Compiler

12 NOV- 23 6:25 PM

Compiler Design:

- O Lexical D. Analysis
- @ Pariser (Syndax Analyzen) **
- @ Gemantie Analysis
- (9) Intermidiate Code generator
- @ Code Opimization

Phases of Compiler Game for platforms HLL Characteris Lexical Analysis Syntax Analysis Symbol nelbrich Toble Parise Tree Sementie Analysis Interremediate Code General 3 address code Code Optimization Bucken Porrat Code Generation Hyemply Code/Machine Code (111)

	neliomo)
82 - M. M. D. C.	
Lexical Analysis [1	exem, Tokemizem, Seamen
	Compiler Design :
1 Tokenization	5 Exceeding length
2) Give Emnon means	Ages Speeding length Sillegal Chanaden
@ Fliminate Comerals, U	CMIE SPORC
Talendal	(Jab, new space, new line)
	emotion > Special Chanadere
keywond	-> Constants
Identifica Sependon	ansold
Lecture-5	11th Combine to the
Find Pinst () in Comp	silen Design
Firest (A) Contains all	learninals paresent in finst
place of every stains	derived by A.
1) 5 -> @bc/ (d)ef/ (ghi	interinediate Con
2) First (Terrminal) = terr	ninal
3) First (E) = E	months of the first
	DAY /

naligno)

S \rightarrow ABC (a,b,c,d,e,f,e) A \rightarrow 0/b/e (a,b,e) B \rightarrow c/d/e (c,d,e) C \rightarrow e/f/e (e,f,e)

E-> AaAb Bblex

1,6.9

上の二個別

algono.

Follow (A) contains set of our terminals priesent immediate in right of 1A1.

* follow neven combain &

 $S \rightarrow ABC$ $A \rightarrow DEF$ $O \rightarrow C$ $C \rightarrow C$

Ledwer OB

What in Parising & Types of Parisens:

Porisers

Top Down Parson

Recursive

LL(1)

Decent

LR(b)

LR(c)

LR(c)

LR(d)

Parising: It is a priocess of deriving string.

Ledove-06 Whole in Ameny & typen of foresure: derive id := id + roeal - integers; assign_stmt \rightarrow id:= exp π ; Top Down Pouson →id:= expr operation term; := Expri opendon term opendon term; := - terim a benopui real which to too southed of I is given id tied _ integer

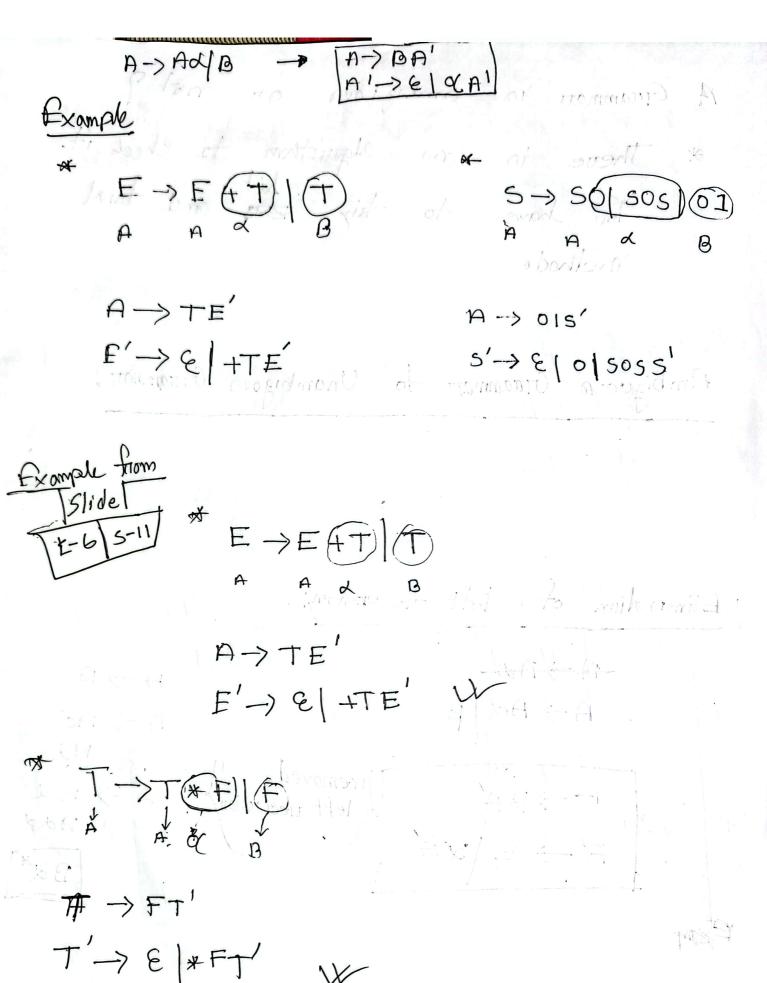
A Grammari is ambigious on not there in no algorithm to check it. You have to tay methodo

Ambiguous Gramman to Unambigous Gramman:

Elimination of left reconsion!

A > Ad | B

left trecursion A-> Ad



Predictive Pousen (IL Top Down Patrising Reconsive - Descent Parising -5-> Aa/Ce Input: aabbée A-> aaB/ aaba add Ed e→ ao D D -> ppg

Priedictive Pariser $\{LL(1)\}$ $E \rightarrow TE'. \qquad \{id, (\} \qquad \{\$, \}\}$ $E' \rightarrow E|+TE' \qquad \{\ell, +\} \qquad \{\$, \}\}$ $T \rightarrow FT' \qquad \{id, (\} \qquad \{+, \$, \}\}$ $T' \rightarrow E|+FT' \qquad \{\ell, *\} \qquad \{+, \$, \}\}$ $F \rightarrow id \mid (E) \qquad \{id, (\} \qquad \{+, \$, \}, *\}$

	M	id	+	*			\$
	E	E→TE'			ESTE	y da 1 S	
	E		E >+JE'	2		£/→&	£'→€
1000	丁	T→FT'		> - 4 x	7->FT		
	丁/		+'>E	T->*F+		7-36	T-1>6
	F	F-> id	\\\\\		F->(f)	. /	

Friday

Stack	impot	production
\$ET+	(id#id)+id\$	ENTE
\$ E'T	(id*id)+id\$	TOFT
\$ETF	(id*id) +id \$	F → (F)3+
	(id*id) tid\$	POP T'3
\$ = 'T') E	id * id) + id \$	ESTEN
\$E'T' \ E'T	id * id) + id \$	本作行名
\$ E'T') E'T'F	id * id) + id\$	F > ind
\$E丁ンE丁id	id* id) + id\$	POP 19
\$ E'T') E'T'	-* id) + id\$	T->*FT
SE'T'DE'T'F*	+ id) + id\$	POP
\$ E'T')E'T'F	\$ bi+ (bi	$bi \leftarrow \exists$
\$巨丁'J巨'丁' id	id) tid\$	POP
\$ 巨丁') 巨'丁')+id \$	7->6
\$ E'T') £')+id\$	\\ \tau \\ \> \e
\$ E'T')) +id \$	909

Stack bong Podeton Input \$ 5-1 7428 + id \$ () 至 >+ 7里 +id\$
\$ bit (bi * bi) \$E POP \$ E'T+ _ -++id \$61-461) \$ 6 id \$ 61 * 60 \$ ET 909 **月** \$ E'T'F + bid \$ bit to bi F>id \$ bid \$ (6) * /; \$ETT id POP 4 ETT & POP POP 1 6 \$ (51 x 51 \$ P1909 161 \$(61 x6) Arrepted #Lot (Lix \$79x (-) 7 'T'a('T'a) · / 9 / \$ 66 + (11 * *4, 1, 15 (L) 34 Li <- 7 £ bit- (21 ヨーコーコードード 9.09 f-1014 [/1] らう ヤマモイヤヨト 4- 61-4-6 (FT) E'T' 3 4 7 13 ('T') 1 909