

Automata #2 Homework

2019-18499 김준혁

실행 결과 (제공 testcase 및 개인 테스트케이스 이용, 터미널에서 바로 입출력)

```
PS C:\Users\r1awn\Desktop\오토마타\hw2> c:: cd 'C:\ProgramData\Local\Programs\Python\Python311\python.exe' '  
_pythonFiles\lib\python\debugpy\adapter/../..\debugpy\lib\python\hw2\hw2_q1.py'  
E:E+T  
E:T  
T:T*F  
T:F  
F:(E)  
F:3  
F:5  
F:7  
  
E:EB  
E:TD  
E:GI  
E:3  
E:5  
E:7  
A:+  
B:AT  
C:*  
D:CF  
G:(  
H:)  
I:EH  
T:TD  
T:GI  
T:3  
T:5  
T:7  
F:GI  
F:3  
F:5  
F:7  
  
PS C:\Users\r1awn\Desktop\오토마타\hw2>
```

Q1의 23hw2.pdf 제공 테스트케이스 결과

```
P5 C:\Users\vr1awn\Desktop\오토마타\hw2> c:: cd "c:\Users\vr1awn\Desktop\오토마타\hw2"; & "c:\Users\vr1awn\AppData\Local\Programs\Python\Python311\python.exe" "c:\Users\vr1awn\.vscode\extensions\ms-python.python-2023.20.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher" "49230" "--" "c:\Users\vr1awn\Desktop\오토마타\hw2\hw2_g1.py"
S:A
A:B
B:C
C:D
D:E
E:F
F:G
G:A
A:1
B:2
C:3
D:4
E:5
F:6
G:7

S:1
S:2
S:3
S:4
S:5
S:6
S:7
A:1
A:2
A:3
A:4
A:5
A:6
A:7
B:2
B:3
B:4
B:5
B:6
B:7
B:1
C:3
C:4
C:5
C:6
C:7
C:1
C:2
D:4
D:5
D:6
D:7
D:1
D:2
D:3
E:5
E:6
E:7
E:1
E:2
E:3
E:4
F:7
F:1
F:2
F:3
F:4
F:5
G:7
G:1
G:2
G:3
G:4
G:5
G:6
```

Q1의 개인 테스트케이스 결과

```
PS C:\Users\r\lawn\Desktop\오토마타\hw2> c:: cd 'c:\Users\r\lawn\
ata\Local\Programs\Python\Python311\python.exe' 'c:\Users\r\lawn\
\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher'
\hw2\hw2_q2.py'
S:AB
S:BC
A:BA
A:3
B:CC
B:5
C:AB
C:3
53353
yes
```

Q2의 lec.pdf 제공 테스트케이스 결과

```
C:\Users\r1am\Desktop>오르마티\hw2 c:: cd 'c:\Users\r1am\Desktop\오르마티\hw2' ; & 'C:\Users\r1am\Desktop\오르마티\hw2\hw2.py'
C:\Users\r1am\Desktop\오르마티\hw2>cd ..
C:\Users\r1am\Desktop\오르마티>cd ..
C:\Users\r1am\Desktop>pythonFiles\11b\python\debuggy\adapter\...\debuggy\launcher "49340" "-" "C:\Users\r1am\Desktop\오르마티\hw2\hw2_g2.py"
S:CB
B:aC
C:CB
C:CC
a:+
a:-
a:*
a:/
C:1
C:2
C:3
C:4
C:5
C:6
C:7
C:8
C:9
C:0
1423+154-6543*-2354+2341
00.
```

Q2의 개인 테스트케이스(다른 문장) 결과

프로그램 알고리즘 설명

hw2_q1.py :

기본 문맥무관언어 → Chomsky Normal Form 변환 알고리즘은 lecture에 제시된 것을 따라, RemoveUnitProduction(), convertChomskyNormForm()으로 단계를 나누어 작성했다. 단위 생성규칙을 제거하는 과정에서, 단위 생성규칙만으로 그래프를 구성, 이를 이용해 $A^* \Rightarrow B$ 관계를 구하였다. Chomsky Normal Form으로 변환하는 단계에서 변수를 최대한 적게 사용하기 위해, 새로운 변수를 도입해 규칙을 만드는 상황에서 이미 그러한 규칙이 만들어졌다면, 이를 다시 재사용하는 식으로 코드를 구성했다.

이러한 과정이 있다보니, slide의 알고리즘 그대로 구현하면, 52개의 변수를 넘어가게 될 입력에도 변수 개수가 넘어가지 않는 경우가 존재한다. 테스트 과정에서도 사용되어야 할 변수의 개수가 적은 것을 확인할 수 있다.

hw2_q2.py :

lec.pdf에 제공된 알고리즘을 따랐으며, CYK 알고리즘의 table을 구성하면서 변수들을 중복으로 저장하지 않도록 set을 사용하여, 각 element가 변수의 set인 2차원 배열이 되도록 하였다. 이후 알고리즘은 lecture의 알고리즘과 거의 동일하며, 최종적으로 table[1][n]의 element(set)에 "S"가 포함되었는지를 확인하여 yes/no를 판별하였다.

테스트에 lecture에 제시된 예제를 조건에 맞게 약간 변화시켰을 때에 같은 결과 및 table값을 가지는 것을 확인했으며, Q2에 대해 제출한 개인 테스트케이스는 "yes"가 나오는 문장을 넣었으며, 보고서에는 "no"가 나오는 문장을 넣은 결과를 첨부하였다.

Environment & How to run

```
root@4a2effdaf185:/home/hw1# python3 --version
Python 3.10.7
root@4a2effdaf185:/home/hw1# cat ./2019-18499/Automata_HW1_2019-18499_compile.sh
root@4a2effdaf185:/home/hw1# cat ./2019-18499/Automata_HW1_2019-18499_Q1.sh
python3 ./hw1_q1.pyroot@4a2effdaf185:/home/hw1# cat ./2019-18499/Automata_HW1_2019-18499_Q2.sh
python3 ./hw1_q2.pyroot@4a2effdaf185:/home/hw1#
```

제공된 docker container 환경에서도 올바른 결과가 나오는 것을 확인하였으며, 보고서에 첨부된 파일은 입출력을 한 번에 터미널에서 볼 수 있도록 개인 환경에서 실행했으며, python 3.11.4를 이용한 vscode 내에서 실행했다.