นายนิธิวุฒิ วิไลนุช 5931033021

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

line := line_num stmt

stmt := asgmnt | if | print | goto | stop

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

id is {A..Z}

const is {1..100}

line_num is {1..1000}

นายนิธิวุฒิ วิไลนุช 5931033021

First set	Follow set
First(pgm) = {line_num, EOF}	Follow(pgm) = {EOF}
First(line) = {line_num}	Follow(line) = {EOF, line_num}
First(stmt) = {id, IF, PRINT, GOTO, STOP}	Follow(stmt) = {EOF, line_num}
First(asgmnt) = {id}	Follow(asgmnt) = {EOF, line_num}
First(exp) = {id, const}	Follow(exp) = {EOF, line_num}
First(exp*) = {+, -, EMPTY}	Follow(exp*) = {EOF, line_num}
First(term) = {id, const}	Follow(term) = {+, -, EOF, line_num}
First(if) = {IF}	Follow(if) = {EOF, line_num}
First(cond) = {id, const}	Follow(cond) = {line_num}
First(cond*) = {<, =}	Follow(cond*) = {line_num}
First(print) = {PRINT}	Follow(print) = {EOF, line_num}
First(goto) = {GOTO}	Follow(goto) = {EOF, line_num}
First(stop) = {STOP}	Follow(stop) = {EOF, line_num}

นายนิธิวุฒิ วิไลนุช 5931033021

Rules set

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

นายนิธิวุฒิ วิไลนุช 5931033021

Parsing Table

	line_num	Id	const	IF	PRINT	GOTO	STOP	+	-	<	=	EOF
pgm	1											2
line	3											
stmt		4		5	6	7	8					
asgmnt		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 อ่านที่ละ line
- Strip หัวท้ายของ line ออก
- แล้ว split ด้วย white space จะทำให้ได้ token แต่ละตัว

นายนิธิวุฒิ วิไลนุช 5931033021

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

https://github.com/nithiwut1997/Compiler