Compiler Design Laboratory (CS 753)

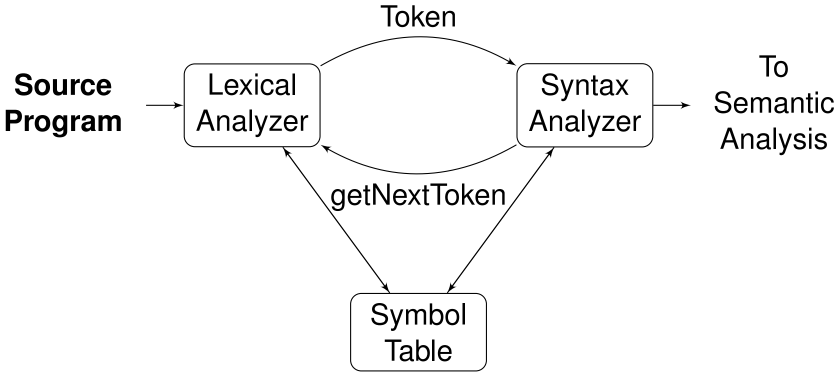
Samit Biswas

samit@cs.iiests.ac.in

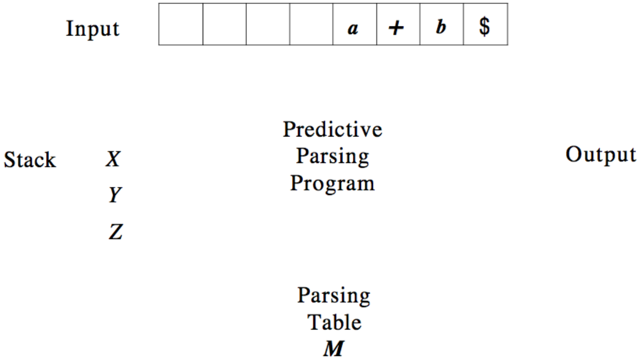
Department of Computer Science and Technology,
Indian Institute of Engineering Science and Technology, Shibpur

August 27, 2018

Syntax Analyser



Model of a Table driven Nonrecursive Predictive Parser.



Input

				a	+	*b*	$

Stack

X

Y

Z

Predictive
Parsing
Program

Output

Parsing
Table
M

Consider the following Grammar

$E \rightarrow TE'$

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

$T \rightarrow FT$

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

$F \rightarrow (E)|id$

Construct the Predictive parsing table.

FIRST(E) = FIRST(T) = FIRST(F) = $\{($, id$\}$
FIRST(E’) =$\{$ +, $\epsilon\}$
FIRST(T’) =$\{$*,$\epsilon\}$
FOLLOW(E) = FOLLOW(E’) =$\{),\$\}$
FOLLOW(T)=FOLLOW(T’)=$\{+,),\$\ \}$
FOLLOW(F) = $\{$*, +,), $\$\}$

Table: Predictive Parsing Table

Non Terminal	*id*	$+$	$*$	(	)	$\$$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E’		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T' \rightarrow FT'$		
T’		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Nonrecursive Predictive Parsing

1. If $X = a =$\$, the parser halts and announces successful completion of parsing.
2. If $X = a \neq$\$ the parser pops off the stack and advances the pointer to the next input symbol.
3. If X is a non terminal, the program consults $M[X, a]$ of parsing table M. The entry will be either an X-production of the grammar or an error entry.
   For example, If M[X,a] = $\{X \rightarrow UVW\}$, the parser replaces X on top of the stack by WVU (with U on top).

Table: Parsing: id+id*id

Stack	Input	Output
$E	id+id*id$	
$E’T	id+id*id$	$E \rightarrow TE'$
$E’T’F	id+id*id$	$T \rightarrow FT'$
$E’T’id	id + id * id $	$F \rightarrow id$
$E’T’	+id*id $	
$E’	+id*id$	$T' \rightarrow \epsilon$
⋮	⋮	⋮
$	$	Accept

Implement a predictive parser using C program.