

I. 구현한 Universal Turing machine이 어떻게 동작하는가?

1.state는 3가지 Hash Map을 가지고 있다.

2.첫 번째 map은 head가 가리키는 symbol에 따른 전이 state를,

두 번째 map은 head가 가리키는 symbol이 바뀌어질 next symbol,

그리고 세 번째 map은 head가 가리키는 symbol에 따라 head가 움직일 방향을 저장한다.

3.tape는 1개이며 Array List로 구현되어 처음에 #과 주어진 입력을 저장한다.

tape head는 처음에 leftmost #를 가리키고 있다.

4.tape head가 가리키는 위치의 symbol을 읽는다.

4-1.현재 state가 저장한 hash map으로부터 next symbol을 읽고 현재 head가 가리키는 위치에 symbol을 대치(replace)한다.

4-2.현재 state가 저장한 hash map으로부터 head direction을 읽어와 tape head의 위치를 갱신(update)한다.

4-3.현재 state가 저장한 hash map으로부터 next state를 읽어와 현재 state를 갱신(update)한다.

5.만약 갱신된 state가 halting state가 아니라면 4를 반복하고, halting state라면 정지한다.

또한 No halting state라면 No를, Yes halting state라면 Yes를 출력하며 tape contents를 출력한다.

II. Ceiling($\log_2 n$)을 계산하는 Turing machine L이 어떻게 동작하는가

1. tape에 주어진 입력 string을 저장한다.

2. 입력 string의 마지막에 0을 저장한다. 0의 개수는 $\log_2 n = k$ 라고 했을 때, 2^k 를 나타낸다.

(즉, $2^k = 1$ 부터 n 과 비교한다.)

3. leftmost 1과 leftmost 0을 번갈아 가면서 각각 x, y 로 바꾼다.

4.만약 1은 남아있는데 0이 남아있지 않은 경우, $n > 2^k$ 라는 것이므로 k 를 1 증가시켜야 한다. k 는 symbol A를 이용한 단일진법으로서 표시된다.

4-1. 3에서 바꾸었던 x, y 를 1과 0으로 되돌린다.

4-2. 0의 개수만큼 0을 복제해야 한다. 이 때 오른쪽의 A개수는 그대로 여야 한다.

left most 0을 x , leftmost A를 y , rightmost #을 A로 바꾼다.

4-3. 더 이상 0이 없다면 x, y를 모두 0으로 되돌린다.

4-4. tape head를 leftmost #으로 이동시킨 후 과정 3으로 되돌아간다.

5. 만약 0은 남아있는데 1이 남아있지 않은 경우, $n \leq 2^k$ 라는 것이므로 symbol A만 놔두고 tape contents를 지운다. 그 후 A를 1로 바꾸어 종료한다.

아래는 위의 turing machine을 state diagram으로 나타낸 것이다.

