PUSAN NATIONAL UNIVERSITY

REPORT

프로그래밍언어론 과제3

과 목 명 :	프로그래밍언어론
지도교수 :	
학 과 :	전기컴퓨터공학부 정보컴퓨터공학전공
학 번 :	
이 름 :	장 수현
제 출 일 :	2020년 4월 23일

부산대학교

1. Show that the language generated by the following grammar is a regular language:

$$S \rightarrow aSa \mid a$$

sol)

위의 정규 문법으로부터 생성되는 언어는 다음과 같다.

$L(G) = \{ a^{2n-1} \mid n \geq 1 \}$

예) a, aaa, aaaaa, ...

그리고 위의 정규 문법을 인식하는 유한 오토마타를 만들 수 있다.

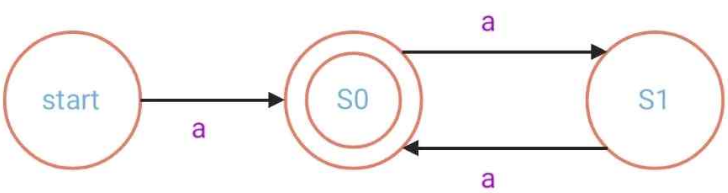

그림 1. 정규 문법 S → aSa | a로부터 생성된 유한 오토마타.

이처럼 오토마타를 만들 수 있게 되면, 그에 대응하는 정규 표현도 만들 수 있다. 따라서 주어진 정규 문법으로부터 생성되는 언어는, 정규 표현식을 이용하여 표현할 수 있는 형식 언어라고 정의되는 정규 언어라고 볼 수 있다.

2. Given the context-free grammar

$$S ::= 0S0 \mid 1S1 \mid 0 \mid 1$$

give the derivation tree for 0110110.

sol)

주어진 문장 0110110을 만들기 위한 parse tree는 다음과 같다.

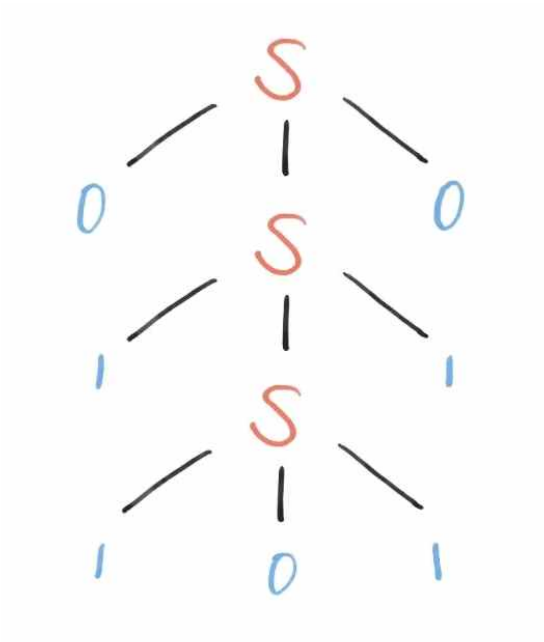

그림 2. 0110110의 parse tree.

3. We know that strings of the form anbn require a context-free grammar for their generation. Consider the following regular grammar:

S ::= aS | bS | a | b

A claim is made that it can generate anbn. For example, a3b3 is generated by the following derivation:

S => aS => aaS => aaaS => aaabS => aaabbbS => aaabbb

Explain this apparent contradiction in using Type3 grammar to generate a Type2 language.

sol)

2타입 Context Free Grammars로부터 anbn라는 언어가 생성되기 위해서는 a 혹은 b의 개수를 count 할 수 있도록 stack을 가지고 있어야 한다. 그래서 2타입 오토마타인 PDA(Pushdown Automata)는 stack을 가지고 있어서 push, pop 혹은 ignore 중 액션을 취할 수 있기 때문에 anbn과 같은 언어의 생성이 가능하다. 그러나, 3타입 오토마타인 FA(Finite Automata)의 경우, stack이 없기 때문에 a와 b의 카운트가 불가능하여 anbn 형태의 언어를 생성할 수 없다. 대신, FA는 Regular Grammar를 인식하여 언어를 생성하기에 적합한 오토마타이다.

4. Prove the following program for integer division:

```
{x≥0 ∧ y>0}
q := 0;
r := x;
while y ≤ r do
        begin
                r := r - y;
                q := q + 1;
        end
{y>r ∧ x=r+y*q}
```

sol)

Invariant for the while loop: $r \geq d$

Now for the while loop to run, the condition $r \geq d$ must be true before every iteration. So, $r \geq d$ is a loop invariant for the while loop. That makes the conditions $r > d$, $r < d$ and $r \leq d$ not loop invariants for the while loop. Condition $r = a$^$q = 0$ is also not true since r is a for the first loop and a is not necessarily 0, so it is true for every iteration.

$r \geq 0$ loop invariant: $a=dq+r$

For the first iteration, r is a and q is 0. So, $dq+r = d*0+a = a$ is true. Let's assume it is true for the nth iteration. Since q is increased by 1 each time, it is n-1 at the start of the iteration. So, $a=d(n-1)+r$ is true. Now q becomes n for the next iteration and r becomes r-d. So, for the n+1th iteration, $dq+r = d(n)+r-d = d(n-1)+r$. From the previous iteration, we know $d(n-1)+r$ is a, so it holds true. It holds true for iterations n and n+1, so it is true for every iteration and is an loop invariant.