Error recovery. Worksheet for top-down, table-driven parser.

Algorithm:

- Use of parsing table:
  - If the parser looks up entry M[*A, a*] and finds that it is blank, then the input symbol *a* is skipped.
  - If the entry is *synch*, then either:
    - Pop the nonterminal on top of the stack or
    - Skip input until one in First(A) is found.
  - If a token on top of the stack does not match the input symbol, then we pop the token from the stack.

	Expr	Expr'	Term	Term'	Factor
First	(, id, num	+, -, ε	(, id, num	*, /, ε	(, id, num

Parse table:

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

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

Complete:

STACK	INPUT	REMARK
$\$E$	) **id** * + **id** $	

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