**Automata Theory Homework3**

컴퓨터공학부

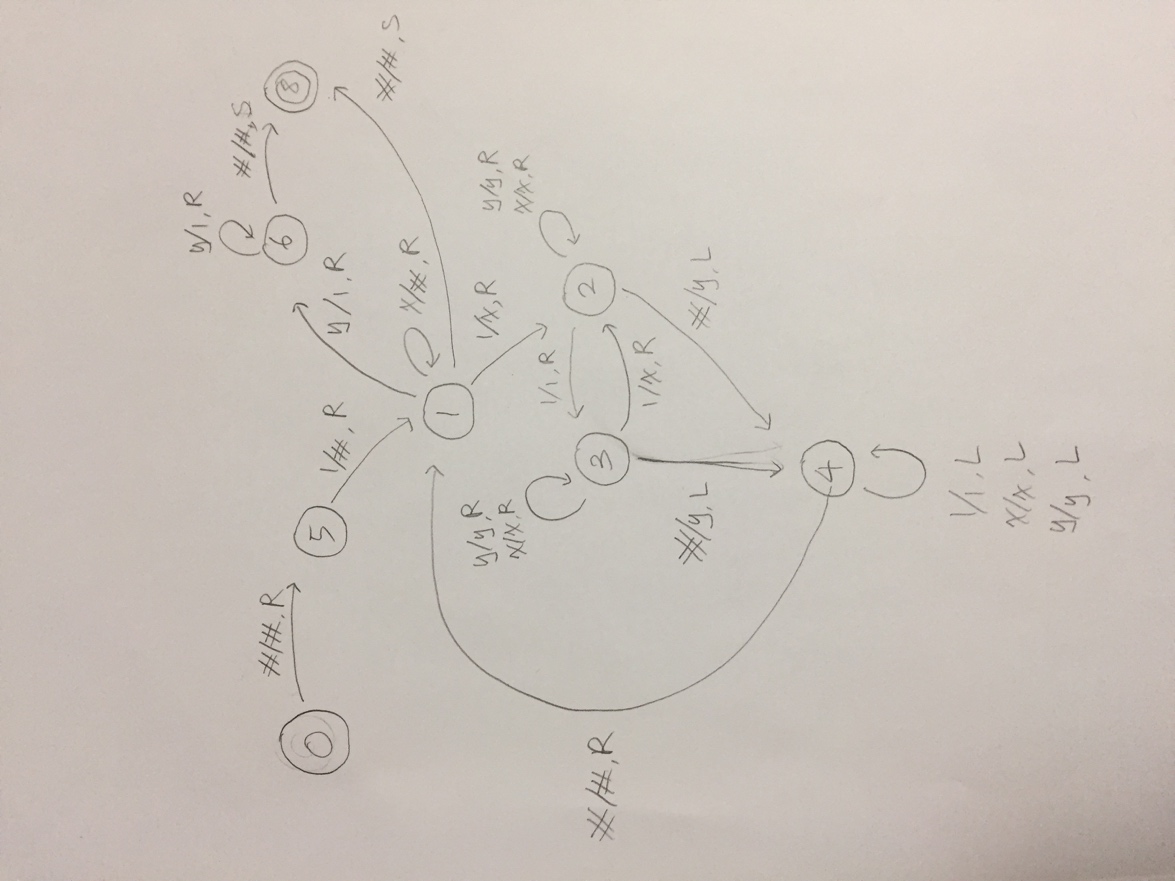
2015-14521

김정래

**2. Turing Machine L that computes log(N)의 상한**

**- input : N in unary notation (N>=1)**

**- output : log(N)의 상한 in unary notation**



위 그림에서 죽는 노드(7) 로 가는 edge는 생략했다.

pf) 우리가 알고 있는 숫자는 십진수이다. 이들을 이진수로 바꾸었을 때 나누기 2나 곱하기 2에 해당하는 연산은 shift로 해결할 수 있다. 오른쪽으로 shift 할 경우 2로 나눈 몫이고 왼쪽으로 shift 할 경우 2를 곱한 값이다.

log(n)의 하한 + 1은 이진수로 표현했을 때 자릿수이다. 귀납법으로 쉽게 증명이 가능하다. 따라서 log(n)의 하한 + 1을 구하는 방법은 숫자가 0이 될때까지 2로 나눈 횟수를 출력하면 된다. 그림에서 아래쪽에 해당하는 TM이 나누기 2를 하는 과정이고 오른쪽 위는 남은 숫자가 더 이상 없을 경우 카운트를 출력하는 과정이다. 문제에서 요구하는 것은 log(N)의 상한이므로 log(N-1)의 하한 + 1 = log(N)의 상한이라는 등식을 이용해서 상한을 구할 것이다. 상세한 과정은 뒤에서 설명할 것이다.

다음은 log(N)의 상한을 계산하는 TM의 전이함수이다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | 7 | 1 |  |  |  |
|  | 0 | 1 | x | y | # |
| 0 | (7,0,S) | (7,1,S) | (7,x,S) | (7,y,S) | (5,#,R) |
| 1 | (7,0,S) | (2,x,R) | (1,#,R) | (6,1,R) | (8,#,S) |
| 2 | (7,0,S) | (3,1,R) | (2,x,R) | (2,y,R) | (4,y,L) |
| 3 | (7,0,S) | (2,x,R) | (3,x,R) | (3,y,R) | (4,y,L) |
| 4 | (7,0,S) | (4,1,L) | (4,x,L) | (4,y,L) | (1,#,R) |
| 5 | (7,0,S) | (1,#,R) | (7,x,S) | (7,y,S) | (7,#,S) |
| 6 | (7,0,S) | (7,1,S) | (7,x,S) | (6,1,R) | (8,#,S) |

ReadMe.txt

입력이 unary로 들어온다고 하나 혹시 몰라 0이라는 숫자가 나올시 바로 dead state(N state)로 가게 했다. 또한 과정 내내 0을 쓰거나 만들지 않는다.

과정은 다음과 같다.

1 은 남아있는 숫자. x는 무의미한 숫자. y는 총 스텝의 수. (카운트)

1. 1을 하나 뺀다. N-1로 만든다.

2. 시작을 할때 앞에 있는 x를 1이 나올때까지 계속 지운다. 그러다가 1 이 나올 경우 3번을, y가 나올 경우 5번을 실행한다.

3. 끝에 도달할때까지1을 두개씩 묶어 첫 1은 x로 바꾸고 두 번째 1은 그대로 놔둔다. 1의 개수를 반으로 줄이는 과정이다.

4. 다시 처음으로 돌아오고 2을 진행한다.

5. y가 나왔다는 것은 남아있는 숫자가 없다는 뜻이므로 카운트를 그대로 출력하면 된다. 따라서 남아있는 y를 모두 1로 바꾸고 출력하면 된다.

따라서 모든 과정을 진행한 후에 남아있는 값은 log(n)의 상한이다.

추가적으로 7번은 입력이 올바르지 않을 경우 들어가는 dead state이고 8번은 입력이 올바를때 들어가는 dead state 이다.