Mock Exam

Compilers, 2022, Term 2

- **0 Score** for the corresponding assignment **for both the copy and source**.

1.

Consider the context-free grammar:

```
S -> S S + | S S * | a
```

and the string aa + a*.

1. Give a leftmost derivation for the string.
2. Give a rightmost derivation for the string.
3. Give a parse tree for the string.

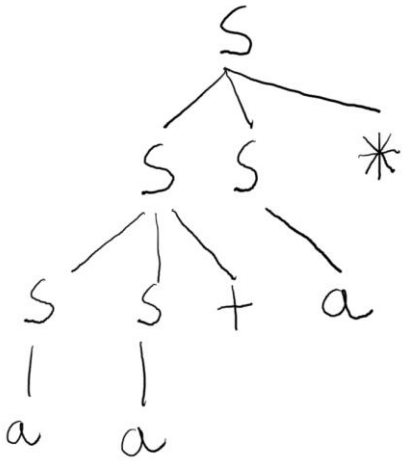
1-1. Give a leftmost derivation for the string.

S => SS* => SS + S* => aS + S* => aa + S* => aa + a*

1-2. Give a rightmost derivation for the string.

S => SS* => Sa* => SS + a* => Sa + a* => aa + a*

1-3. Give a parse tree for the string.



2. Repeat 1 above for each of the following grammars and strings

1. S -> 0 S 1 | 0 1 with string 00011l.
2. S -> + S S | * S S | a with string + * aaa.

(In 1, with string 000111)

2-1. S -> 0S1|01 with string 00011|.

Leftmost: S => 0S1 => 00S11 => 000111

Rightmost: S => 0S1 => 00S11 => 000111

Parse tree:

2-2. S->+SS|*SS|a with string +*aaa.

Leftmost: S => +SS => +*SSS => +*aSS => +*aaS => +*aaa

Rightmost: S => +SS => +Sa => +*SSa => +*Saa => +*aaa

Parse tree:

3. Answer 1, 2, 3 & 5.

Design grammars for the following languages:

1. The set of all strings of 0s and 1s such that every 0 is immediately followed by at least one 1.
2. ! The set of all strings of 0s and 1s that are palindromes; that is, the string reads the same backward as forward.
3. ! The set of all strings of 0s and 1s with an equal number of 0s and 1s.
4. !! The set of all strings of 0s and 1s with an unequal number of 0s and 1s.
5. ! The set of all strings of 0s and as in which 011 does not appear as a substring.
6. !! The set of all strings of 0s and 1s of the form xy, where x<>y and x and y are of the same length.

3-1. The set of all strings of 0s and 1s such that every 0 is immediately followed by at least one 1.

S -> (0?1)*

*. 정규 표현식으로 표현됨

모든 0 바로 뒤에 적어도 하나의 1이 오도록 하기 위해서 ? (앞에 한문자가 있거나 없거나) 표현을 사용해 0~1 or 1을 표현하고 이를 *를 이용하여 반복하여 표현

3-2. ! The set of all strings of 0s and 1s that are palindromes; that is, the string reads the same backward as forward.

S -> 1S1 | 0S0 | 1 | 0 | EPSILON

회문 구성을 위해서 1S1, 0S0으로 두가지 논터미널를 잡은 뒤, 내부에서 0, 1또는 3글자 이상으로 구성된 S 확장이 매치되도록 문법을 구성

3-3. ! The set of all strings of 0s and 1s with an equal number of 0s and 1s.

S -> 0S1S | 1S0S | EPSILON

0과 1이 동일한 두가지 논터미널 0S1S, 1S0S을 설정하여, 추후 확장간 동계속 0, 1이 동시에 추가되도록 문법을 구성

3-5. ! The set of all strings of 0s and as in which 011 does not appear as a substring.

S -> 1*(0+1?)*

*. 정규표현식으로 표현됨

011의 부분 문자열을 가지지 않기 위해서, 0+ 확장 뒤에는 최대 1개의 1이 있게 하고 이 형태를 *를 이용하여 확장하고 (“(0+1?)*”) 무조건 1로 시작하는 조건을 설정하여 011 형태의 부분 문자열이 들어가지 않도록 표현

4.

There is an extended grammar notation in common use. In this notation, square and curly braces in production bodies are metasymbols (like -> or |) with the following meanings:

1. Square braces around a grammar symbol or symbols denotes that these constructs are optional. Thus, production A -> X[Y]Z has the same effect as the two productions A -> XYZ and A -> XZ.
2. Curly braces around a grammar symbol or symbols says that these symbols may be repeated any number of times, including zero times. Thus, A -> X{YZ} has the same effect as the infinite sequence of productions A -> X, A -> XYZ, A -> XYZYZ, and so on.

Show that these two extensions do not add power to grammars; that is, any language that can be generated by a grammar with these extensions can be generated by a grammar without the extensions.

4-1.

A -> X[Y]Z는 비확장 문법 A -> XZ | XYZ로 표현 가능하다.

4-2.

A -> X{YZ}는 비확장 문법 A -> XB, B -> YZB | EPSILON으로 표현 가능하다.

4-2, 4-2의 예시를 확인하면 [] , {}등의 확장 문법은 모두 확장 문법으로 풀어 쓸 수 있으므로, 확장 문법과 비확장 문법 간에 편의성을 제외하면 문법의 표현력 차이는 없다.

5.

Use the braces described in Exercise 4.2.4 to simplify the following grammar for statement blocks and conditional statements:

```
stmt -> if expr then stmt else stmt
      | if stmt them stmt
      | begin stmtList end
stmtList -> stmt; stmtList | stmt
```

4.2.4. (본 과제 4번)에서 사용한 확장 문법을 이용하여 위의 문법을 단순화하면 아래와 같다.

stmt -> if expr then stmt [else stmt] | begin stmtList end
stmtList -> stmt [; stmtList]

<The End of the Mock Test>