Lec 04/04

$S \to Ac \mid Bc \qquad A \to aA \mid a \qquad B \to bB \mid b$

Recursive descent

- Procedure for each non-terminal
- match against each terminal.

$FS(S) = \{a, b\} \qquad FS(A) = \{a\} \qquad FS(B) = \{b\}$

lookahead = next char to be produced
match('x') ⇒ if 'x' == lookahead then we backtrack. if equal then we increment the lookahead.

non deterministic search

```
S()
if (l='a')
  A()
  match('c')
else if (l='b')
  B()
  match('c')
```

```
A()
if (*)
  match('a')
  A()
else
  match('a')
```

```
B()
if (*)
  match('b')
  B()
else
  match('b')
```

aaaac
↑ ↑ error ⇒ backtrack.

if (*) ⇒ both productions can make the lookahead ('a') so we put a * that allows backtracking. So we first explore one production, if successful then correct, else backtrack.

① First Set (S)

(1) First Set (S)

(2) Follow Set (N) — set of terminals that can appear just after that non terminal has finished producing.

$S \to Ac \mid Bc \mid a$
$A \to aA \mid a \mid \varepsilon$
$B \to bB \mid b$

follow (A) = {c}
follow (B) = {c}
follow (S) = ϕ

$S \to Ac \mid Bc \mid a \mid Sc$
follow (S) = {c}

empty string / eoS $\to$ $

$A \alpha B \Rightarrow$

follow (A) = { (α = terminal) ? α : firstset(α)
(if α is nullable) then union with first set of B ... }

Fixed Point Solution

$x = f(x)$ recursively we apply f to x and if we get to some $x^* = f(x^*)$ then x^* is the fixed pt sol^n for the eq^n.

(x defined in terms of f(x))

Parse Table

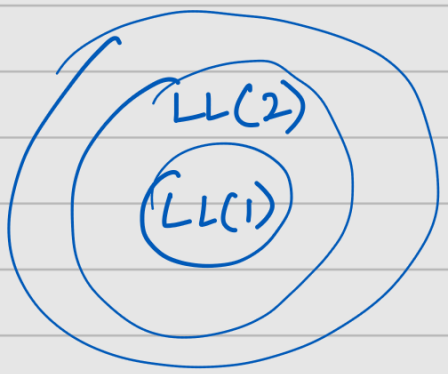
any cell having no entry $\Rightarrow$ that lookahead can't be formed from that non-terminal $\Rightarrow$ error

any cell can't have more than 1 entry

2 or more entries ⇒ backtracking ✗
we want 1 (only production which we can use & all possible strings can be produced and we don't need backtrack)

LL(1) Grammar — with a single lookahead
↑ — using left most derivation
inputs left to right lookahead

inputs left to right lookahead
using left most derivation with a single lookahead



CFG

LL(2) > LL(1)

LL(1) grammar — grammars for which parse table can be constructed.

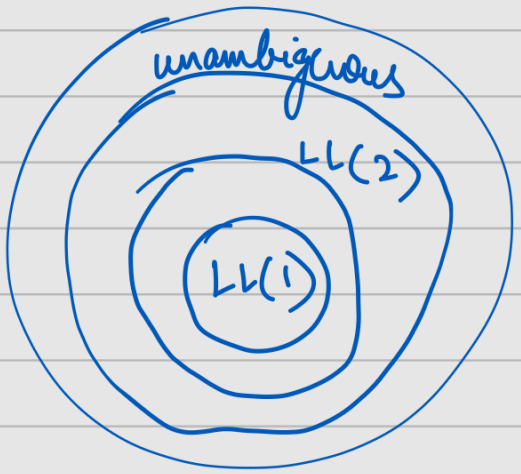
↳ each cell has ≤ 1 entry

The entry allows all correct possible strings to be produced & any not possible string then u get stuck at 0 entry pt.
✓ NO Backtracking needed.

	a	b	c	$ (eos)
S		P_1		
A	$A \to aA$ $A \to a$ } conflict			

B	

Conflict
- (1) First - First conflict
- (2) First - Follow conflict



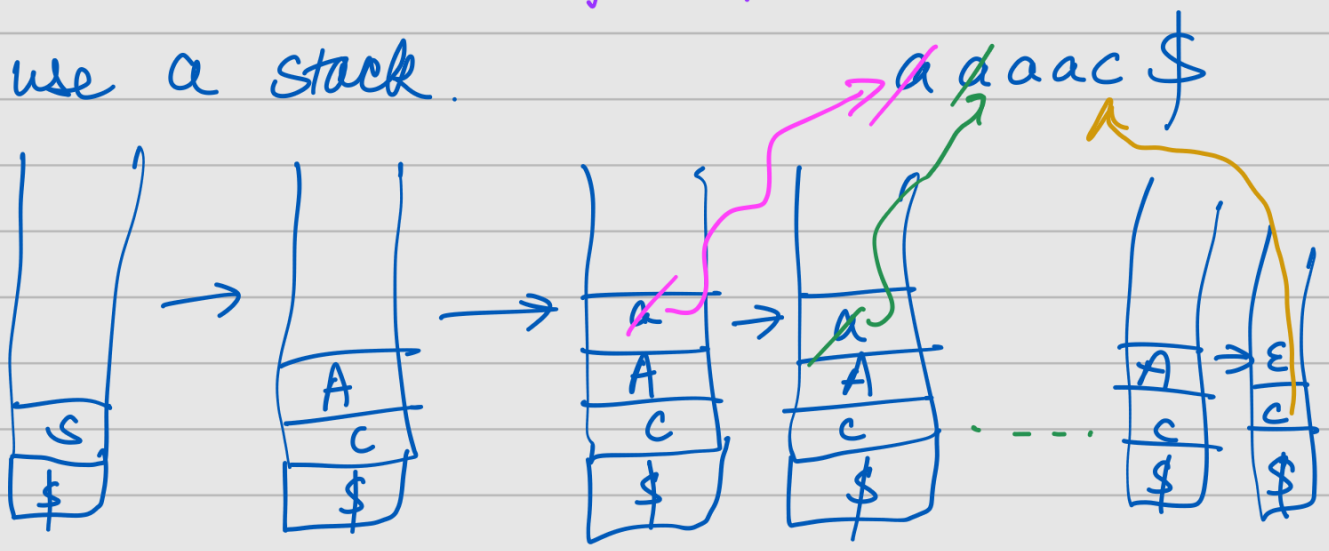
CFG

No ambiguous grammar parse table can be constructed

But not all unambiguous grammars are LL(1) i.e. can give conflicts.

To construct parser from parse table

use a stack.



Problems:

1) Left recursive
2) 2 productions of a non-term have the same first terminal — Left factor

$S \rightarrow$ if cond then S else S | if cond then S

left factor

left factor

(can take factor common out)