Ch5: Use LL(1) parse table

Content

There're a LL(1) grammar and parse table. Please use these to parse the following code. (write all the parsing steps(Stack,Input,Apply Rule))

S → TS′	1
S′ → +TS′	2
S′ → λ	3
T → FT′	4
T′ → *FT′	5
T′ → λ	6
F → (S)	7
F → id	8

Non-terminal	Input symbols					
	Id	+	*	(	)	$
S	1			1		
S′		2			3	3
T	4			4		
T′		6	5		6	6
F	8			7		

Input : (id+id)*id$

You have already answered the quiz.

Stack	Apply Rule	Remain Input
S		(id+id)*id$
	[1]S->TS'	
$S'T		(id+id)id$
	[4]T->FT'	
$S'T'F		(id+id)id$
	[7]F->(S)	
$S'T')S(		(id+id)id$
	Match	
$S'T')S		id+id)id$
	[1]S->TS'	
$S'T')S'T		id+id)id$
	[4]T->FT'	
$S'T')S'T'F		id+id)id$
	[8]F->id	
$S'T')S'T'id		id+id)id$
	Match	
$S'T')S'T'		+id)id$
	[6]T'->lambda	

Stack	Input
$S'T')S'	+id)id$
[2]S'->+TS'	
$S'T')S'T+	+id)id$
Match	
$S'T')S'T	id)id$
[4]T->FT'	
$S'T')S'T'F	id)id$
[8]F->id	
$S'T')S'T'id	id)id$
Match	
$S'T')S'T'	)id$
[6]T'->lambda	
$S'T')S'	)id$
[3]S'->lambda	
$S'T')	)id$
Match	
$S'T'	id$
[5]T'->FT'	
$S'T'F	*id$
Match	
$S'T'F	id$
[8]F->id	
$S'T'id	id$
Match	
$S'T'	$
[6]T'->lambda	
$S'	$
[3]S'->lambda	
$	$
Accept!	