

Compiler(Retro Basic)

Part Language Analysis

จาก grammar พบว่ามี left recursion การทำ LL(1) parser จะมีจุดที่ต้องแก้ไข คือ ที่ exp และ cond (ใช้ exp* และ cond*)

pgm := line pgm EOF	id is {A..Z}
line := line_num stmt	const is {1..100}
stmt := asgmnt if print goto stop	line_num is {1..1000}
asgmnt := id = exp	
exp := term exp*	
exp* := + term - term EMPTY	
term := id const	
if := IF cond line_num	
cond := term cond*	
cond* := < term = term	
print := PRINT id	
goto := GOTO line_num	
stop := STOP	

First set

$\text{First}(\text{pgm}) = \{\text{line_num}, \text{EOF}\}$

$\text{First}(\text{line}) = \{\text{line_num}\}$

$\text{First}(\text{stmt}) = \{\text{id}, \text{IF}, \text{PRINT}, \text{GOTO}, \text{STOP}\}$

$\text{First}(\text{asgmt}) = \{\text{id}\}$

$\text{First}(\text{exp}) = \{\text{id}, \text{const}\}$

$\text{First}(\text{exp}^*) = \{+, -, \text{EMPTY}\}$

$\text{First}(\text{term}) = \{\text{id}, \text{const}\}$

$\text{First}(\text{if}) = \{\text{IF}\}$

$\text{First}(\text{cond}) = \{\text{id}, \text{const}\}$

$\text{First}(\text{cond}^*) = \{<, =\}$

$\text{First}(\text{print}) = \{\text{PRINT}\}$

$\text{First}(\text{goto}) = \{\text{GOTO}\}$

$\text{First}(\text{stop}) = \{\text{STOP}\}$

Follow set

$\text{Follow}(\text{pgm}) = \{\text{EOF}\}$

$\text{Follow}(\text{line}) = \{\text{EOF}, \text{line_num}\}$

$\text{Follow}(\text{stmt}) = \{\text{EOF}, \text{line_num}\}$

$\text{Follow}(\text{asgmt}) = \{\text{EOF}, \text{line_num}\}$

$\text{Follow}(\text{exp}) = \{\text{EOF}, \text{line_num}\}$

$\text{Follow}(\text{exp}^*) = \{\text{EOF}, \text{line_num}\}$

$\text{Follow}(\text{term}) = \{+, -, \text{EOF}, \text{line_num}\}$

$\text{Follow}(\text{if}) = \{\text{EOF}, \text{line_num}\}$

$\text{Follow}(\text{cond}) = \{\text{line_num}\}$

$\text{Follow}(\text{cond}^*) = \{\text{line_num}\}$

$\text{Follow}(\text{print}) = \{\text{EOF}, \text{line_num}\}$

$\text{Follow}(\text{goto}) = \{\text{EOF}, \text{line_num}\}$

$\text{Follow}(\text{stop}) = \{\text{EOF}, \text{line_num}\}$

Rules set

- 1) $\text{pgm} \rightarrow \text{line pgm}$
- 2) $\text{pgm} \rightarrow \text{EOF}$
- 3) $\text{line} \rightarrow \text{line_num stmt}$
- 4) $\text{stmt} \rightarrow \text{asgmt}$
- 5) $\text{stmt} \rightarrow \text{if}$
- 6) $\text{stmt} \rightarrow \text{print}$
- 7) $\text{stmt} \rightarrow \text{goto}$
- 8) $\text{stmt} \rightarrow \text{stop}$
- 9) $\text{asgmt} \rightarrow \text{id} = \text{exp}$
- 10) $\text{exp} \rightarrow \text{term exp}^*$
- 11) $\text{exp}^* \rightarrow + \text{term}$
- 12) $\text{exp}^* \rightarrow - \text{term}$
- 13) $\text{exp}^* \rightarrow \text{EMPTY}$
- 14) $\text{term} \rightarrow \text{id}$
- 15) $\text{term} \rightarrow \text{const}$
- 16) $\text{if} \rightarrow \text{IF cond line_num}$
- 17) $\text{cond} \rightarrow \text{term cond}^*$
- 18) $\text{cond}^* \rightarrow < \text{term}$
- 19) $\text{cond}^* \rightarrow = \text{term}$
- 20) $\text{print} \rightarrow \text{PRINT id}$
- 21) $\text{goto} \rightarrow \text{GOTO line_num}$
- 22) $\text{stop} \rightarrow \text{STOP}$

Parsing Table

	line_num	ld	const	IF	PRINT	GOTO	STOP	+	-	<	=	EOF
pgm	1											2
line	3											
stmt		4		5	6	7	8					
asgmt		9										
exp		10	10									
exp*	13							11	12			13
term		14	15									
if				16								
cond		17	17									
cond*										18	19	
print					20							
goto						21						
stop							22					

Part Lexical Scanner

- ใช้ภาษา python
- เปิดไฟล์ input อ่านทีละ line
- Strip หัวท้ายของ line ออก
- แล้ว split ด้วย white space จะทำให้ได้ token แต่ละตัว

Part Parser

- ทำ parsing table และ next set (follow set) ด้วย dict
- ถ้าไม่พบกฎใน parsing table จะมี exception “rule not defined”
- สร้าง set ของ alphabet(A-Z, a-z)
- สร้าง set ของ terminals(+, -, IF, <, =, PRINT, GOTO, STOP, EOF)
- Generate bcode จาก list ของ token
- ถ้า token ไม่ใช่ terminal ของ top ของ stack จะไปทำการหา rule ที่เหมาะสมและ push ลง stack
- ถ้า token match จะทำการ pop ออก