

(정의) 논리식(Boolean Expression)

$$e \rightarrow e \vee e \mid e \wedge e \mid \neg e \mid (e) \mid v \mid T \mid F$$

논리식은 $\{T, F\}$ 를 값으로 하고 변수와 식을 만드는 연산자 \vee, \wedge, \neg 를 허용하는 식이다.

n 개의 변수로 이루어진 논리식을 $B(v_1, v_2, \dots, v_n)$ 이라고 하자. n 개의 변수 v_1, v_2, \dots, v_n 에 $\{T, F\}$ 값을 배정하는 것을 assignment(2^n 개)라고 부르고, 이 중 논리식을 $B(v_1, v_2, \dots, v_n)$ 의 값을 T 로 하는 assignment를 truth assignment라고 부른다.

(정의) n 변수 논리식 $B(v_1, v_2, \dots, v_n)$ 의 truth assignment가 존재하면 논리식 $B(v_1, v_2, \dots, v_n)$ 은 만족한다(satisfiable)고 말한다.

(정의) 임의의 Bool식 $B(v_1, v_2, \dots, v_n)$ 이 satisfiable한가 안한가의 문제를 satisfiability(SAT) 문제라고 부른다.

(정리) SAT는 최초의 NP-complete 문제이다.(Cook's Theorem)

(1) $SAT \in \mathbf{NP}$.

(2) $\forall P \in \mathbf{NP}, P' \leq_p SAT$.

(사실) $SAT \leq_p P$ 이면 P 도 NP-complete이다.

(증명) (1) $SAT \in \mathbf{NP}$.

n 변수 이므로 2^n 개의 서로 다른 assignment가 있다. 하나의 assignment가 Bool식 $B(v_1, v_2, \dots, v_n)$ 을 T 로 하는가 F 로 하는가를 확인하는데는 polynomial 시간이면 충분하다.

Polynomial 알고리즘을 2^n 개를 nondeterministic하게 동시에 돌리는 것이 SAT를 위한 알고리즘이다.

$\therefore SAT \in \mathbf{NP}$ 이다.

(2) $\forall P \in \mathbf{NP}, P \leq_p SAT$.

임의의 $P \in \mathbf{NP}$ 이므로 Nondeterministic TM(NTM)에서 $p(n)$ 번에 움직임이 있고 하자.

(a) $\alpha_1 \Rightarrow \alpha_2 \Rightarrow^* \dots \Rightarrow \alpha_{p(n)}$. $1 \leq \forall i \leq p(n): \alpha_i \in \Gamma^* \times Q \times \Gamma^* \wedge |\alpha_i| \leq p(n)$.

$1 \leq \forall i, \forall j \leq p(n): \alpha_i = X_{i,1}X_{i,2} \dots X_{i,p(n)}$. $X_{ij} \in Q \cup \Gamma$.

$(p(n))^2$ cells

$O(p(n))$ cells

(b) $y_{ijA} \stackrel{\text{def}}{=} X_{ij} = A$.

$(p(n))^2 \cdot (|Q| + |\Gamma|)$ 변수

$O(p(n))$ 변수

(c) 임의의 P 를 위한 다항식 시간 TM M 은 입력 문자열 $w \in \Sigma^*$ 에 관하여

$E_{M,w}$ = 규칙을 지키며 \wedge 잘 시작해서 \wedge 잘 한 후에 \wedge 올바르게 끝낸다.

4개의 논리식의 \wedge 로 표시된다.

(1) 규칙을 지키며(U : Unique): 한 셀에는 하나의 심벌만 있어야한다.

$$U = \bigwedge_{1 \leq \forall i \leq p(n)} \bigwedge_{1 \leq \forall j \leq p(n)} \bigwedge_{\forall A \neq B \in Q \cup \Gamma} \neg(y_{ijA} \wedge y_{ijB})$$

$$(p(n))^2 \cdot (|Q| + |\Gamma|)^2 \text{개의 } \wedge \text{ 식} \qquad O(p(n)) \text{ 식}$$

(2) 잘 시작해서 (\mathcal{S} : Start)

$$w = a_1 \cdots a_n \text{ 이라면}$$

$$\mathcal{S} = y_{1,1,q_0} \wedge y_{1,2,a_1} \wedge y_{1,3,a_2} \wedge \cdots \wedge y_{1,n+1,a_n} \wedge y_{1,n+2,B} \wedge \cdots \wedge y_{1,p(n),B} \cdot$$

$$p(n) \text{개 변수의 } \wedge. \qquad O(p(n)) \text{ 식}$$

(3) 잘 한 후에 (\mathcal{N} : Next)

$$\mathcal{N} = \bigwedge_{1 \leq i < p(n)} \bigwedge_{1 \leq j < p(n)} (A_{i,j} \vee B_{i,j}).$$

$$(p(n) - 1)^2 \text{개의 } \wedge \text{ 식} \qquad O(p(n)) \text{ 식}$$

(A) $X_{i,j}$ 가 상태이면, 세 개의 cell이 창(window)이다.

$$X_{i,j-1} X_{i,j} X_{i,j+1} = DqA \text{ 라고 하자, 단 } q \in Q, D, A \in Q.$$

$$(1) \delta(q, A) = (p, C, R) \text{ 이면} \quad \cdots DqA \cdots \Rightarrow \cdots D C p \cdots$$

$$A_{i,j} = y_{i,j-1,D} \wedge y_{i,j,q} \wedge y_{i,j+1,A} \wedge y_{i+1,j-1,D} \wedge y_{i-1,j,C} \wedge y_{i-1,j+1,p} \cdot$$

$$(2) \delta(q, A) = (p, C, L) \text{ 이면} \quad \cdots DqA \cdots \Rightarrow \cdots p D C \cdots$$

$$A_{i,j} = y_{i,j-1,D} \wedge y_{i,j,q} \wedge y_{i,j+1,A} \wedge y_{i+1,j-1,p} \wedge y_{i-1,j,D} \wedge y_{i-1,j+1,C} \cdot$$

(B) $X_{i,j-1}$ 이나 $X_{i,j+1}$ 이 상태이거나, $X_{i,j}$ 가 tape 심벌이다.

$$Q = \{q_1, q_2, \dots, q_m\} \text{ 이고 } \Gamma = \{Z_1, Z_2, \dots, Z_r\} \text{ 라고 하자.}$$

$$B_{i,j} = (\bigvee_{1 \leq k \leq m} y_{i,j-1,q_k}) \qquad X_{i,j-1} \in Q \vee$$

$$\vee (\bigvee_{1 \leq k \leq m} y_{i,j+1,q_k}) \qquad X_{i,j+1} \in Q \vee$$

$$\vee ((\bigvee_{1 \leq k \leq r} y_{i,j,Z_k}) \wedge (X_{i,j} \in \Gamma \wedge$$

$$(\bigvee_{1 \leq k \leq r} (y_{i,j,Z_k} \wedge y_{i+1,j,Z_k}))) \qquad X_{i,j} = X_{i+1,j})$$

(4) 올바르게 끝낸다. (\mathcal{F} : Finish)

$$\alpha_{p(n)} \text{ 에서 끝나고, 최종상태가 } \{f_1, f_2, \dots, f_q\} \text{ 라면,}$$

$$\mathcal{F} = \bigvee_{1 \leq j \leq p(n)} \bigvee_{1 \leq k \leq q} y_{p(n),j,f_k} \cdot$$

$$|F| \cdot p(n) \text{ 개 변수의 } \vee. \qquad O(p(n)) \text{ 식}$$

(1) 규칙을 지키며 \wedge (2) 잘 시작해서 \wedge (3) 잘 한 후에 \wedge (4) 올바르게 끝내는 위의 증명 과정이 **모든 NP** 문제를 $O(p(n))$ 의 크기를 가지는 SAT로 polynomial reduction 하였으므로 SAT는 정의대로 NP-completeness이다.