Error recovery in predictive parsing

- An error is detected during the predictive parsing when the terminal on top of the stack does not match the next input symbol, or when nonterminal A on top of the stack, a is the next input symbol, and parsing table entry M[A,a] is empty.
- Panic-mode error recovery is based on the idea of skipping symbols on the input until a token in a selected set of synchronizing tokens.

How to select synchronizing set?

- Place all symbols in FOLLOW(A) into the synchronizing set for nonterminal A. If we skip tokens until an element of FOLLOW(A) is seen and pop A from the stack, it likely that parsing can continue.
- We might add keywords that begins statements to the synchronizing sets for the nonterminals generating expressions.

How to select synchronizing set? (II)

- If a nonterminal can generate the empty string, then the production deriving ε can be used as a default. This may postpone some error detection, but cannot cause an error to be missed. This approach reduces the number of nonterminals that have to be considered during error recovery.
- If a terminal on top of stack cannot be matched, a simple idea is to pop the terminal, issue a message saying that the terminal was inserted.

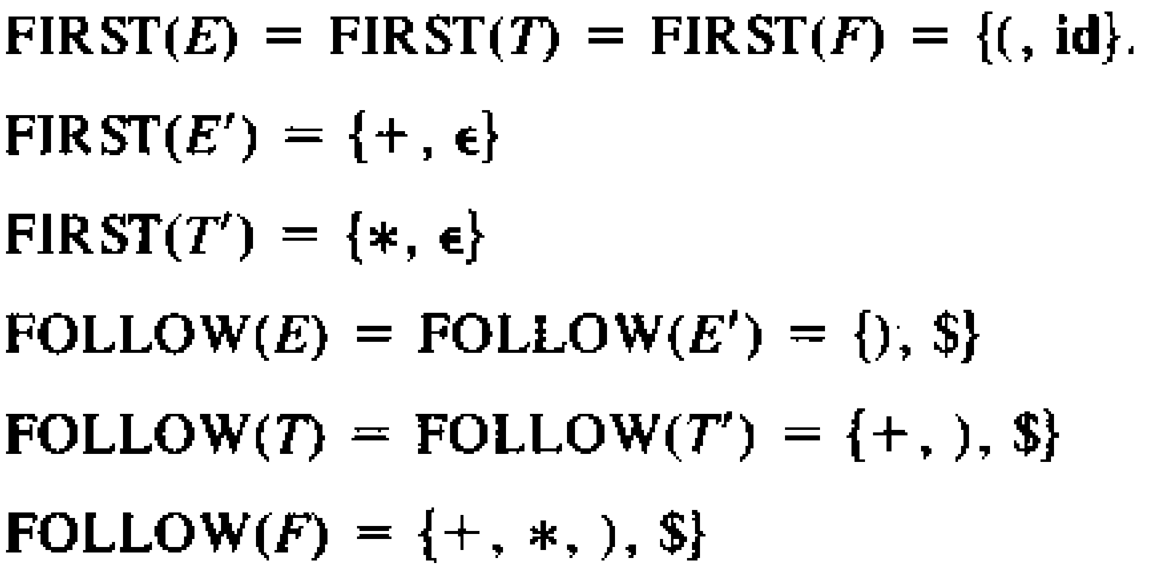
Example: error recovery

"synch" indicating synchronizing tokens obtained from FOLLOW set of the nonterminal in question.

If the parser looks up entry M[A,a] and finds that it is blank, the input symbol a is skipped.

If the entry is synch, the the nonterminal on top of the stack is popped.

If a token on top of the stack does not match the input symbol, then we pop the token from the stack.



$$\text{FIRST}(E) = \text{FIRST}(T) = \text{FIRST}(F) = \{(, \textbf{id}\}.$$

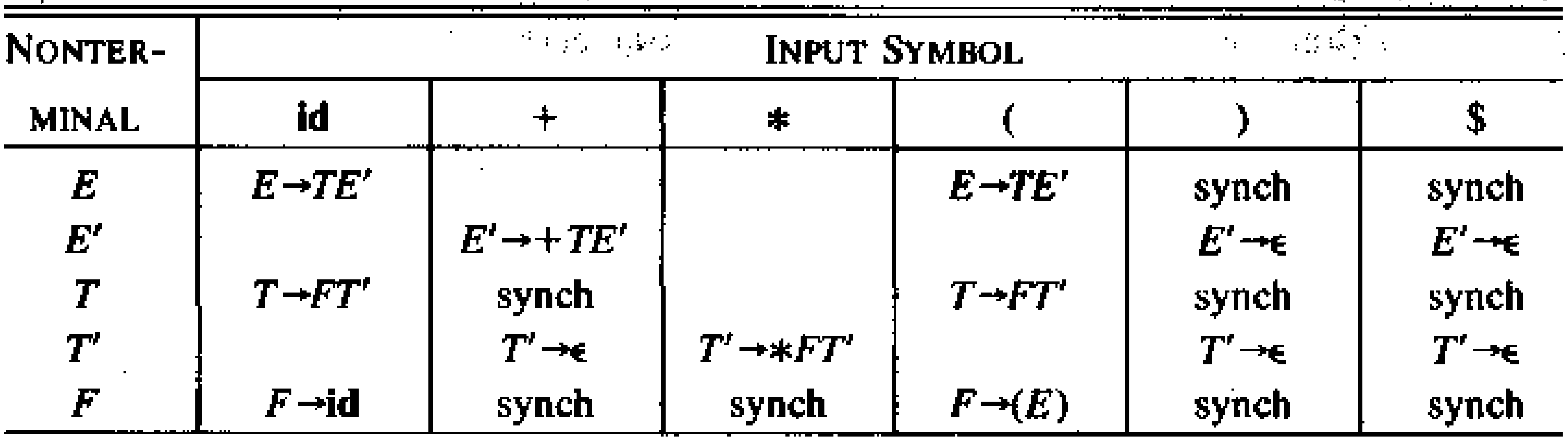
$$\text{FIRST}(E') = \{+, \epsilon\}$$

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

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

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

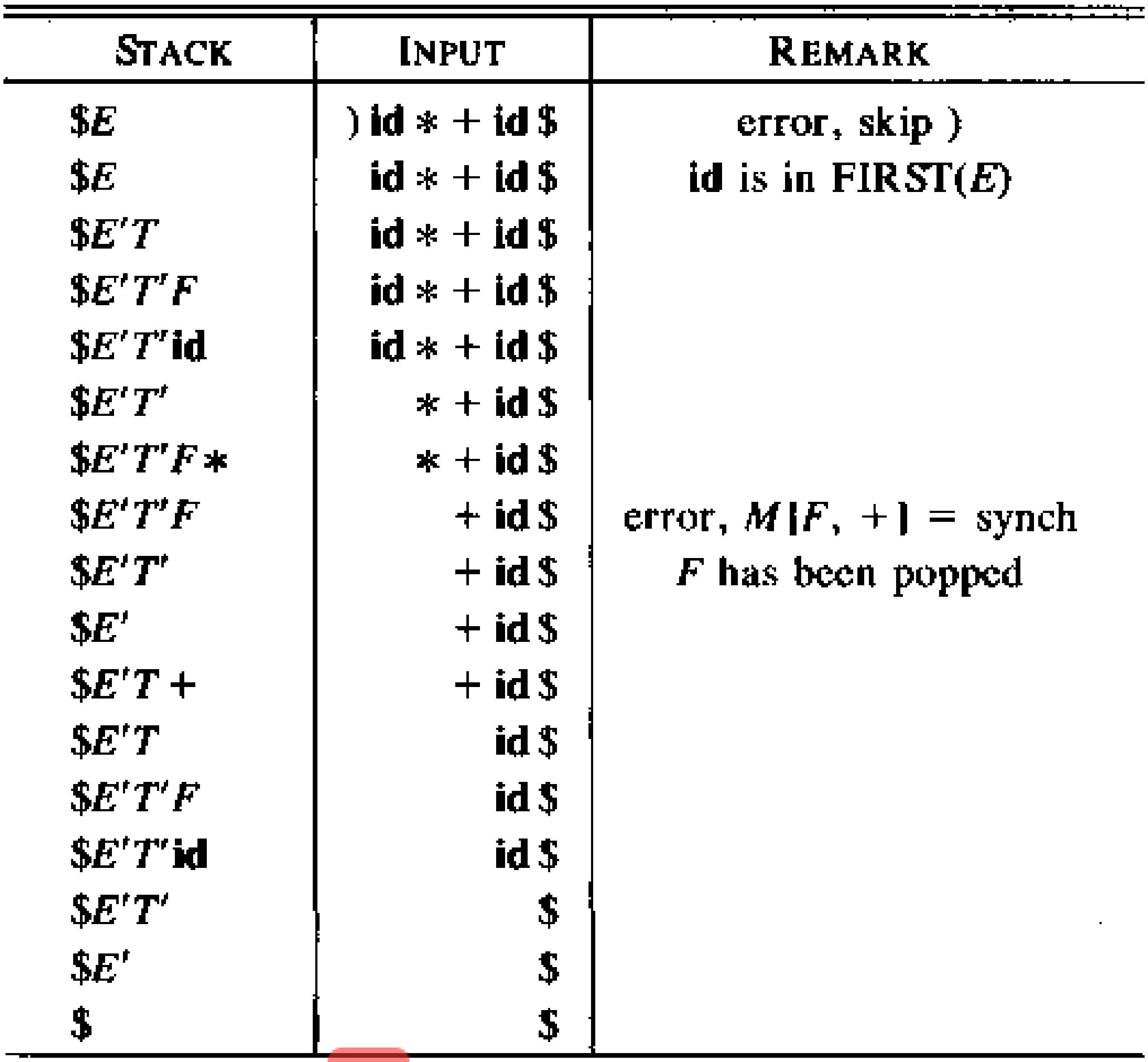
$$\text{FOLLOW}(F) = \{+, *,), \$\}$$



NONTER-	INPUT SYMBOL					
MINAL	**id**	+	*	(	)	$
E	$E \to TE'$			$E \to TE'$	synch	synch
E'		$E' \to +TE'$			$E' \to \epsilon$	$E' \to \epsilon$
T	$T \to FT'$	synch		$T \to FT'$	synch	synch
T'		$T' \to \epsilon$	$T' \to *FT'$		$T' \to \epsilon$	$T' \to \epsilon$
F	$F \to \textbf{id}$	synch	synch	$F \to (E)$	synch	synch

Fig. 4.18. Synchronizing tokens added to parsing table of Fig. 4.15.

Example: error recovery (II)



STACK	INPUT	REMARK
$\$E$	$)\,\mathbf{id} * + \mathbf{id}\,\$$	error, skip)
$\$E$	$\mathbf{id} * + \mathbf{id}\,\$$	**id** is in FIRST(E)
$\$E'T$	$\mathbf{id} * + \mathbf{id}\,\$$	
$\$E'T'F$	$\mathbf{id} * + \mathbf{id}\,\$$	
$\$E'T'\mathbf{id}$	$\mathbf{id} * + \mathbf{id}\,\$$	
$\$E'T'$	$* + \mathbf{id}\,\$$	
$\$E'T'F *$	$* + \mathbf{id}\,\$$	
$\$E'T'F$	$+ \mathbf{id}\,\$$	error, $M[F, +]$ = synch
$\$E'T'$	$+ \mathbf{id}\,\$$	F has been popped
$\$E'$	$+ \mathbf{id}\,\$$	
$\$E'T +$	$+ \mathbf{id}\,\$$	
$\$E'T$	$\mathbf{id}\,\$$	
$\$E'T'F$	$\mathbf{id}\,\$$	
$\$E'T'\mathbf{id}$	$\mathbf{id}\,\$$	
$\$E'T'$	$\$$	
$\$E'$	$\$$	
$\$$	$\$$	

Fig. 4.19. Parsing and error recovery moves made by predictive parser.