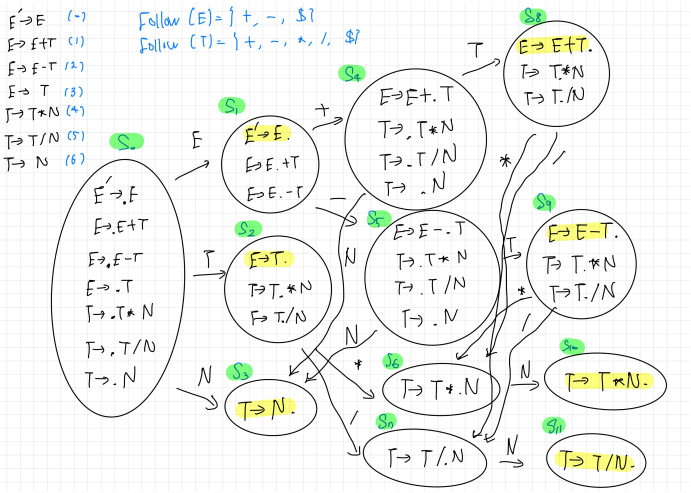
프로그래밍 언어론 HW#1 보고서

이름 : 김주형

학번 : 2021093518

학과 : 컴퓨터소프트웨어학부

0. BNF에 대한 파싱 테이블 구하기



State	Action						Goto	
	N	+	−	*	/	$	E	T
0	S3						1	2
1		S4	S5			Acc		
2		r3	r3	S6	S7	r3		
3		r6	r6	r6	r6	r6		
4	S3							8
5	S3							9
6	S10							
7	S11							
8		r1	r1	S6	S7	r1		
9		r2	r2	S6	S7	r2		
10		r4	r4	r4	r4	r4		
11		r5	r5	r5	r5	r5		

1, 2, 3번 문제에 대한 실행(출력) 결과를 확인하는 방법

1,2,3번 문제를 한 `2021093518.py` 파일의 `SyntaxAnalyzer` 클래스 내에 `method` 로 구현해두었습니다.

한 파일을 통해 여러 expression에 대해 테스트해보고, 각 문제에 대한 결과를 출력하기 위해 **2개의 arugument**를 **command line**으로 전달하여 실행할 수 있도록 하였습니다.

실행 형식은 `python3 2021093518.py [테스트 파일 이름] [문제 번호]` 이고, 예시로 `input1.txt` 파일에 대해 `3` 번 문제의 출력을 확인하기 위해서는 아래와 같이 실행하면 확인할 수 있습니다.

```
python3 2021093518.py input1.txt 3
```

operator의 순서와 개수를 바꿔가며 여러 expression에 대해 올바른 결과가 나오는 것을 test하여 확인하였고, 실행 결과 확인을 위해 과제에 하나의 `input.zip` 으로 압축하여 3개의 테스트 expression 파일을 업로드 하였습니다. (과제 예시에 해당하는 2개의 파일 + 아래의 테스트를 위한 길이가 긴 expression을 가지는 파일)

아래는 **35 + 12 / 15 * 50 - 10 + 10 / 20 - 10** expression이 저장된 `input3.txt` 파일을 1, 2, 3번 문제에 대한 출력 결과입니다.

1번에서 Tokens과 lexems가 올바르게 나눠지는 것을 확인할 수 있고, 2번에서 parsing table을 통해 올바르게 Accept 되는 것을 확인할 수 있으며, 마지막으로 3번에서 올바른 순서대로 parsing이 된 것을 확인할 수 있습니다.

구현에 대한 설명은 `2021093518.py` 파일 내에 각 줄마다 자세하게 주석으로 설명을 덧붙였습니다.

1 번

```
~/ELE4014/HW1 (0.084s)
python3 2021093518.py input3.txt 1

Lexemes : [35, '+', 12, '/', 15, '*', 50, '-', 10, '+', 10, '/', 20, '-', 10, '$']

Tokens : ['N', '+', 'N', '/', 'N', '*', 'N', '-', 'N', '+', 'N', '/', 'N', '-', 'N', '$']
```

2 번

```
Tracing Start!!

+-------+--------------------------+-------------------------------------+--------------------------+
| STATE |          STACK           |                INPUT                |          ACTION          |
+-------+--------------------------+-------------------------------------+--------------------------+
|   0   | 0                        |       N + N / N * N - N + N / N - N $ |                 Shift 3 |
|   1   | 0 N 3                    |         + N / N * N - N + N / N - N $ |    Reduce 6 (Goto[0, T]) |
|   2   | 0 T 2                    |         + N / N * N - N + N / N - N $ |    Reduce 3 (Goto[0, E]) |
|   3   | 0 E 1                    |         + N / N * N - N + N / N - N $ |                 Shift 4 |
|   4   | 0 E 1 + 4                |           N / N * N - N + N / N - N $ |                 Shift 3 |
|   5   | 0 E 1 + 4 N 3            |             / N * N - N + N / N - N $ |    Reduce 6 (Goto[4, T]) |
|   6   | 0 E 1 + 4 T 8            |             / N * N - N + N / N - N $ |                 Shift 7 |
|   7   | 0 E 1 + 4 T 8 / 7        |               N * N - N + N / N - N $ |                Shift 11 |
|   8   | 0 E 1 + 4 T 8 / 7 N 11   |                 * N - N + N / N - N $ |    Reduce 5 (Goto[4, T]) |
|   9   | 0 E 1 + 4 T 8            |                 * N - N + N / N - N $ |                 Shift 6 |
|  10   | 0 E 1 + 4 T 8 * 6        |                   N - N + N / N - N $ |                Shift 10 |
|  11   | 0 E 1 + 4 T 8 * 6 N 10   |                     - N + N / N - N $ |    Reduce 4 (Goto[4, T]) |
|  12   | 0 E 1 + 4 T 8            |                     - N + N / N - N $ |    Reduce 1 (Goto[0, E]) |
|  13   | 0 E 1                    |                     - N + N / N - N $ |                 Shift 5 |
|  14   | 0 E 1 - 5                |                       N + N / N - N $ |                 Shift 3 |
|  15   | 0 E 1 - 5 N 3            |                         + N / N - N $ |    Reduce 6 (Goto[5, T]) |
|  16   | 0 E 1 - 5 T 9            |                         + N / N - N $ |    Reduce 2 (Goto[0, E]) |
|  17   | 0 E 1                    |                         + N / N - N $ |                 Shift 4 |
|  18   | 0 E 1 + 4                |                           N / N - N $ |                 Shift 3 |
|  19   | 0 E 1 + 4 N 3            |                             / N - N $ |    Reduce 6 (Goto[4, T]) |
|  20   | 0 E 1 + 4 T 8            |                             / N - N $ |                 Shift 7 |
|  21   | 0 E 1 + 4 T 8 / 7        |                               N - N $ |                Shift 11 |
|  22   | 0 E 1 + 4 T 8 / 7 N 11   |                                 - N $ |    Reduce 5 (Goto[4, T]) |
|  23   | 0 E 1 + 4 T 8            |                                 - N $ |    Reduce 1 (Goto[0, E]) |
|  24   | 0 E 1                    |                                 - N $ |                 Shift 5 |
|  25   | 0 E 1 - 5                |                                   N $ |                 Shift 3 |
|  26   | 0 E 1 - 5 N 3            |                                     $ |    Reduce 6 (Goto[5, T]) |
|  27   | 0 E 1 - 5 T 9            |                                     $ |    Reduce 2 (Goto[0, E]) |
|  28   | 0 E 1                    |                                     $ |                  Accept |
+-------+--------------------------+-------------------------------------+--------------------------+

Result : 55.5
```

3번

```
================================================
Start !!

enter E
enter T
enter T'
epsilon
exit T'
exit T
enter E'
enter T
enter T'
enter T'
enter T'
epsilon
exit T'
exit T'
exit T'
exit T
enter E'
enter T
enter T'
epsilon
exit T'
exit T
enter E'
enter T
enter T'
enter T'
epsilon
exit T'
exit T'
exit T
enter E'
enter T
enter T'
epsilon
exit T'
exit T
enter E'
epsilon
exit E'
exit E'
exit E'
exit E'
exit E'
exit E
Result : 55.5
```