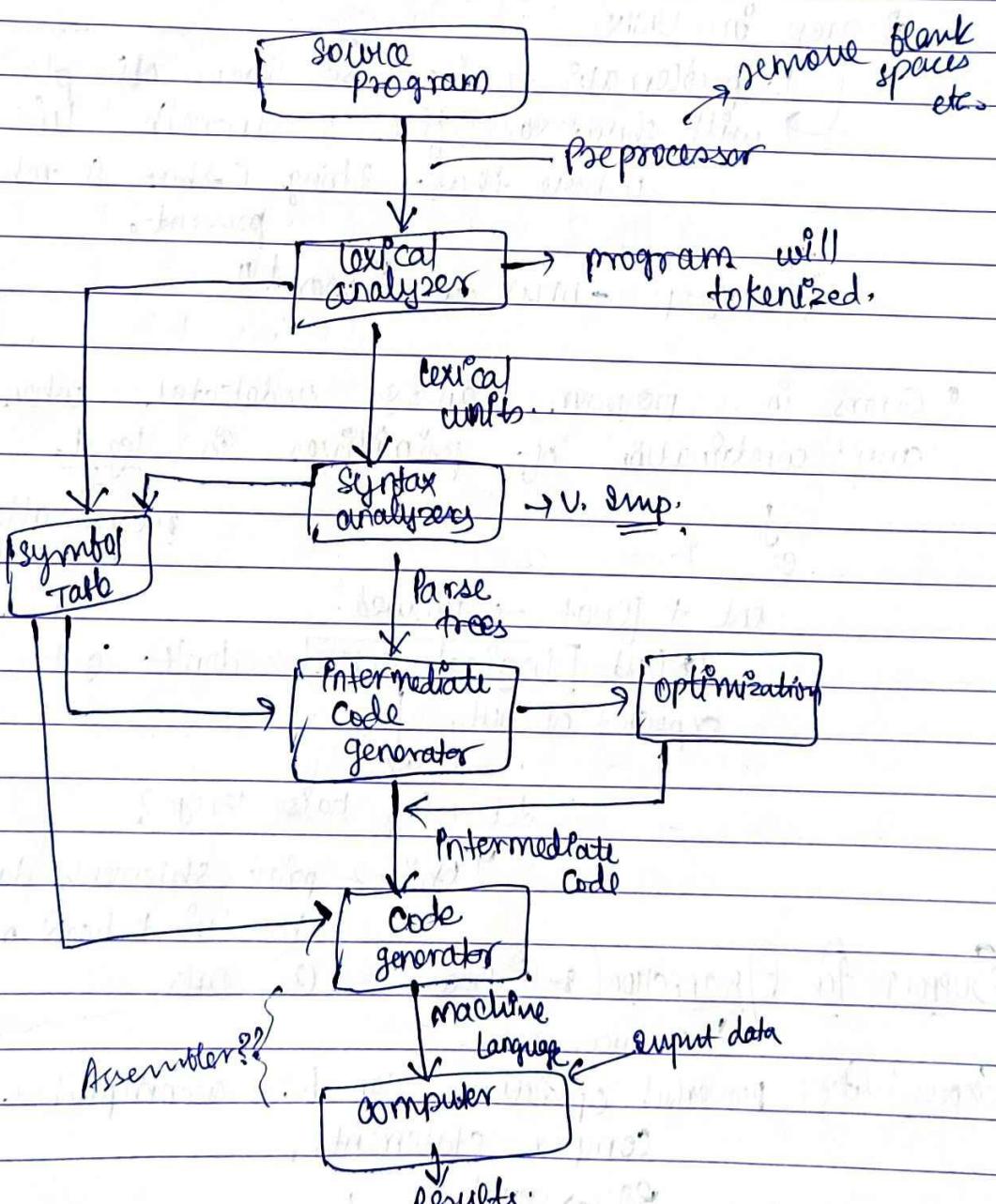
↳ eg. count++;
 count = count + 1;
Internally yeh karta → toh directly yeh likhdo
yeh.

WORK
ISSUE

try, catch → Java, C++;

→ consider maintenance & then Develop code.

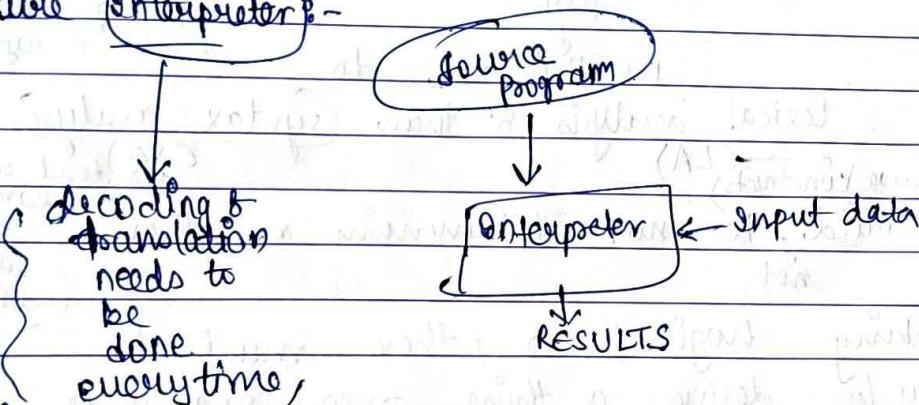
Compiler & assembler → don't use hate hair to translate user code to machine understandable language.



WOMK = $\frac{C}{50} \times 100$

- ① Using PDA → Can we see whether variable declared while also or not?
- ② Can count no. of parameters passed in function, agree & toh error?

Pure Interpreter :-



while in compiler, PK vaar compile krke kitne vaar mri execute kro!

(Disadvantage.)

Hybrid Implementation :- { Java uses both → compiler + interpreter }

End of Preliminaries :-

int a;
int c;
char b;

→ no initialization of garbage is error, they will show error!

c = a + b; → legal ✓ (as takes ASCII value & add)

scanf
bro,
no
error!

Name	Type	Value
a	int	
c	int	
b	char	

→ using this, can we identify whether variable declared or not? and declared more than one

or not?
(for this, use 'hash' data structure)

WOMK
30%
=

WOMK
30%
=

17 Jan 2023

Wednesday

Lec-4 :-

Lexical analysis :- To recognise syntax we are passing is that part of language or not.

for this we do

(use grammar)
(to analyze)

(deal with small scale language constructs) Lexical analysis & then Syntax analysis (CSA) (deal with large scale language constructs)

& there should be non-determinism as then not

backtracking hogi ! so, there must be one way to define a thing, so grammar must be deterministic

4 o/w also syntax analyzer hai,
but for linear nahi, exponential time gega !

Lexical Analyzer :- Pattern matching algorithm

Token → o/p

Pattern → something we looking for

Lexeme

return type : int

printf

("Total = %d\n", score);

lexeme

Identifier

literal lexeme

Identifier
(as will identify something)

Token : keyword, operator, identifier

earlier pre processor → white spaces remove nhi

prta tha, tab lexical analyzer prta tha
yeh kaam.

WOMK

(GCC) → compiler → save errors dilchahta, jab ki^o
 interpreter → 1st error par hi ruk jayega, can't proceed further.

→ automatically do some security mechanism, & suggest changes
 (does not mean code sahi)
 only for hojaoayega
 lexical analyzer, not code

① if () likhna tha,
 if () likh diya,

how to correct the error?

[or delete]

(skip) this, note the error,

kyi var replace krna hog and proceed further.

par replace casther than skip

if i (

↳ can be variable name (par bracket mili, so can't be variable)

lexical analyzer does 2 things :-

- ① scanning
- ② analyzing

Attributes of Tokens

Tricky problem in Token recognition :-

do 5 I = 1025

do 100 n = 2 * 10, 1

100 nfac = nfac * n

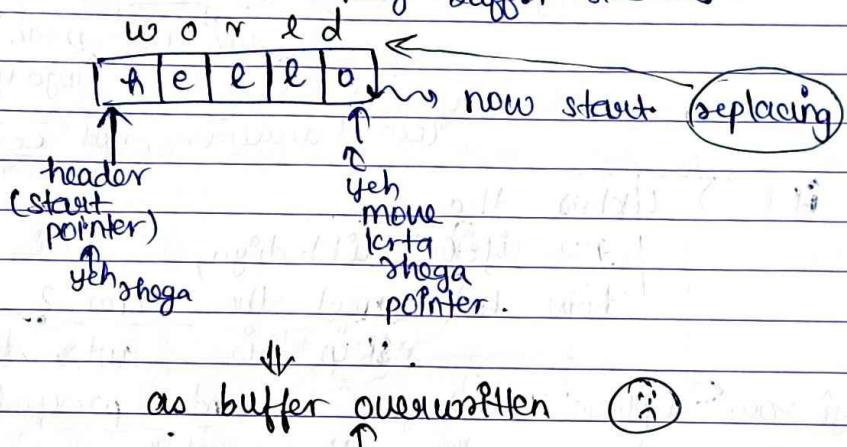
keyword responsible for loop

↑
 keyword is responsible for assignment

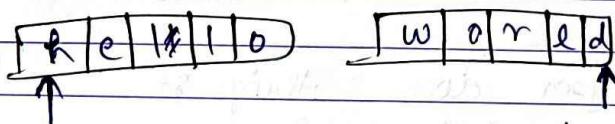
jab tak yhaan na aaye,
 pta nahi chlega kya lealem
 krrha hai (do) → so num-determinism, itna aage aake
 paa che jaanae pth raha hai!

W.M.K
= 30M. AI

int helloworld = 1;
already scanned. now let us identify it, by
putting in buffer,
say buffer size = 5.



so can't extract helloworld,
thus go to 2 buffer scheme.



now can extract 😊

But if length of variable is more
than this, then 😢

false replace karna hoga.
same problem?

so every language already defined buffer size
(4096 character k 2 buffer), use ↑ variable
name can't write.

agar in extreme case karna phl jaye,
then we use '+' for it.

W_WM_W
= 3
= M.

(dead state
trap state)

int
TTT

tak can't decide anything, step aye
jaana hogा → kitne step → say 'k'

then 'k' look ahead.

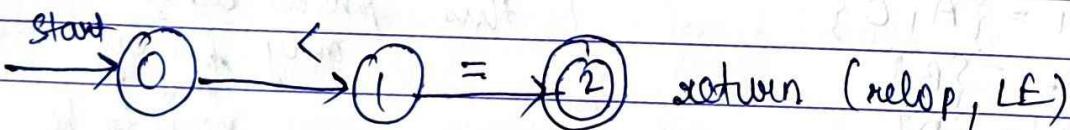
Usually $k=1$.

GCC has DFA table.

agar blunt \uparrow → chikat,
possible are go for optimized / minimized table

for a regular expression, we can have any no. of DFA.
work li hogा.

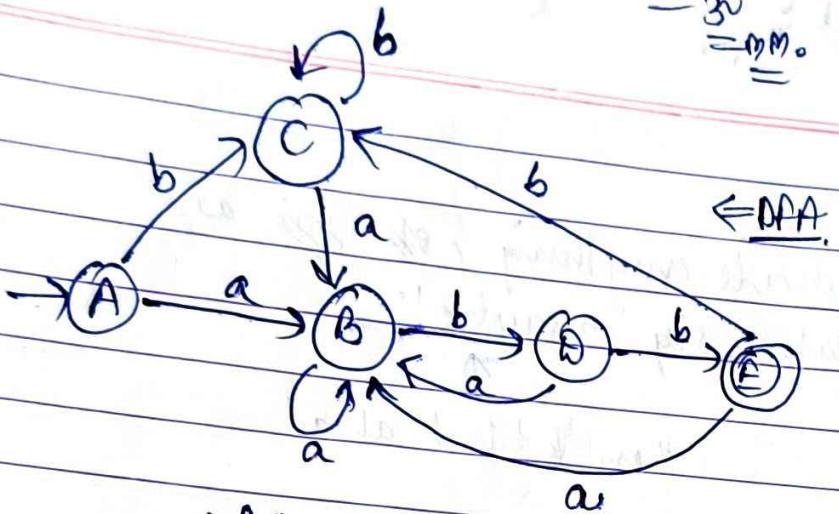
can't be NFA \Rightarrow as non-determinism nahi
from chahihe.



final state, so stop at \leftarrow ,
don't proceed further.

(no dead state needed,
as error aaya,
tum se kya bs!)

in computer agar
final state par nahi
sho patuncha, par match
nhi gya, toh nuk
jago!



Minimize it !!

$$\Pi_0 = \langle Q_0^0, Q_1^0 \rangle$$

$$Q_0^0 = \{E\}$$

$$Q_1^0 = \{A, B, C, D\}$$

$$\Rightarrow \Pi_1 = \langle Q_0^1, Q_1^1, Q_2^1, Q_3^1 \rangle$$

$$Q_0^1 = \{E\}$$

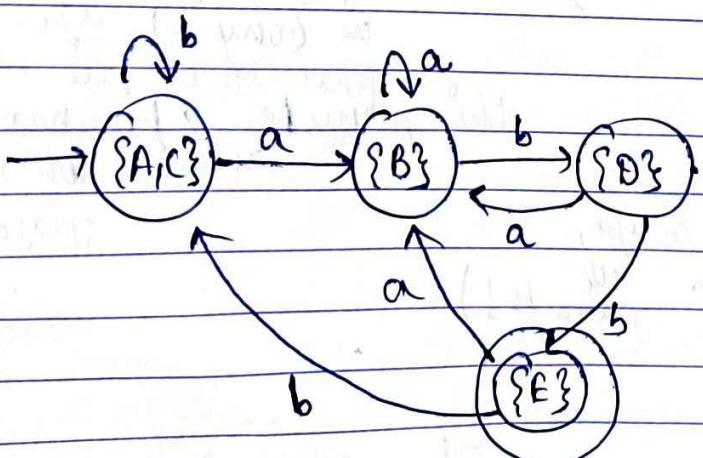
$$Q_1^1 = \{A, C\}$$

$$Q_2^1 = \{B\}$$

$$Q_3^1 = \{D\}$$

So,

minimized DFA :-



	a	b
A	B	C
B	B B	D
C	B	C
D	B	E
E	B	C

this process takes
only linear
time &
needs to be
done once,

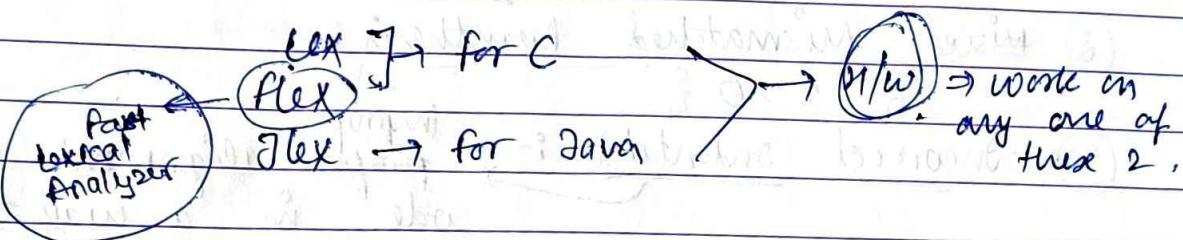
jab DFA table
ko aane
use kena
hoga,
so minimization
must !!

WJMK
= ~~3~~
~~3~~

WJMK
= ~~3~~
~~3~~

- ① Is DFA table sufficient to bring out tokens?
↳ Yes!

But in addition to that, some info also seq. sometimes.



Lexical Analyzer → also remove | skip tab spaces, also |

Q: Find 5 errors that can arise at lexical phase.

(or) syntax or semantic phase
phi ho state that?

- ① lexical error / scanning error :- (lexical analysis errors)

↳ 1st phase of compiling process.

↳ break code into tokens (like keywords, identifiers, etc.)

↳ errors are related to recognition of these tokens.

Eg:

① ~~int~~ c = 10 + @3 ;

② ~~int~~ c ;

③ ~~int~~ a ; (misspelled keyword)

④ ~~int~~ c = 5 # 3 ; (invalid character)

⑤ ~~float~~ a = 3.14 . 2 ; (numerical literal with invalid suffix)

⑥ string name = "Aditi gupta" ; (Unclosed string literal)

① Syntax Error :-

→ ② Involves structure of code and violations of the language's grammar rules.
③ Missing semi colon :-

int x = 5;

④ Mismatched Parenthesis :-

if (x > 0 {

⑤ Incorrect Indentation :-

improper
proper alignment of
code in a way that
doesn't adhere to the expected indentation rules of a
program.

e.g. :-

```
for( )  
{ if( ) {  
    if++;  
}
```

Correct:

```
for( )  
{ if( ) {
```

⑥ Misplaced Else clause :-

e.g.: if(x > 0) {
 print("positive");
}

here else clause lacks curly braces {} to explicitly define the block of code associated with it. else clause is not enclosed in curly braces, it associates itself with nearest 'if' statement.

WOMK $\frac{c}{sw}$ $\frac{M}{M}$

23 Jan 2023
Tuesday

Lec-5

→ 2 functions

to compute

for any

grammar.

① find the FIRST and FOLLOW :-

$$S \rightarrow ABCD | E$$

$$A \rightarrow a | e$$

$$B \rightarrow bA$$

$$C \rightarrow a | e$$

$$D \rightarrow d$$

To tokens from lexical phase, moves to syntax analyzer.

responsible for checking whether
i/p language is part of grammar or not.

$$E \rightarrow E + T$$

$$E \rightarrow T$$

$$T \rightarrow F$$

$$F \rightarrow id$$

see whether id + id part

of grammar ka yaa :

nahi ??.

yes!

$$\text{or } E \rightarrow E + T$$

$$\Rightarrow E \rightarrow T + T$$

$$\Rightarrow E \rightarrow F + F$$

$$\Rightarrow E \rightarrow F + F$$

$$\Rightarrow E \rightarrow id + id$$

but compiler kaise check krega ???

① $E \rightarrow E + T / T$

$T \rightarrow F$

$F \rightarrow id$

$\text{first}(E) = \text{first symbol that}$
 $\text{will replace } E$

(it
both
terminal
as non-terminal)

$\text{first}('+') = \{ + \}$

$\text{first}(id) = \{ id \}$

$$WOMK = \frac{S}{T} = MM.$$

$\text{First}(E) = ?$

$$\begin{array}{l} E \rightarrow E + T \\ E \rightarrow T \end{array}$$

First of E is E

recursion \rightarrow so drop

for 'First' \rightarrow LHS fi^0 dekhni
aur fi^0 of

First of E is (T)

non-terminal

so, First of $T = (F)$

Non-terminal,

First of F = (id)

terminal

so yeh
likeho!

First of $(A) = (B) \rightarrow$ non-terminal

\Rightarrow then first of $(A) = \text{first}(B)$

First of Terminal = that terminal only.
so, $\text{first}(' - ') = \{-\}$

so,

$\text{first}(E) = \{id\}$

and, $\text{first}(T) = \{id\}$

$\text{first}(F) = \{id\}$

①

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

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

$$F \rightarrow id$$

②

$$\text{first}() = \{ \}$$

↑
should
contains
terminabs
only!

$$\text{first}(' + ') = \{+\}$$

$$\text{first}(' - ') = \{-\}$$

$$\text{first}(' * ') = \{* \}$$

$$\text{first}(' / ') = \{/ \}$$

$$\text{first}(E) = \{id\}$$

$$\text{first}(T) = \{id\}$$

$$\text{first}(F) = \{id\}$$

$$\text{first}(' id ') = \{id\}$$

③ $S \rightarrow aA \mid ab$ (see first letter only!!) $\rightarrow S$ derives aA or ab

(grammar
is ambiguous
as there
are 2 a 's)

$$\text{first}(S) = \{a\}$$

$$\neq \{a, ab\}$$

as there should not be redundancy!!

first
letter is
 (a) .

$$S \rightarrow a \mid A \mid cd$$

$$A \rightarrow bA$$

$$\text{First}(S) = \{a, c, d\}$$

~~$$\text{work} \in \Sigma^*$$~~

$$S \rightarrow ab \mid A$$
~~$$\text{First}(S) = \{a, \text{First}(A)\}$$~~

- jo left side par hoga, sof unke liye 'first' find keliye!!
- we avoid ambiguous grammar.

$$E \rightarrow E + T \mid \del{E * T} \mid T \mid e$$

$$T \rightarrow T * F \mid \del{T / F} \mid F \mid e$$

$$F \rightarrow id$$

$$\text{First}(F) = \{id\}, e\}$$

$$\text{First}(F) = \{id\}$$

$$\text{First}(T) = \{id, e\}$$

① $S \rightarrow ABCD \mid e$

$$A \rightarrow a \mid e$$

$$B \rightarrow bA \mid aA$$

$$C \rightarrow a \mid e$$

$$D \rightarrow d$$

$$\text{First}(S) = \{a, e, \del{b}\}$$

$$\text{First}(A) = \{a, e\}$$

$$\text{First}(B) = \{b, a\}$$

$$\text{First}(C) = \{a, e\}$$

$$\text{First}(D) = \{d\}$$

② Compute 'follow':-

what symbol come after the non-terminal.

for that look on RHS, not on LHS.

$$E \rightarrow EFT \cdot E + T \mid E - T \mid T \mid e$$

$$T \rightarrow T * F \mid \del{T / F} \mid F \mid e$$

$F \rightarrow id$ → can't go to T, as '+' is barrier!!.

$$\text{follow}(E) = \{ \$, +, - \}$$

(Nowhere else 'E' nahi hai in RHS)

reason ← stack mein
base neeche → the cycle

hota, tab yahan pallancha,
means recognition is done!!,

augmented
symbol

↑
use thi represent pro, \$

dili gupta

$$WOMK = \{ \text{S}, \text{M} \}$$

so again, $E \rightarrow E + T$
 $\text{follow}(T) = \text{follow}(E)$

$$\text{follow}(T) = \{ *, +, -, \$ \}$$

let $E \rightarrow T$ $S \rightarrow Fa$
 $E \rightarrow T$

$$Fa = Ta$$

so, $\text{follow}(T) = \text{follow}(E)$.

so,

$$\text{follow}(T) = \{ *, +, -, \$ \}$$

$$\text{follow}(F) = \{ *, +, -, \$ \}$$

eg: $\Rightarrow Id + Id * Id$

$$\Rightarrow F + Id * Id$$

$$\Rightarrow T + Id * Id$$

$$\Rightarrow E + Id * Id$$

$$\Rightarrow E + F * Id$$

$$\Rightarrow E + T * Id$$

$$\Rightarrow E * Id$$

→ follow of E is *.

$$E \rightarrow E + T$$

follow of T is *.

$$T \rightarrow F$$

follow of F is *

'\$' will be contained

in 'follow' of starting symbol or

in ~~and~~ 'follow' of symbol following
starting symbol

so, $\text{follow}(E)$ mein \$ hogta,

but $\text{follow}(T)$ includes $\text{follow}(E)$,

so, us mein \$ hogta,

and as $\text{follow}(F)$ includes $\text{follow}(T)$,
toh us mein \$ hogta!

$\text{first}(C) = \{\epsilon\}$
can contain ' ϵ '.

WDMK^o
~~non-term.~~

$\text{follow}(A) = \{\}$
contains only non-terminals.
" }
never contains ϵ !!

⑥ $S \rightarrow A B C D | \epsilon$ there are adjacent non-terminals !!
 $A \rightarrow a | \epsilon$ $\text{follow}(A) = \{B\} \rightarrow$ but can't write non-terminals,
 $B \rightarrow b | A$ ~~A~~ $\rightarrow \text{follow}(A) \rightarrow \text{follow}(B)$
 $C \rightarrow a | \epsilon$
 $D \rightarrow d$.

$S \rightarrow A B C D$

$S \rightarrow A B | A C D$

follow of A is 'b'.

so, $\text{follow}(B)$ is $\text{first}(C) = a$

again C is ϵ . $\rightarrow S \rightarrow A B \epsilon$

$\text{follow}(B)$ is $\text{first}(D)$

If $D \rightarrow \epsilon$

then,

$\text{follow}(C) = \{ \} \text{ follow}(S)$.

$\text{follow}(A) = \{a, d, b\}$

$\text{follow}(B) = \{a, d\}$

$\text{follow}(C) = \{d\}$

$\text{follow}(D) = \{\$\}$

$\text{follow}(S) = \{\$\}$

① $S \rightarrow a B h$

$B \rightarrow c C$

$C \rightarrow b C | \epsilon$

$D \rightarrow E F$

$E \rightarrow g | E$

$F \rightarrow f | E$

$\text{first}(S) = \{a\}$

$\text{first}(B) = \{c\}$

$\text{first}(C) = \{\epsilon, b\}$

$\text{first}(D) = \{g, f, E\}$

$\text{first}(E) = \{g, f\}$

$\text{first}(F) = \{f, E\}$

$\text{follow}(S) = \{\$\}$

$\text{follow}(B) = \text{first}(D) = \{g, f, E\}$

$\text{follow}(C) = \text{follow}(B) = \{g, f, E\}$

$\text{follow}(D) = \{h\}$

$\text{follow}(E) = \text{first}(F) = \{f, E\}$

$\text{follow}(F) = \text{follow}(D) = \{h\}$

as, $E \rightarrow E, F \rightarrow E$

$\rightarrow F F \rightarrow E$

$\rightarrow D \rightarrow E$

thus,

$S \rightarrow a B h$
 $\rightarrow \text{follow}(A)$ contains ('h')

WORK
GOO.

① $S \rightarrow AB \text{ or } | e$

$k \rightarrow C$
↑
not reachable / Useless.

so, nothing in follow(k)

so, pehle grammar ko simplify kro !!, & then proceed!
and

never take ambiguous grammar, do
backtracking !!

② Is it possible to find that grammar is ambiguous
or not ???

any algorithm ?? (w/o selecting
input & different LMD or
RMD possible)

$S \rightarrow AB | AC$
↑
ambiguous grammar!

$\left\{ \begin{array}{l} S \rightarrow AB | AC \\ B \rightarrow d \\ C \rightarrow e \end{array} \right.$ → there is non-determinism !!
kisko loge pehle !!

G but not ambiguous ??

③ ANSI C grammar

(Agar left recursion → toh → left associativity hogi,
agar right → right " " !!)
so, associativity & precedence are defined in grammar only.

WJMK
= C
MM.

WJMK
= C
MM.

~~multiplication~~ : addition *

24 Jan 2024

Lec-6 :-

Wednesday

FIRST :-

algo to compute first

$$S \rightarrow AB$$

$$A \rightarrow E$$

$$B \rightarrow a$$

$$First(S) = \{a\}$$

$$\neq \{\epsilon, a\}$$

[top down
bottom up] & types of parsing

cmo, rms \rightarrow are known as 'Parsing',
we'll take inputs from left (can take
from right as well) and move to right
(left).

then
grammar should not

have left recursion
 \hookrightarrow agar hai toh remove it

then
passing ke liye
like Jaa skta ho!

open backtracking
kerni hogi
but nahi
pura chalte

$$E \rightarrow E + T$$

esse chalta jaayega?

lim stree pehle 1st
character delete
hain \rightarrow & wo toh
juk hi nahi
raha!

WJMK
30
=

To remove left recursion :-

$$A \rightarrow A\alpha | \beta$$

$$\Rightarrow A \rightarrow \beta A' \quad] \quad \text{with accept same input}$$

$$A' \rightarrow \alpha A' | \epsilon \quad] \quad \text{→ } \epsilon \text{ introduced}$$

e.g.: $E \rightarrow E + T | (T)$

$$\Rightarrow E \rightarrow TE' \quad]$$

$$E' \rightarrow +TE' | \epsilon$$

e.g.:

$$A \rightarrow S\alpha \quad] \quad \text{↑ indirect left recursion}$$

$$S \rightarrow A\alpha \quad] \quad \text{↑ remove it as well!}$$

$\Rightarrow A \rightarrow A\alpha \alpha \rightarrow \text{Now got direct left recursion}$

e.g.:

$$A \rightarrow A\alpha \quad (\text{no } \beta \text{ is present})$$

$$\Rightarrow A \rightarrow A\alpha$$

$$A' \rightarrow \alpha A' | \epsilon$$

as unit production \Rightarrow so can remove it
 as unit production & unreachable states are
 nothing, but just burden for us !!
so Ans: $A' \rightarrow \alpha A' | \epsilon$

e.g.: $E \rightarrow E + T | T$

$$T \rightarrow T * F | F$$

$$F \rightarrow (E) | id$$

$$\Rightarrow E \rightarrow TE'$$

$$E' \rightarrow +TE' | \epsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' | \epsilon$$

$$F \rightarrow (E) | id$$

↑ iska kaaran, no left recursion !

WJMWK
SEM.

$E \rightarrow E + T \mid T \mid F$

① $E \rightarrow E + T \mid T \mid F$

⇒ $E \rightarrow TE' \mid FE'$
 $E' \rightarrow +TE' \mid E$

② $E \rightarrow E + T \mid T \mid F \mid ET \mid F$

⇒ $E \rightarrow TE' \mid FE'$
 $E' \rightarrow +TE' \mid +FE' \mid E$

③ $A \rightarrow a\beta \mid a\gamma \rightarrow$ it is ambiguous, No!

remove
it

But non-deterministic
hai!

means ki iss raste se nahi mila toh dusre se
chlo,
agar match nahi huya, well say input is
wrong, not production chosen!!.

$A \rightarrow a\beta \mid a\gamma$

⇒ $A \rightarrow aA' \rightarrow$ left factoring

$A' \rightarrow \beta \mid \gamma$

∴ Remove left Recursion,
remove non-deterministic, find
first and follow, create table.

WJMK = $\frac{g}{\Sigma \text{opp.}}$

① $E \rightarrow TE'$

$E' \rightarrow +TE' \mid \epsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid \epsilon$

$F \rightarrow \text{id} \mid (E)$

$\text{first}(T') = \{ (*), +, \epsilon \}$

$\text{first}(E) = \{ +, \epsilon \}$

$\text{first}(F) = \{ \text{id}, (\}$

$\text{first}(T) = \{ \text{id}, (\}$

$\text{follow}(E) = \{ \$, ')' \}$

$\text{follow}(F) = \{ (*), +, \$, ')' \}$

$\text{follow}(T') = \{ +, \$, ')' \}$

$\text{follow}(E') = \{ \$, ')' \}$

$\text{follow}(T) = \{ +, \$, ')' \}$

∴

remove :- ① left recursion

② non-determinism

compute 1st & follow.

30 Jan 2023
Tuesday

Lec-7 :-

- ① $E \rightarrow TE'$
- $E' \rightarrow +TE' \quad | \quad \epsilon$
- $T \rightarrow FT'$
- $T' \rightarrow *FT' \quad | \quad \epsilon$
- $f \rightarrow id \quad | \quad (CE)$

$$\begin{aligned} \text{first}(E) &= \{ id, (C) \} \\ \text{first}(E') &= \{ \}, \\ \text{first}(T) &= \{ \}, \\ \text{first}(T') &= \{ \}, \\ \text{first}(f) &= \{ \} \end{aligned}$$

yet dono fill hongi
as $\epsilon \in \text{first}(E)$

(LL(1) Table)

	+	*	id	\$	C)
E			$E \rightarrow TE'$		$E \rightarrow TE'$	
E'	$E' \rightarrow +TE'$			$E' \rightarrow E$		$E' \rightarrow \epsilon$
T			$T \rightarrow FT'$		$T \rightarrow FT'$	
T'	$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$		$T' \rightarrow \epsilon$
F			$f \rightarrow id$		$f \rightarrow (E)$	

Table includes all terminals & non-terminals.

$\text{first}(E)$ (no entry for ϵ)
main E

↑ column ↑ row

so dekho follow(E')

getting $\$,)$
because of $E' \rightarrow E$

$E \rightarrow TE'$

ab dekho in
which production

E gives id & C ,
s.t. LHS, E ho!

WOMK
= Σ^*
= Σ^m .

$A \xrightarrow{\text{reg.}} \alpha\beta / \alpha\beta$

If any cell of table contains more than one entry, then that grammar is non-deterministic, so we'll not do parsing of such grammar.

as then which one to fetch?

Then backtracking comes in !!



→ If parser locates any empty cell then it shows error !!

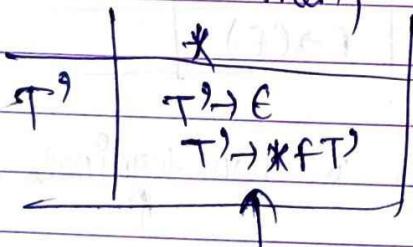
of $\exists \text{First} | \text{first} \text{ or } \text{First} | \text{Follow}$ conflict then there is non-determinism or ambiguity.

$A \xrightarrow{\text{ }} \alpha\beta$

if T' is having First | first conflict.
 $\text{First}(T') = \{a, b\}$

if, By chance, $\text{Follow}(T') = \{*, +, \$,)\}$

then,



acc. to some parser
then grammar has to be modified

2 entries \Rightarrow first/follow conflict!

①

best case = $O(n)$,
worst case can
be $O(n^3)$

Parser

Top-Down

Bottom up

everytime
 $O(n)$

Backtracking

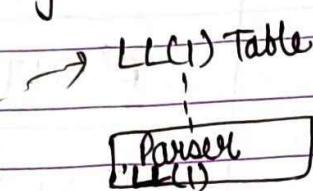
Predictive (look ahead symbol)

Can take ambiguous grammar also.

W3MK
= 50
DM =

If in LL(1) table, if no multiple entry in grammar,
then it is LL(1) grammar; else not LL(1);
we'll reject it!

is used
as
times,
life long,
tab ki,
First, Follow
are
computed
only
once !-



Input symbols



\$ → indicates that end of punch give

Stack

Input

Action

\$ E'

$E \rightarrow TE'$

↑
no symbol present
in stack, so at
stage of
acceptance

id + id * Pd \$

(E, id)

TE'

when 1st char, as pthle T ~~as~~ being, fir E' , pop E

\$ E' T

$T \rightarrow FT'$

\$ E' T' F

$F \rightarrow Id$

\$ E' T' Id

Matching pop stack move f/p
pointer

id + id + id \$

id + id * Pd \$

id + id * Pd \$

when both same, pop Id & move ~~f/p~~ pointer.

WJM
= JK
mm.

\$ E'T'	+ id * id \$	Matching pop stack move i/p pointer
\$ E'	+ id * id \$	T' → E
\$ E'T+	+ id * id \$	E' → T+E
\$ E'T	id * id \$	T → F T'
\$ E'T'F	id * id \$	F → id
\$ E'T'Id	id * id \$	matching pop stack move i/p pointer
\$ E'T'	* id \$	T' → * F T'
\$ E'T'F*	* id \$	matching pop stack move i/p pointer
\$ E'T'F	id \$	F → id
\$ E'T'Id	id \$	matching pop stack move i/p pointer
\$ E'T'	\$	T' → E
\$ E'	\$	(replace T' by E)
\$ E'	\$	E' → E
\$ E'	\$	SUCCESS!! OR ACCEPT!!
same \$ → success		

W3MK = ~~MM~~

let Input :> id + id * id \$

error recovery mechanism \rightarrow show it as error later

Now skip it!

as, (E,) \rightarrow is thal^o cell !!.

so, write Error there at end, abhi
kuch skip or any other production like 'E' \rightarrow to
remove it and abhi ignore it!

Geh tab kiya e jat want to accept, jab ki galat!

This is what LL(1) parser

does

↑

also known as
hand made parser

and no compiler currently follows this
mechanism

Pop - Down

left recursion eliminate kro hain pehle.

But in Bottom Up

↑
no worry of left recursion

hai toh sehne do, no need to
remove it !

① Construction of stem set
 \rightarrow (transition diagram)

DFA bnaana for given grammar as input!

E \rightarrow ETT | T

T \rightarrow TF | F

F \rightarrow (E) | Id

WJMK $\frac{c}{S} \frac{M}{M}$

so, we'll create item sets!

let $E \in \text{Input symbol}$

like \$, here we'll introduce a unit production,

terminal (E')

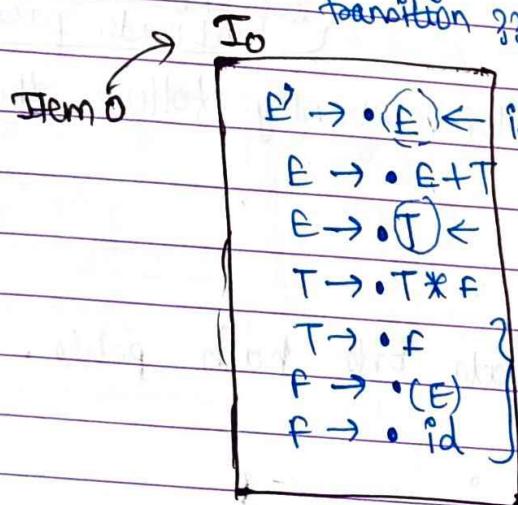
known as augumental
production with new non-

$$S \cdot t \cdot \Rightarrow \\ E' \rightarrow E$$

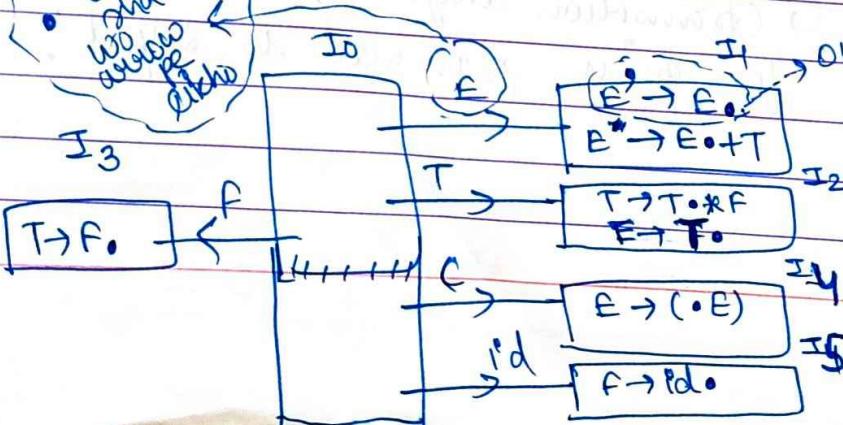
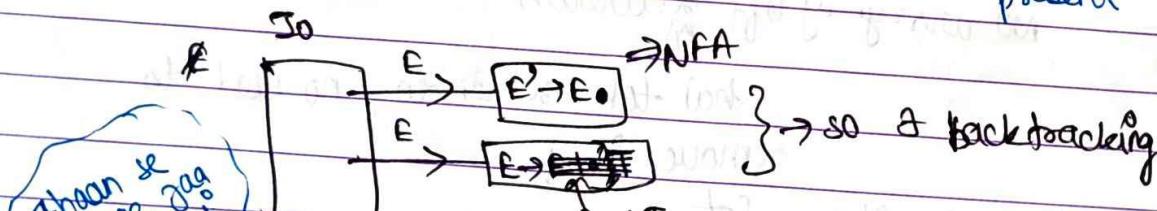
(Indicates
initially no
transition ???)

$$E' \rightarrow E \cdot$$

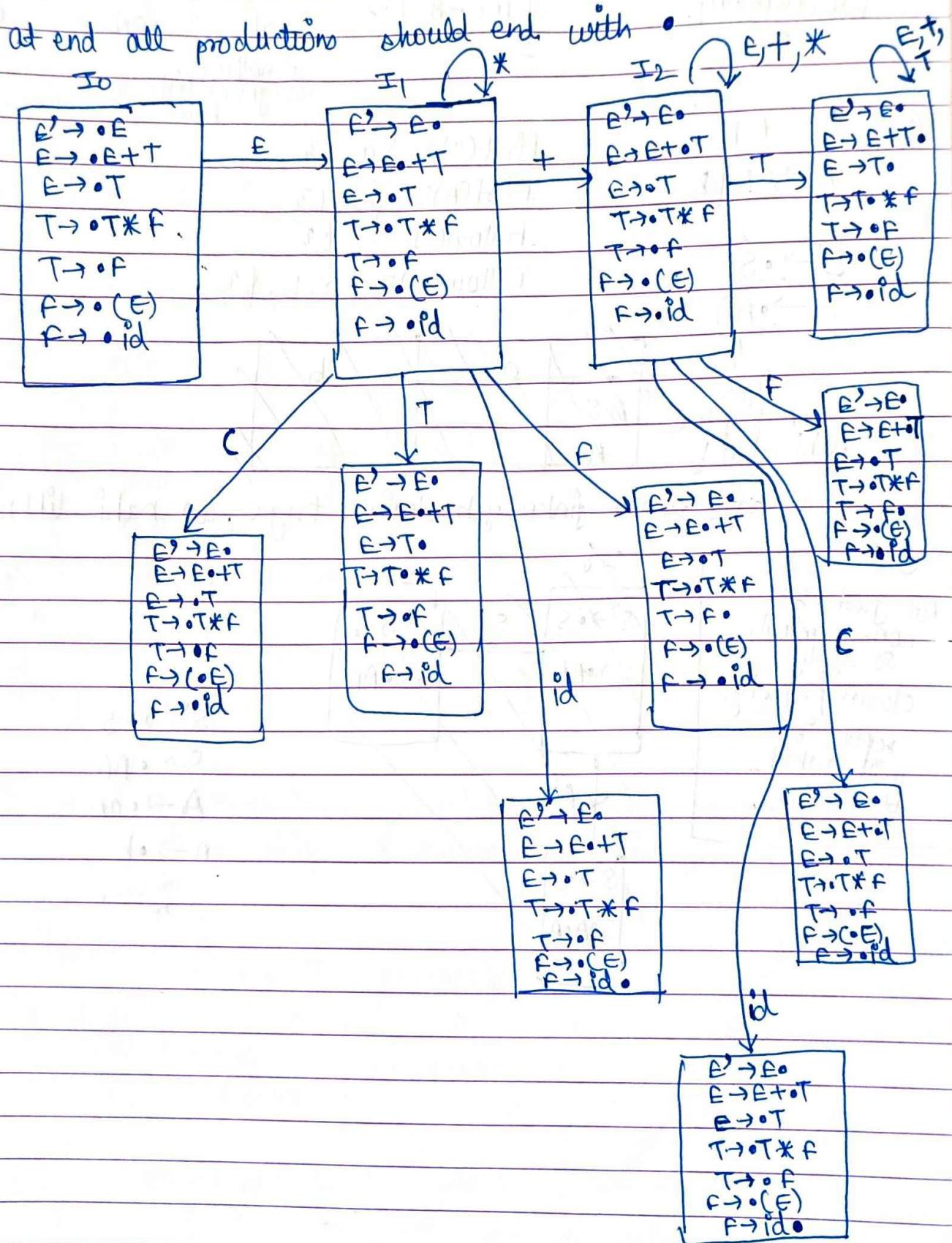
{ means there was
initially some state
isse transition kare by E ,
yhaan aaye!)



isko replace kro jinko yeh derive kar raha hai
isko replace kro jinko yeh derive kar raha hai!
This won't be written,
if $T \rightarrow F$ is not present.



WJMk
300,

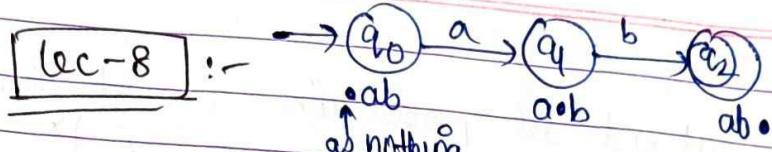


31 Jan, 2024

wednesday

$$\text{WORK} = \overline{\overline{S}} \overline{\overline{M}} \overline{\overline{K}}$$

$$\text{WORK} = \overline{\overline{S}} \overline{\overline{M}} \overline{\overline{K}}$$



as nothing
recognized till
now

① $S \rightarrow A A$
 $A \rightarrow aA \mid b$

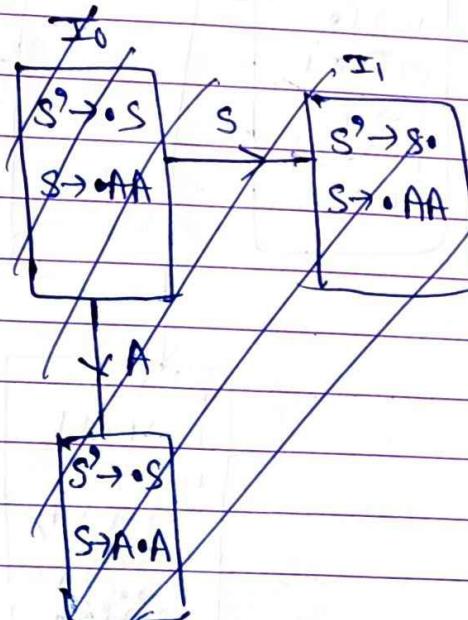
$$S' \rightarrow \bullet S$$

$$S \rightarrow \bullet A A$$

↑
single
non-terminal
nahi hain!

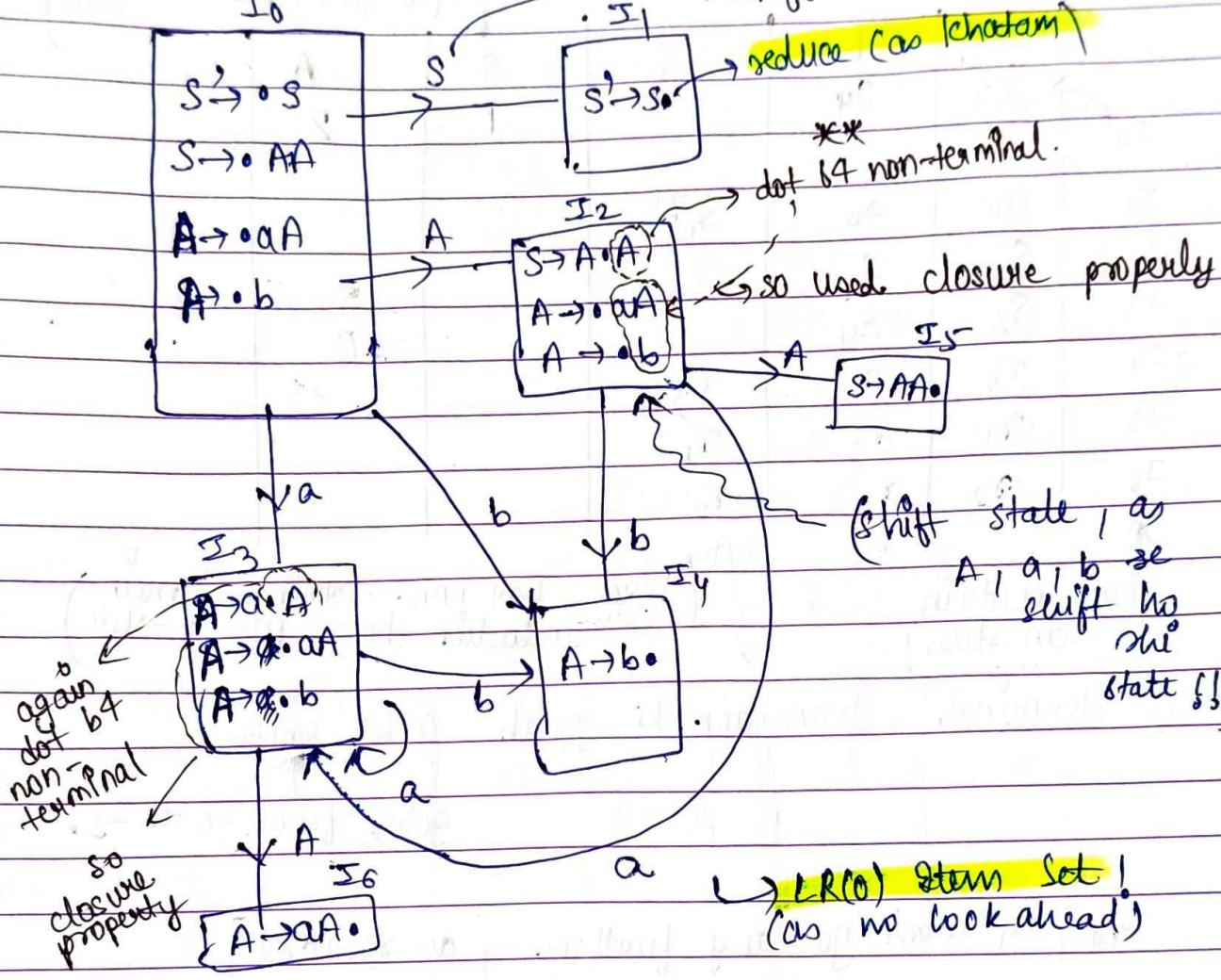
so ange derive jiske yeh derive krega, wo nahi likhna!

Dot just b4
non-terminal,
so apply
closure property,
separate it
with what
it derives.



$$\begin{aligned} S' &\rightarrow \bullet S \\ S &\rightarrow \bullet A A \\ A &\rightarrow \bullet a A \\ A &\rightarrow \bullet b \end{aligned}$$

Consequently we are making minimized DFA



Instead of transition, we'll use two terms :-
shift & reduce functions.

Top Down :-

$$E \rightarrow E+T \text{ (produce)}$$

Bottom Up :-

$$\begin{aligned} &\Rightarrow E \\ &\Rightarrow E+T \end{aligned} \quad \text{(reduce)}$$

Now we'll make ~~LL(1)~~ table :-

~~LL(1)~~
LR(0)

as no lookahead

if using this table,
then
LR(0) parser

W3MK

column contains terminals as well as non-terminals
(as going from I_0 to I_1 , state)

	a	b	\$	s	A
I_0	s_3	s_4			2
I_1	r_0	r_0			
I_2	s_3	s_4			
I_3	s_3	s_4			
I_4	r_3	r_3			
I_5	r_1	r_1			
I_6	r_2	r_2			

rows contains
item states!

for terminal, transition ki jgaah likho shift

(here, have non-terminal main reduction hence par \Rightarrow shift)

going from I_0 to I_3 .

as, I_1 won't go any further, as reduced!

↑ right
see on ~~left~~ side!

it is $S' \rightarrow S$.

ustka production number

likho in all
non-terminals

likho.

$r_0 \rightarrow$ going to reduce on basis of production $s \rightarrow$
reduction \rightarrow rule no s_3 . $s' \rightarrow s$

production
rule
no.

- ① $S \rightarrow S$
- ② $S \rightarrow AA$
- ③ $A \rightarrow aA$
- ④ $A \rightarrow b$

WJMk $\frac{S}{S'} = \underline{\underline{S}}$

as Bottom-up wala applying

$$S' \rightarrow S$$

jab S ke end

mein pauncha, so final state pe
paunch gye



Thus when no nahi, accept

echo in \$.

parsing is same, but table will differ.

(Another parser) SLR Table

*(multiple lines to differentiate
two terminal & non-")*

I	a	b	\$	S	A
I ₀	s ₃				
I ₁		s ₄		Accept	
I ₂	s ₃	s ₄			5
I ₃	s ₃	s ₄			6
I ₄	r ₃	r ₃	r ₃		
I ₅			r ₁		
I ₆	r ₂	r ₂	r ₂		

as follow(s)

non-terminal
main follow
non-terminal
main shift

By default, in any parser, I₁ will get accepted.

first reduction happening || ③ $A \rightarrow b$

left side ka follow dekho $\rightarrow s^*$ of us mein
bhdo r₃.

WAMK $\frac{0}{3}$ mm.

① 4 parsers in Bottom-up:-

- 1) LR(0) \rightarrow not used in modern compiler.
- 2) SLR
- 3) CLR
- 4) LALR

(RCI)

↑
there is look ahead!

-- In sub tables mein, agar kisi cell mein multiple entry
then grammar is ambiguous, so we'll reject
it!

① Advantage of SLR over LR(0) :-

- 1) • less grammar rejection
- 2) • identify error quickly
- 3) • no shift / shift conflict
- 4) • reduce / reduce conflict can have \leftarrow & ~~state~~. reduce can happen at same state.

5 Feb, 2023
Monday

PPL
lec-9

$$LR(1) = LR(0) + \text{lookahead}$$

Item set Item set

$$E \rightarrow E + T$$

$$E \rightarrow T$$

$$T \rightarrow T * F$$

$$T \rightarrow F$$

$$F \rightarrow (F)$$

$$F \rightarrow id$$

pehle $LR(0)$ bnaayo \rightarrow usmeni change
 kro to make $LR(1)$.

$$E' \rightarrow \bullet E$$
 (augmented production)

$$E \rightarrow \bullet E + T$$

$$E \rightarrow \bullet T$$

$$T \rightarrow \bullet T * F$$

$$T \rightarrow \bullet F$$

$$F \rightarrow \bullet (F)$$

$$F \rightarrow \bullet id$$

as lookahead \leftarrow add follow of E'

$$E' \rightarrow \bullet E, \$$$

$$E \rightarrow \bullet E + T, \$$$

$$E \rightarrow \bullet T, \$$$

yehe
done
just
bcz of $E \rightarrow \bullet E$

toh E'
ka follow

li likha

ghar par, inka khud ka follow nahi nikala.

$$T \rightarrow \bullet T * F, \$$$

$$T \rightarrow \bullet F, \$$$

$$F \rightarrow \bullet (F), \$$$

$$F \rightarrow \bullet id, \$$$

→ If there are more than 1 follow for 1 production, then use ↑
 Item set

WOMK = ~~30~~
~~30~~ M.

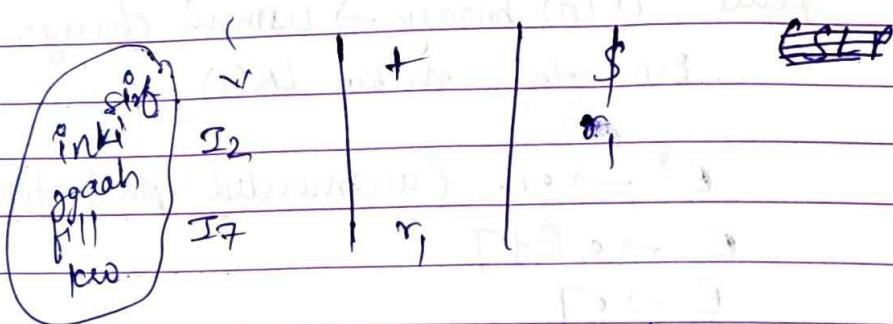
I₂ (say)

eg: $\{ E \rightarrow E + T, \$ \}$ → done ke alag-2 Item
 $\{ E \rightarrow E + T, \$ \}$ honge II.
I₇ (say)

Can merge these two Item sets?

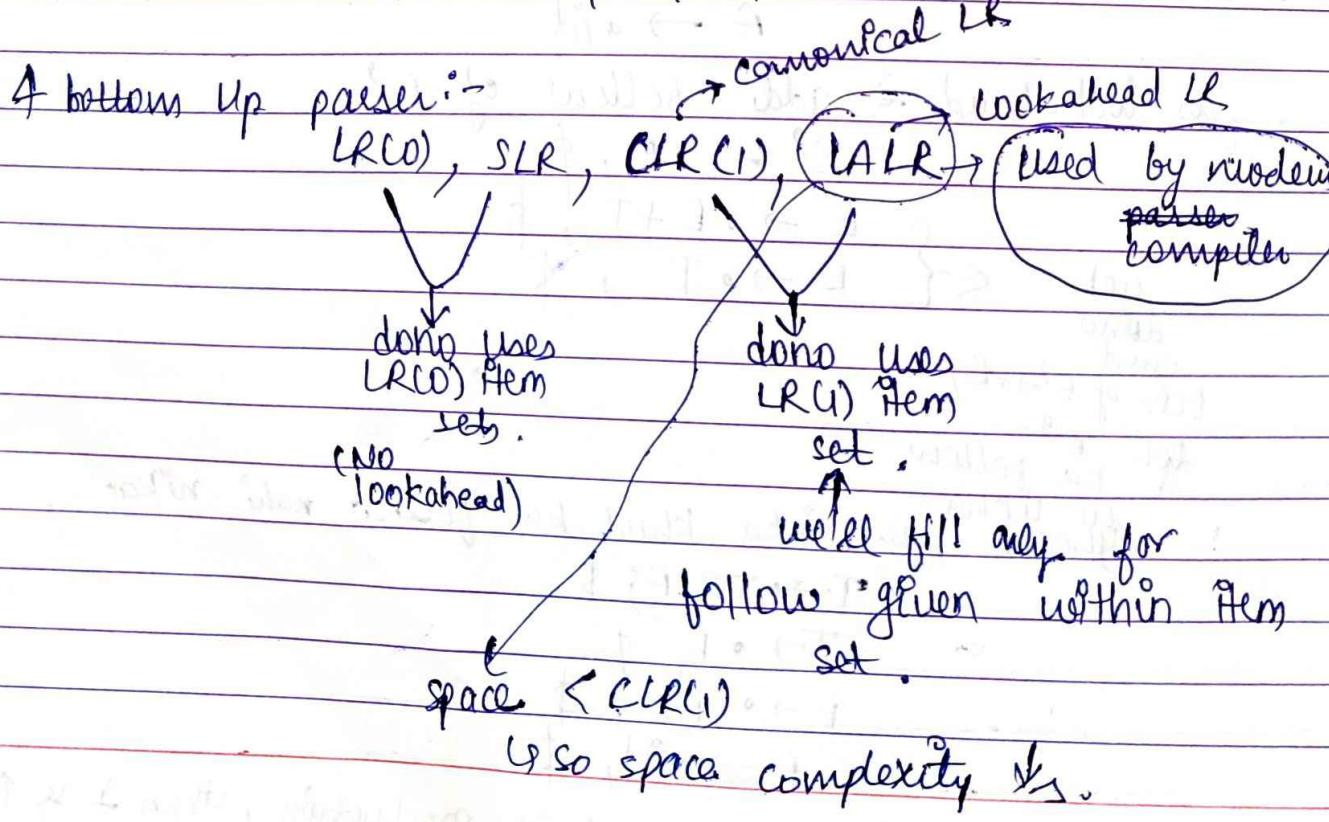
~~yes~~

as reductions no. (r_1, \dots, r_n) → same
hoga done ko

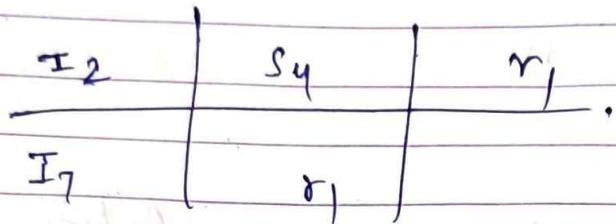


Let I₂, I₇ merged → named as I₂₇

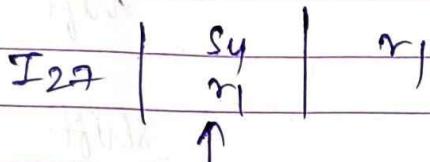
I₂₇ | - + | \$ → NO problem!



WOMK = $\frac{S}{S^*}$



\Downarrow merge



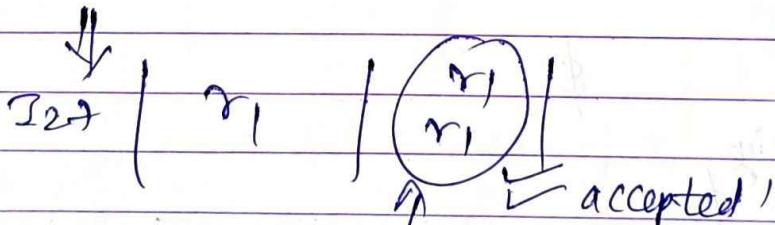
conflict, won't get accepted !! (by LALR)

(but accepted by SLR)

There is no reduce-reduce

conflict,

can have shift-shift | shift-reduce
conflict.



① So reduce no. of states + go for parsing.

② CLR will accept she gyada grammar ???

jab tak na kham ki' deile & show kovi grammar
will accept, can tell ki' yeh conflict ke kaaran
yeh parser accept nahi' keega, but yeh kelega!

~~WDM~~ ~~DP~~ ~~MM~~

$S \rightarrow AA$

$A \rightarrow aA$

$A \rightarrow (b)$

or ~~waste (3)~~

Stack

\$0

\$0 a_3 means (a_1, a_2)

\$0 a3 a3

\$0 a3 a3 b4

↓
pop this
, and
6 litchma
tha!,

Input

aab\$

ab\$

b\$

\$

Action

shift

shift

shift

$\xrightarrow{r_3} A \rightarrow (b) \xrightarrow{l} \text{so } 1x_2 = 2$
pop b4 from stack (2 things)
→ again \$0 334

↑ only 1 pop
left then !!

~~\$0 a3 a3~~

↓
reduce key
no option with \$
so mistake then !!

\$0 a3 a6

↓ pop

\$

\$0

\$

so Input won't accept!

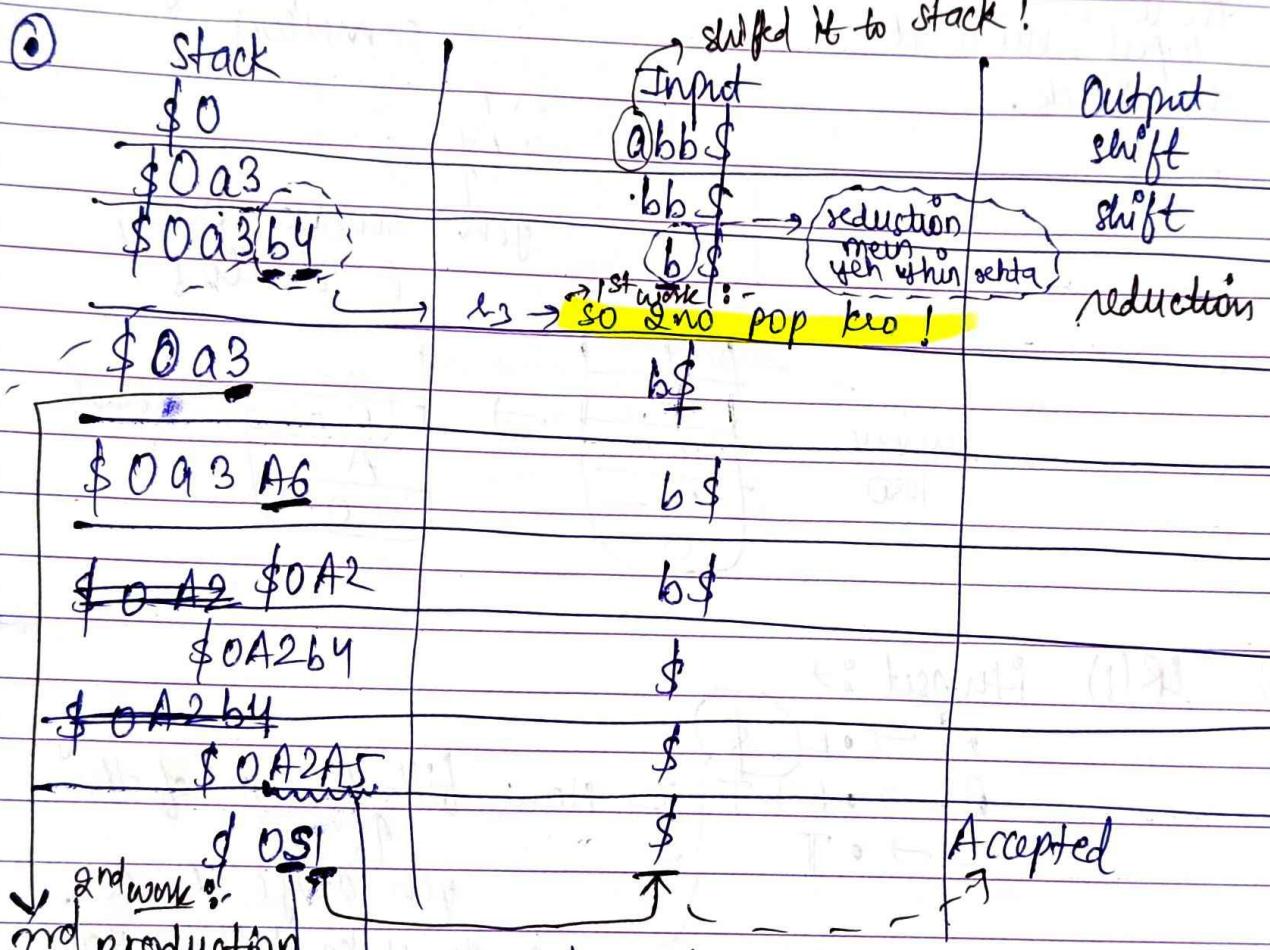
$$WAMK = \frac{S}{N} = \frac{M}{M}$$

If you know input is wrong \rightarrow apply Error Recovery Method.

SLR table use kiya \rightarrow SLR parser

LR

"



3rd production
Left side A
 $\Rightarrow (I_3, A) = 6$. yeh reduce bcoz of r,
so 1st production ka left wala likha i.e.
S and To ke s pe 1, toh wo likha jayega

\uparrow so likha A6 (yeh 2 yellow kaam ek
saath keno hain is reduction)

3rd production.
kiuki
 $b4 \rightarrow dlt$ kiya,
tha 3rd production
k kaam!!
(r_3)

q/w do for SLR table.

WORK

There are 2 actions :-

① shift

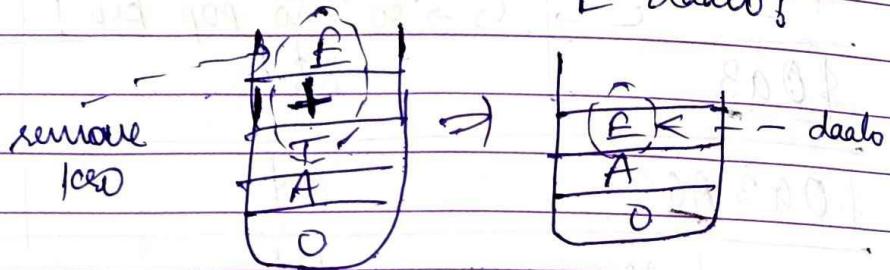
whatever
there is in
input, move it
to stack.

② Reduce

removes something from
stack & putting
something

$\Rightarrow E$
 $\Rightarrow E + T$

yeh remove kare,
E daalo!



① LR(0) Hence :-

$$E' \rightarrow \cdot E (\$)$$

$$E \rightarrow \cdot E + T$$

$$E \rightarrow \cdot T$$

} Now fill follow of this

yeh aage E se.

so, jis production
k bagan yeh daao
like.

$$\left. \begin{array}{l} E \rightarrow \cdot E + T, \$ \\ E \rightarrow \cdot T, \$ \end{array} \right\}$$

$$E \rightarrow \cdot E + T, + \quad \text{(as follow}(E)\text{ contains +)}$$

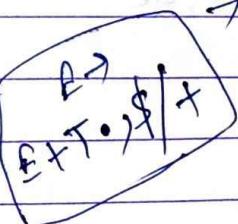
$$E \rightarrow \cdot T, +$$

$$T \rightarrow \cdot T * F, \$$$

$$T \rightarrow \cdot F, \$$$

$$T \rightarrow \cdot T * F, +$$

$$T \rightarrow \cdot F, + \quad \text{(very lengthy)}$$



WJMK
3v
29

7 February 2023

Wednesday

[lec-10] :-

$$\begin{aligned} \textcircled{1} \quad S &\rightarrow AA \\ A &\rightarrow aA \\ A &\rightarrow b \end{aligned}$$

for augmented production, take \$
as default follow.

$$\text{first}(S) = \{a, b\}$$

$$\text{first}(A) = \{a, b\}$$

$$\text{follow}(S) = \{\$\}$$

$$\text{follow}(A) = \{b, \$, a\}$$

LH(1) itemset :-

α β

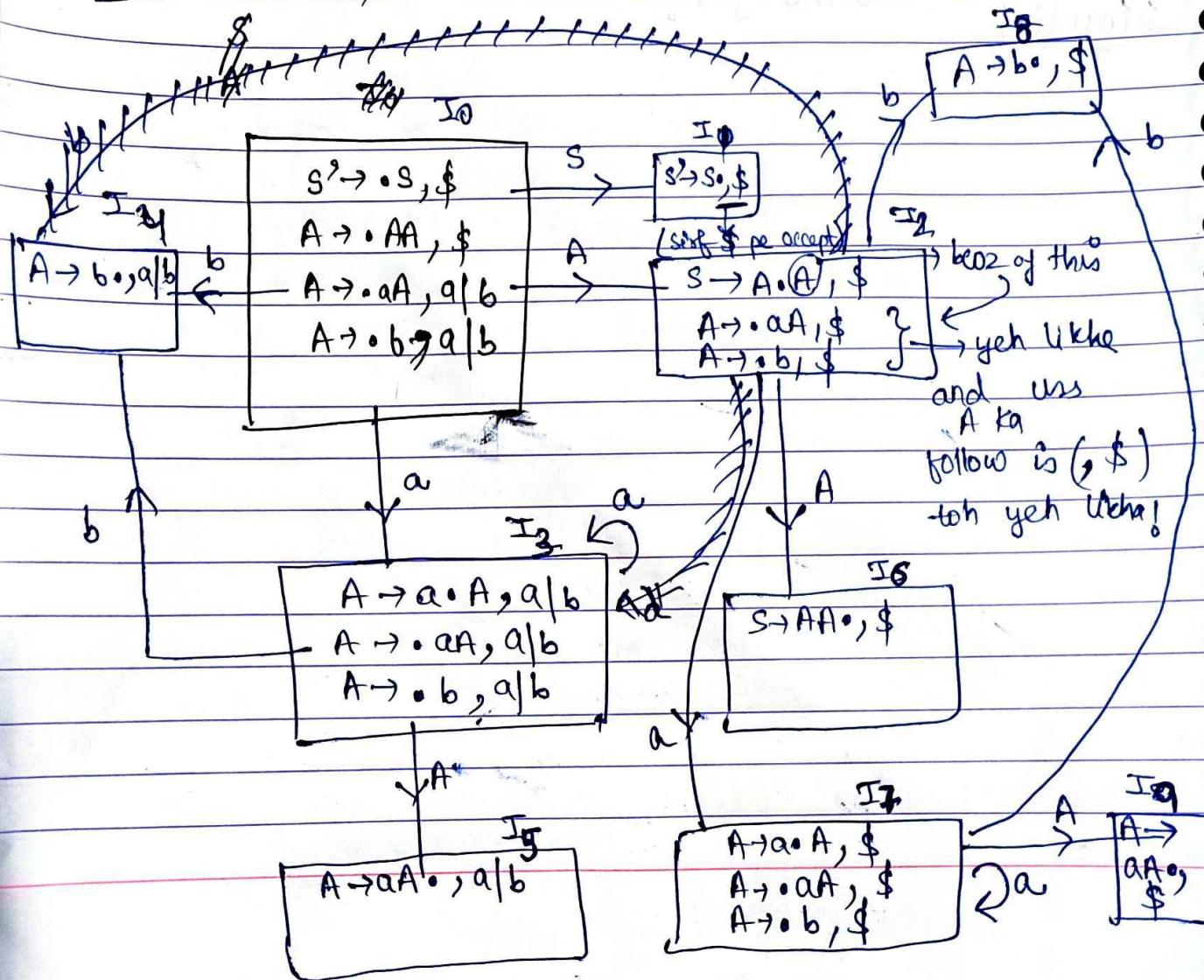
$$S' \rightarrow \cdot S, \$$$

$$S \rightarrow \cdot AA, \$$$

$$\begin{array}{l} \text{isko} \\ \text{karanjan} \end{array} \quad A \rightarrow \cdot aA, a/b$$

$$\begin{array}{l} \text{yeh} \\ \text{taya} \end{array} \quad A \rightarrow \cdot b, a/b$$

so, A ka follow is $A, \$$,
means A ka first.



as bottom up parser,
when reached
to production,
means accepted!

W.M.K \Rightarrow canonical

CLR Table :- (made from LR(1) Item set)

	a	b	\$	A	S
I ₀	S ₃	S ₄		R ₂	I
I ₁			Accepted		
I ₂	S ₇ S₄	S₈ S ₈		R ₆	6
I ₃	R₃ S ₃	S ₄			5
I ₄	R ₃ (as 3 rd production)		R ₃		
I ₅			R ₈		
I ₆			R ₁		
I ₇	S ₇	S ₈		R ₉	
I ₈			R ₃		
I ₉			R ₂		

if u do this table directly to
do parsing, then it is
canonical parsing

→ agar kisi cell mein ek se \uparrow elements
aage toh ~~two~~ two canonical
grammar nahi hoga!
(in canonical LR table)

WOMK

① for same grammar:-

$$\begin{aligned} S &\rightarrow AA \\ A &\rightarrow aA \\ A &\rightarrow b \end{aligned}$$

LR(0) \rightarrow 7 rows

SLR \rightarrow 7 rows

CLR(1) \rightarrow 10 rows \rightarrow complexity ↑

need to reduce it
↑

so went to another parser

↑
Lookahead LR Parser

↑

we will merge some states.

ycb $A \rightarrow b \cdot, \$ \rightarrow I_8$ both are reductions, getting merged,
 $I_2 \text{ se}$ $A \rightarrow b \cdot, a/b \rightarrow I_4$ instead of 2 rows, write
 aaya Instead of 2 rows, write Single row: I_48 $I_4 \text{ se aaya!}$

I_{48}	a	b	\$	s	A
	r_3	r_3	r_3		

\downarrow from I_4 \downarrow from I_8

But here conflicts mein arrive \rightarrow same cell, 2 shifts at different item set!

WJMK
~~5~~
~~M~~

	a	b	\$	s	A
I ₀	S ₃	S ₄			
I ₂	S ₇	S ₈			

↑
 Can't merge !!.

only
 then can
 merge !!.

if 2 and 6
 can merge

• Can 2, 6 be merged ??.

again

I ₃	S ₃	S ₄		8
I ₆	S ₈	S ₇		9

similar,
 don't consider
 follow part, just
 merge them!

I ₃ 8	S ₃ 8	S ₄ 7	*	89
------------------	------------------	------------------	---	----

LLC0

SLR

CLR

LALR → most
 complex
 use this

POA

Input

so table use
 freeze → we
 passing
 we are
 doing.

Stack

$$W \cup M \cup C \\ = S \\ = M \cup C \\ =$$

$$W \cup M \cup C \\ = S \\ = M \cup C \\ =$$

13 Feb 2024

Tuesday

[lec-11]

(Attribute grammar)

{ Int i; float p = 10.2; } → is it valid in C ??

↓

Java won't allow

↑

Illegal

↑

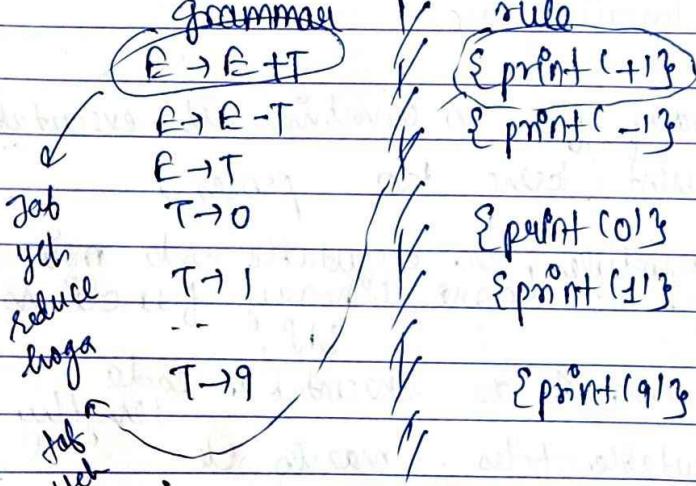
this check happens at semantic analyzer, not syntax analyzer.

Yes! can write!, but not

correct & dropping out some value.

for C → there is no attribute grammar.
some languages → uses " "

Attribute grammar :- SOT scheme. (Syntax directed Translation)



Job
yeh
reduce
hoga
print hoga!
→ Job + dekhogi → print hoga → chahiye production
use huiyi ho yaa na!

WJMk = SMM

SDD (syntax directed definition)

$$E \rightarrow E + T \quad (E \cdot \text{code}) = (E \cdot \text{code} //$$

↓ $T \cdot \text{code} // '+'$ production over

logi tabhi

onle

execute

hoga

(essa nahi
hoga in
SPT).

- ① All variables must be declared b4 they are referred.

These both are different attributes

Knuth (1968) → attribute grammar.

Predicate function = ?

Syntax rule : $\langle \text{expr} \rangle \rightarrow \langle \text{var} \rangle$

Semantic rule : $\langle \text{expr} \rangle \cdot \text{actual_type} \leftarrow \langle \text{var} \rangle$.

actual_type

Predicate : $\langle \text{expr} \rangle \cdot \text{actual_type} ==$

$\langle \text{expr} \rangle \cdot \text{expected_type}$

int a; float b;

means $a = b$ is valid. (tab ki vali hai)

$a \cdot \text{int} == b \cdot \text{float}$

$a \cdot \text{int} == b \cdot \text{float}$

↑ as not true!

so won't go!

so throw error!

eg : $\langle \text{var} \rangle \rightarrow A | B | C$

{ $\langle \text{var} \rangle \cdot \text{actual_type} \leftarrow \text{loop-up}(\langle \text{var} \rangle, \text{string})$

here no predicate required as declaring only variables that can be of any type, koi assigning wera nahi ho sake hai!!

WJMK

Translation / gen. code $i^o \rightarrow$ also done by semantic analyzer.

C statement :-

for Expr 1 ; Expr 2 ; Expr 3
 {
 ...
 }

meaning :

Expr 1;
loop : if Expr 2 == 0 goto out

Expr 3;

goto loop

out : ...

Example :-

switch (op) {

Case '+' :

{ return a + b;
 break; }

Case '-' :

{ return a - b;
 break; }

Case '*' :

{ return a * b;
 break; }

Case '/' : { return a/b;
 break; }

WSMK = $\sum_{i=1}^n M_i$

- ① In instruction set \rightarrow we have only goto, no switch,
no for loop
 \hookrightarrow so change accordingly.

if $a ==$ goto A
 if $a == b$ goto C
 A : add a b
 goto D] Jump table
 B : subtract a b
 goto D
 D :

- ② Denotational Semantics :- \rightarrow (applying maths to bring value)

Eg :- $\text{int} \leftarrow \text{bin_num} \rightarrow \text{binary}$ \leftarrow corr value.

(bin. num) \rightarrow '0' |

| 1

| <bin-num> '0'

| 1 —————— '1'

$$M_{bin}('0') = 0$$

$$M_{bin}('1') = 1$$

$$\begin{aligned} M_{bin}(<\text{bin_num}>'0') \\ = 2 * \end{aligned}$$

$$M_{bin}(<\text{bin_num}>)$$

$$\begin{aligned} M_{bin}(<\text{bin_num}>'1') \\ = 2 * M_{bin}(<\text{bin_num}>) + 1; \end{aligned}$$

①

$$\text{sum} = 2 * x + 1 \quad \{ \text{sum} > 1 \}$$

o/w
problem in
syntax

\hookrightarrow but we
tak nahi dia.

For that kya precond chahiye??



0 > 0

\checkmark \rightarrow so yeh gaaya!!

\hookrightarrow after this \hookrightarrow this should satisfies.

WOMK = $\frac{S}{M}$

even,

$x > 1 \vee$

$x > 10 \vee$

$x = 1 \vee$

so it is weakest pre-condition.

so for everything,

need pre & post condⁿ,

? need?

If $(a == b) \vee (b == c)$ \rightarrow difference

If $(\underline{a == b}) \wedge (\underline{b == c})$

→ yeh true → tab yeh nahi dethe hoga!!

→ precondition ??

① $x = x + y - 3 \quad (x > 10)$

↳ precondⁿ $\rightarrow x + y > 13 \Rightarrow \text{or } y > 13 - x$

② $\{x = A \text{ AND } y = B\}$

$t = x;$

$x = y;$

$y = t;$

$\{x = B \text{ AND } y = A\}$

for such purposes
we go for
Axiomatic
analysis!

③ Variables:-

↳ case sensitive ho yaa nahi?

↳ how chalate hoga?

'km', 'KM' \rightarrow giving same ans

↑

black box easy !!

across 2 char think $\Rightarrow (26)^2$ possibilities again case sensitive nahi
o/w $(52)^2$ possibilities.

WJMK = $\frac{C}{T} \sum m^k$

- Want to have reserved ~~to~~ words ??

↳ Yes!

↳ confusion ↓,
problem ↓.

But ↑ bhi nahi hone chahiye !

↳ restricts end user !
as ↑ it

- ② Can't remember IP address ; so use domain Name.

$a = b$

$l\text{-value} = r\text{-value}$

↑
address
(memory location)

↑ also memory location,
but can be constant
also).

ispe jo value
hoga !!.

- Need of Data type?

↑
taaki pachle ske kiske katni' memory

↑
(to reserve address)

size decided by instruction set
only !!

not 32

↑ can declare size also in CPP.

WJMK
= $\frac{e}{5}$
= 0.2

Name Admrs :-

C-based

↳ Rose, ROSE, rose ← affects readability.
So,

Case sensitivity \Rightarrow affects readability

① Java \rightarrow parseInt \rightarrow affects readability.

② Difference b/w structure and union?

↑
Adv-38
↑
space 😊

③ can make * as division ~~operator~~ operator, is my language?

↓
Yes!

} As Go will do BINDING

↓
of * with division operator

done \rightarrow
~~when~~
at design time (not binding)
binding (compile time or execution time binding)

int a;
(explicitly telling that it is of int type)
(implicit)
all are static binding
\$ a; if 1st symbol is available, then variable is of int type, no need to write int.
a = 10;
int only for 10
or based on inference

W3MK
= $\frac{C}{S} \cdot M \cdot M$
=

Dynamic binding happens at runtime.

C# → 1st language → which gave 'dynamic' name.

list = [10.2, 11.2] → vector
list = 47 → scalar

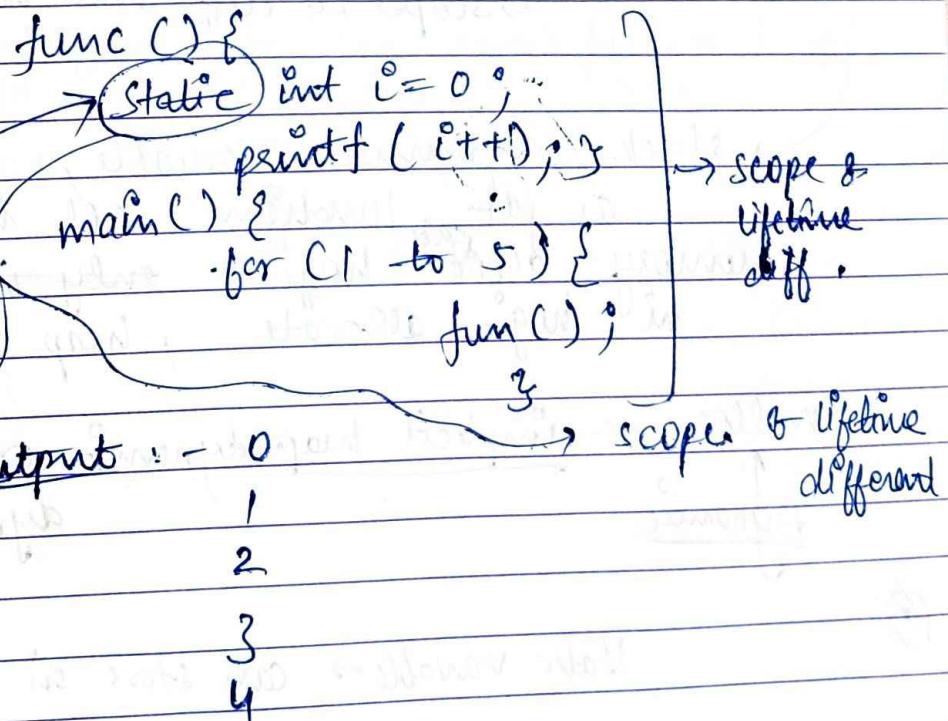
at one pt. of time it is float list

other time, it is int

so, execution time deciding kaunsa data type hogya

thus dynamic binding.

Storage Binding :-



WJMK
= S M =

① func() { int c; }

int main() {
 int a;
 func(); }

Scope & Lifetime
same.

Here, lifetime & scope
are same or not??

Where it

can be used?

Kab tak memory
allocated rhegi ??

② Static Variable | Dynamic Variable = ??
Ke ed

global, & static

↳ scope ka difference hai

→ stack dynamic variable,
as jab ~~call~~ function call hogi, tabhi
memory allo~~cate~~ hogi & only in stack mein
hi hogi allocate, heap mein nahi.

malloc ↲ Implicit heap dynamic or Explicit heap
dynamic ???
Dynamic

③ static variable → can store in stack but not
in stack dynamic.

W_W M_W = $\frac{e}{\sum m_i}$.

function big() {

 function sub1() {

 var x = 7; function sub2();

 } function sub2() {

 var y = x;

 var z = 3;

 var x = 3; function sub2();

 }

 y = 3

as yeh

hai immediate
parent

- overflow, underflow ke lyse kya pre cond'n / post cond'n useful? (H/W)
what use AXPomatic analysis ??

WDMK
= 30
= 100

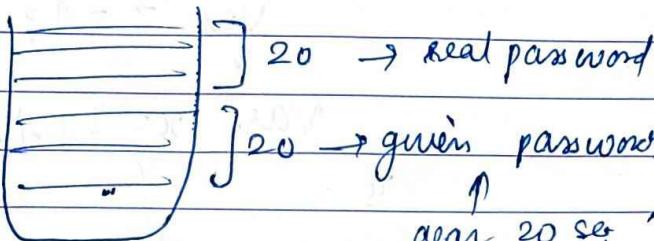
14 Feb 2024
Wednesday

[lec-12] :-

{ vi¹⁰ file.

gives all extension
with files of
file

{ given password [20];
real password [20];



agar 20 se char de diye
(as C has no provision
to check that).

toh real password will
change, will be overwritten.

so very careful while taking char as
input.

if we swap

these two times, then

real password over write na hoga

hoga, as always go upar ko, toh
kisi aur cheez ko impact kar dega.

↓

scanf, gets → both have problem!

so, fgets → use /cro! → as it has size as one of
parameters.

→ gcc provides provision to see assembly code.

[gcc file.c -S
vi file.s]

WORK C
DATA

Data Types \Rightarrow

`char c[20];`

at compile stage, what things to be maintained??

int

↳ only size ka dhyaan skho!

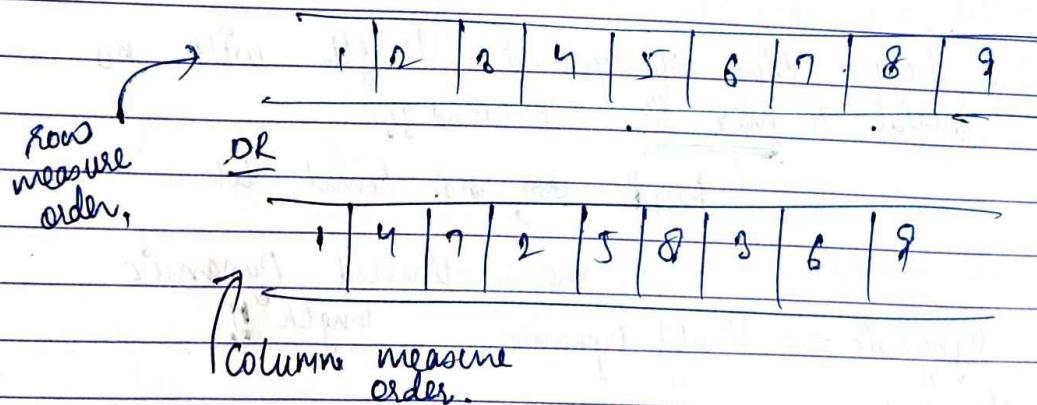
- ↳ string
- ↳ continuous allocation.
- ↳ starting address.
- ↳ maintain length also
- ↳ " current "

↑ agar <20 huya toh!

- Row / column measure order?

1	2	3
4	5	6
7	8	9

Storage is horizontal tape. only :-



\rightarrow C mein konsa use hota? (R/W) (Prove Also!)

int a[3][3]; \rightarrow if works \Rightarrow then row konsa and column konsa??

Pnt a[1][3]; \rightarrow " , , ,

\rightarrow which one is better??

WEEK
= 5
= 100

Q) Array \rightarrow static

means at compilation time only you know the size.

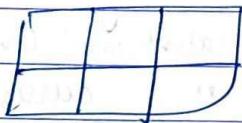
\rightarrow Fixed stack - Dynamic

\rightarrow Fixed Heap - Dynamic \rightarrow malloc

\rightarrow Heap Dynamic

Even size also not decided yet.

Rectangular array :-



Jagged array :-



Q: Given float number, check whether it is true or false?

\hookrightarrow IEEE \rightarrow float tedious.

\hookrightarrow can do it with 'int' data type !!

Q: want to declare strings of dynamic length with no initial & max size declared ??

A:

\hookrightarrow hog \Rightarrow no limit to h

means limited dynamic

Dynamic == Limited Dynamic length !!

string a;

string a[20];

max_length

max_length = 20

\Rightarrow go limit hogi
use data type ki!

- linked list (no max limit)
- should language supports pointer type or reference type or both?

①

Dangling Pointer :-

a heap pointer that contains address of dynamic variable that has been deallocated.
 ↓ also case of memory leakage!

`int a = 2, b = 3; → No side effect!`

`c = a + b;`

a, b ke' value same rhe'

but yeh bahi gya

so side effect !!.

②

`int a = 5;`

`int fun1() {`

`a = 17;`

`return 3;`

`}`
`void main () {`

`a = a + fun1(); } → but this is side effect!`

as changing !!.

5 3

WORK :-
3Mm.

20 Feb 2024

Tuesday

Lec-13 :-

Sub programs :-

- How to prove that 'c' follows row-measure order ?

Code :-

$\text{int } a[2][3] = \{1, 2, 3, 4, 5, 6\};$

↳ error → Incomplement ~~any~~ element type.

↳ $a[2][2]$

$a[2][3]$

↑ now no error !!

so row based.

↳ as then no. of columns define kene pdhle hain abhi.

① Row based better or column based ??

② which lesser is better ??

③ Why happening that

$i == 22 \cdot 3$ false!

$i == j$ true!

kya yeh store keke
compare mati pr raha?
to kya raha hai ???

EXPRESSIONS :-

~~WORK~~ ~~STUDY~~

int a = 5;

int fun1() {

a = 17;

return 3; }

void main() {

a = (a) + fun1();

};

a pehle hua,

but fir

6h

change

hoga!!

answer : 8

pointing \rightarrow [8] \rightarrow 17. \rightarrow so ans = 20?
or 8?

a

(20)

here it is
global
variable!!

(Side effect happens !!)

when Evaluating expression :-

Precedence, Associativity, parenthesis &
operand evaluation Order,
 \uparrow
all 4 are important !!

① Transparency :-

$\rightarrow \text{result}1 = (\text{fun}(a) + b) / (\text{fun}(a) - c);$

$\text{temp} = \text{fun}(a);$

$\text{result}2 = (\text{temp} + b) / (\text{temp} - c);$

same result! but agar kaijii side effect
happens \rightarrow then not same!!

WEEK =
= 5 MM

① Enumeration Types :-

enum week = {mon, tue, ..., sun}
↓ ↓ ↓
0 1 6

enum month = {tue, wed, ..., }
↓ ↓
0 1

what is value of tue?

acc. to week, it is 1,
— month, — 0.

according to Python

↳ which will take second one!
↳ so (0).

if want both to be used!

↳ so

↳ like week, tue, month • tue.

UNION TYPES :-

- Can use UNION as alternate for structures

↳ NOTE:

{int a;

char c; };

a = 1.2 → allowed in C, in case of

UNION

↳ problem!

② UNION PUNNING

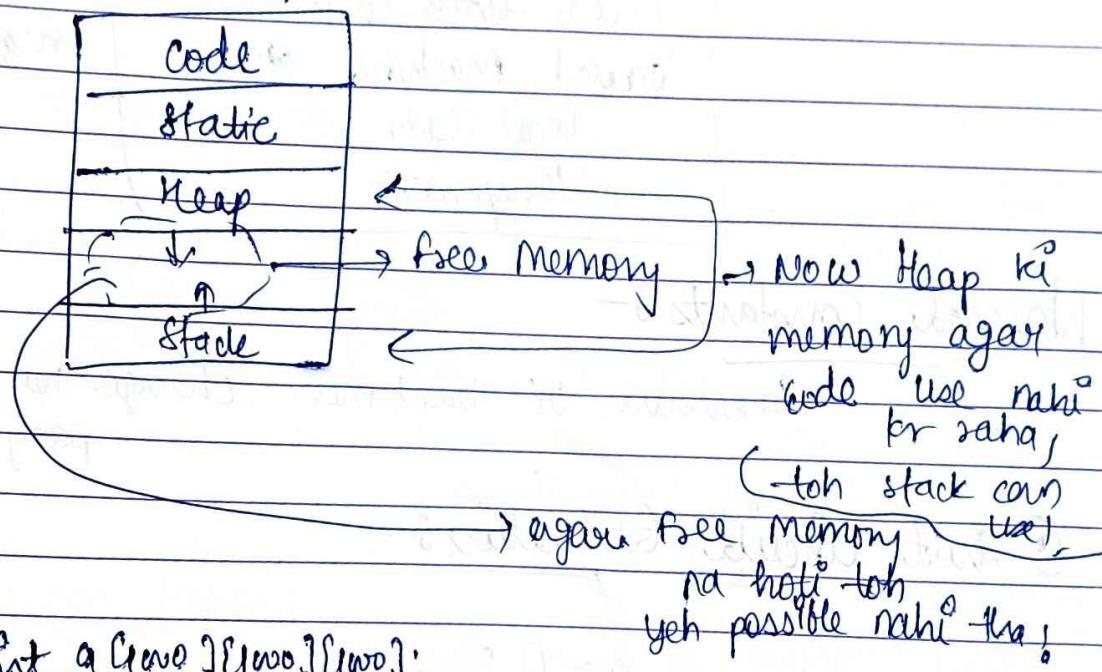
- ↳ complex data-type conversion
(like float) se nkt to check whether

WORK
= 50%
DO

positive or negative?)

↳ as much fast!

Runtime Memory:-



not a C program;

↳ gcc → compile time no error,
run time → only segmentation fault.

as compiler not aware of it,
→ when executing, only then
allocating space → then coming to know
of Segmentation fault.

① Main fun → calling sub fun,
↳ which goes, do work &
comes back!

↳ then also what things to be maintained?
② instruction pise oaks execution pise start
hogi ⇒

$$W_{MM} = \frac{C}{P_{MM}}$$

Actual Parameters

Returned values

control link :-

Access ~~link~~ link

saved machine status

local data

temperaries

→ yeh
maintain
kera
pdhega !!.

Named Constants :-

Generates ki baad mein change na ke
paayein !!.

① Short Circuit Evaluation :-

$$a = 0;$$

$$a * b * c * d.$$

→ don't want to process it,
as '0' hi dayega
answer.

so short circuit

$$a * b * c * d + f$$

→ just add this

↑ skip this

$$\text{if } a == b \text{ & } a == c$$

→ if this is true → only then will
go here

so short circuit !!

WJMK
=
SMMO
=

Subprograms :-

- ↳ use of no. of lines reduces,
Scalability ↑.
- procedure → can do side effects indirectly!

① difference b/w procedure & function ??

↳ return something,

↳ does not return!
→ Then why procedure??

- Each subprogram has a single entry point.

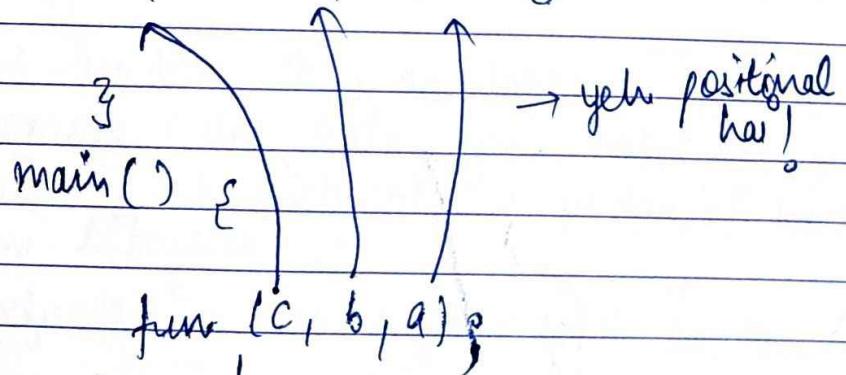
so fn → has only 1 entry point,
but can have many exit points.

② void fn is procedure!
↳ JIO base kaam kare de!!.

② Parameters Profile :-

No, order, types of variable ↑ should be maintained.

int func(int a, int b, int c) {



} }

but
want

c → a ke paas
jaye!

→ No syntax

Order !!

a will get value of c.

~~WORM~~
~~IN~~
~~MM.~~

Parameters :-

① Formal

② Actual

③ Positional

④ key words

sum, (length = my_length,
list = my_array)

FUNCTION :-

global, pass by reference \rightarrow also side effect beta,
not only for (return value)

procedure can also do side effect
Indirectly !!

WJMK
2024

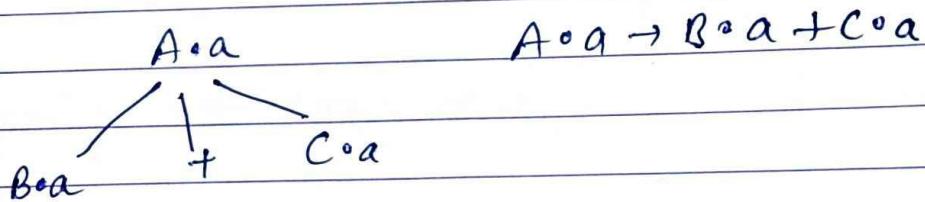
21 Feb 2024
Wednesday

Lec-14 :-

Attribute grammar: (Important)

synthesized, inherited
if you give if your parents
money to parents
give money to you.

Q) What is difference b/w syntax tree & Parse tree?



$A \cdot a = B \cdot a$
↳ synthesized attribute

$B \cdot a = A \cdot a$
↳ inherited attribute.

Q) S-attribute, L-attribute, synthesized, inherited.
↳

can take
money from
parent as

well as
elder sibling.

$B \cdot a = A \cdot a + C \cdot a$

Pseudo random generator

WJMK 3
3
3
3

function → single entry point, multiple exit points.

multiple entries

↳ in PYTHON.

↳ but have to use specific keyword,

co-routine

↳ fn has multiple entries & 1 or more exits

① main main variable → no scope in fn

↳ but some have (as maine

billaya, toh executing in my thread only,
so can use my variable).

↳ toh aap kya want? decide!

② rand() se fissa generate kiye → must be unique!

How to remember?

① main main array rkhlo to compare with previous.

② go for static keyword if want to remember history.

③ agar return type nahi milta in function, by default kya hota?
and agar

④ Pass by reference

↳ two different variables pointing to same, ek mem change will effect other as well.

caller (a, b, d) ??

caller (d=a, b=b, c=c) ??

yeh yhaan aayega ??

W3MK 

Parameters Passing :-

- ① Pass-by-reference - Result
- ② Pass-by-name - Result

(Pass-by-dictionary \Rightarrow Pass-by-reference)

Q: (Pass-by-function

↳ will it be = to Pass-by-reference ??

- ③ ad-hoc binding (Avoid it)

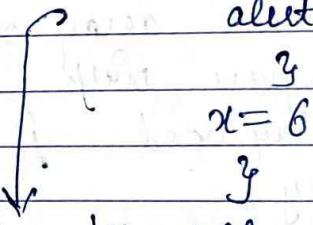
↳ ques of readability comes

↳ can't tell output as pta nahi
kisi value it will take.

fun sub1() {

 fun sub2() {

 alert(x);



↳ if we'll call sub2() \rightarrow '6' will be printed.

(C does not support this kind of use)

↳ kya variable, fir declare)

- ④ calling subprogram indirectly:-

↳ Event based (User activity based)

⑤ + itself calls ??

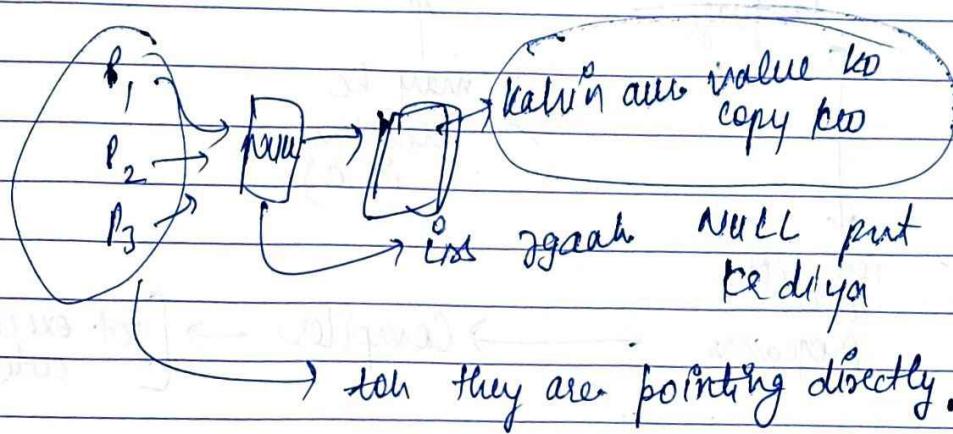
- ⑥ 3 pointers pointing to same location & u want to deactivate the location

↳ for that put all pointers = NULL.

WJMKE = $\frac{e}{\lambda}$
= $\frac{\lambda}{e}$

Will the user follow this?

as esse ho sata bhet saare
pointer or user kisi ko (= NULL) kena bhet jaaye!



① Garbage collector ko bhi list hao to maintain !!

② When calling fn inside another fn \rightarrow
It goes like Activation Record !!

Q: which abstract data types to be maintained?

③ a, b, c = fun() \rightarrow whether 'c' supports this ??
 \hookrightarrow Python supports \hookrightarrow NO!

CLOSURE: - (not in C, as 1 fn can't use variables of
other, while calling it)

\hookrightarrow So yeh Akhna hai ya nahi decide bt
designing language.

WJMK = ISOM,

- ① gcc written in C and use compiling C.
so same language. so can compile that language.

