Cook定理的证明

NP问题的定义

A problem P $\in$ class NP:
p is a decision problem,$\exists$ a polynomial p and a polynomial-time algorithm A
such that, $\forall$ instance x of P:
x is a YES-instance if and only if
there exists a string y with $|y| \leq p(|x|)$ such that A(x,y) returns YES.

一些例子:

3-SATISFIABILITY(3-SAT):

INSTANCE: A set U of varibales and collection C=$\{c_1, c_2, ...c_m\}$ of clauses,such that $|c_i|$=3 for $1 \leq i \leq m$

QUESTION: Is there a truth assignment for U that satisfies all the clauses in C?

VERTEX COVER(VC):

INSTANCE: A graph G=(V,E),and a posttive integer K$\leq$|V|

QUESTION: Is there a vertex cover of size K or less for G?

SAT问题

SATISFIABILITY:
INSTANCE:A set U of variables and collection C of clauses over U.
QUSETION: Is there a satisfying truth assignment for C?

size:|I|=|C||U|,I is an Instance of **SAT** e.g.

Instance:U=$\{A、B、C、D、E\}$,C=$\{A \vee B, \overline{A} \vee C \vee D, \overline{A} \vee E, \overline{B} \vee \overline{C} \vee \overline{E}, C \vee E\}$
Question: Is there a satisfying truth assignment for C?

(U,C)是一个Yes-Instance
certificate y=$\{A = true, B = false, C = true, D = false, E = true\}$

|I|=5 × 5=25

1.SAT is NP

$\forall$I$\in${Instance of **SAT**},若I$\in${Yes-Instance of **SAT**},则I的truth assignment y即为I的certificate.

若I=(U,C),那么|y|=|U|$\leq$|C||U|=|I|,并且存在一个运算时间为O(|I|)的算法verify I.

$\therefore$**SAT** is NP.

2.∀D is NP,D ∝SAT

2.1 NP问题的图灵机表示

设**D** is NP,x为**D**的一个Instance,|x|=n,若x是**D**的一个Yes-Instance,那么∃c为x的certificate,B=B(x,c)为certifier,且|c|和B的运算时间均为n的polynomial size.

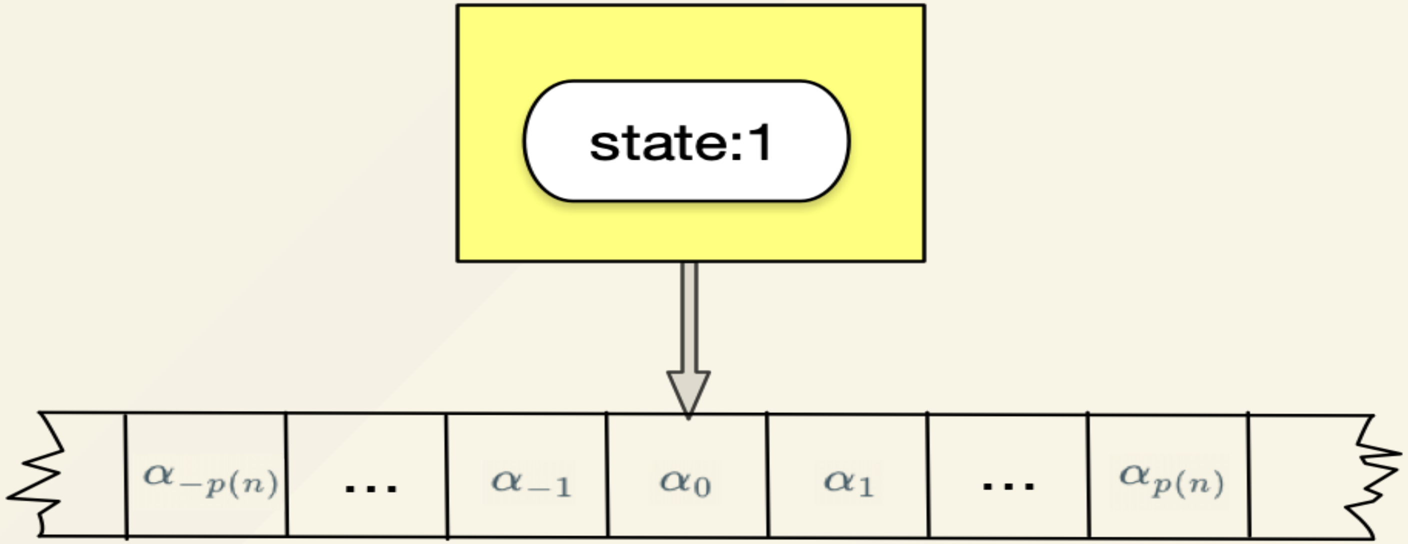
假设已经给定一组(x,c),算法B可以在一个确定性图灵机 TM上求解出来.B的运算时间为$p(n)$ ($p(n)$为n的一个special polynomial),并且有:

1)状态集$State = \{0,1,2,...q-1\}$,假定开始状态为1,停机时状态为0(Yes-output),2(No-output)

2)符号集$Symbol = \{0,1\}$(对于不同的问题可能有不同的符号集,但是两个已足够)

3)动作集$Action = \{-1,0,+1\}$分别表示tape向左移一格,不动,向右移一格

在纸带上取一个格子表示0号格子,左边的格子分别为-1,-2,-3,...,右边的格子分别为+1,+2,+3,....将0号格子作为分隔,在0号格子的左边放置c的encoding,右边放置x的encoding,并且tape初始时位于1号格子,停机时位于0号格子.由于TM的运算时间为$p(n)$步,所以tape的移动不会超出$-p(n) \sim p(n)$之外,记初始时刻纸带从$-p(n) \sim p(n)$的符号为$\alpha_{-p(n)}, \ldots \alpha_{p(n)}$



图灵机TM的初始状态

算法B被表示为图灵机TM上的transition function f:

$$f : State \times Symbol \mapsto State \times Symbol \times Action$$

方便起见,将f拆成三个函数:

$$T : State \times Symbol \mapsto State$$
$$U : State \times Symbol \mapsto Symbol$$
$$D : State \times Symbol \mapsto Action$$

分别改变TM的当前状态,当前格子的符号和tape的移动方向.

2.2 图灵机运行过程的表示

for $i = 0, 1, 2 ... p(n), j = -p(n) ... p(n), k = 0, 1$

Q_{ij}表示在第i步时,TM的状态是否为j;

T_{ij}表示在第i步时,tape是否位于格子j;

S_{ijk}表示在第i步时,格子j的符号是否为k

我们希望用这些变量描述图灵机的运行过程.

2.2.1 系统初始时的状态固定

1.图灵机初始时处于状态1: $Q_{01} = 1$

2.tape初始时处于1号格子: $T_{01} = 1$

3.纸带上初始时有一组符号分配(0号格子左边为c的encoding,右边为x的encoding):

$$\begin{cases} S_{01\alpha_{(-p(n))}} = 1 \\ S_{01\alpha_{(-p(n)+1)}} = 1 \\ \dots \\ S_{01\alpha_{p(n)}} = 1 \end{cases}$$

2.2.1中的子句数量为$O(p(n))$

2.2.2 系统停机时的状态固定

1.图灵机停机时处于状态0: $Q_{p(n)0} = 1$

2.tape停机时处于0号格子: $T_{p(n)0} = 1$

3.纸带在停机时0号格子上的符号为1: $S_{p(n)01} = 1$

2.2.2中子句数量为3

2.2.3 每步中,TM的状态有且仅有一个

$$\begin{cases} Q_{i0} \vee Q_{i1} \vee \ldots \vee Q_{i(q-1)} = 1 \\ \neg(Q_{ij} \wedge Q_{ik}) = \overline{Q_{ij}} \vee \overline{Q_{ik}} = 1 \quad (j \neq k) \end{cases}$$

其中,$i \in \{1, 2, \ldots p(n) - 1\}, j, k \in State$

2.2.3中子句数量为$O(p(n))$

2.2.4 每步中,tape只在纸带上的其中某一个格子出现

$$\begin{cases} T_{i(-p(n))} \vee \ldots \vee T_{ip(n)} = 1 \\ \neg(T_{ij} \wedge T_{ik}) = \overline{T_{ij}} \vee \overline{T_{ik}} = 1 \quad (j \neq k) \end{cases}$$

其中,$i \in \{1, 2, \ldots p(n) - 1\}, j, k \in \{-p(n), \ldots p(n)\}$

2.2.4中子句数量为$O(p^3(n))$

2.2.5 每步中,纸带上的每个格子有且仅有一个符号

$$\begin{cases} S_{ij0} \vee S_{ij1} = 1 \\ \neg(S_{ij0} \wedge S_{ij1}) = \overline{S_{ij0}} \vee \overline{S_{ij1}} = 1 \end{cases}$$

其中,$i \in \{1, 2, ...p(n)\}, j \in \{-p(n), ...p(n)\}$

2.2.5中子句数量为$O(p^2(n))$

2.2.6 从一步到下一步的transition

1.图灵机TM从第i到第i+1步的变化(改变状态)满足transition function T:

$$T_{ij} \wedge Q_{ik} \wedge S_{ijl} \rightarrow Q_{(i+1)T(k,l)}$$

即:

$$(T_{ij} \wedge Q_{ik} \wedge S_{ijl}) \vee \overline{Q_{(i+1)T(k,l)}} = 1$$

其中,$i \in \{0, 1, 2, ...p(n) - 1\}, j \in \{-p(n), ...p(n)\}, k \in State, l \in Symbol$

1中子句数量为$O(p^2(n))$

2.tape从第i到第i+1步的变化(移动)满足transition function D:

$$T_{ij} \wedge Q_{ik} \wedge S_{ijl} \rightarrow S_{(i+1)jU(k,l)}$$

即:

$$(T_{ij} \wedge Q_{ik} \wedge S_{ijl}) \vee \overline{S_{(i+1)jU(k,l)}} = 1$$

其中,$i \in \{0, 1, 2, ...p(n) - 1\}, j \in \{-p(n), ...p(n)\}, k \in State, l \in Symbol$

2中子句数量为$O(p^2(n))$

3.纸带从第i到第i+1步的变化(改变格子上的符号)满足transition function U:

$$T_{ij} \wedge Q_{ik} \wedge S_{ijl} \rightarrow T_{(i+1)(j+D(k,l))}$$
$$S_{ijk} \rightarrow T_{ij} \vee S_{(i+1)jk}$$

即:

$$(T_{ij} \wedge Q_{ik} \wedge S_{ijl}) \vee \overline{T_{(i+1)(j+D(k,l))}} = 1$$
$$S_{ijk} \vee \overline{T_{ij} \vee S_{(i+1)jk}} = 1$$

其中,$i \in \{0, 1, 2, ...p(n) - 1\}, j \in \{-p(n), ...p(n)\}, k \in State, l \in Symbol$

3中子句数量为$O(p^2(n))$
2.2.6中子句数量为$O(p^2(n))$

2.3 NP$\propto$SAT

x为**D**的一个Yes-Instance,当且仅当,

$\exists f$:Instance of **D**$\rightarrow$Instance of **SAT**,s.t,

1). f的转化时间是polynomial size

2). f(x)为**SAT**的一个Yes-Instance

证明:若x是Yes-Instance,按照2.1中的方式表示问题D,记2.2.1-2.2.6中所有子句的集合为$C=\{c_1,c_2,\dots c_m\}$,且$|C|=O(p^3(n))$,构造**SAT**的一个Instance φ_x:其文字集U=$\{\alpha_{-p(n)},\dots\alpha_{-1}\}$,子句集为$C$

1)由于图灵机在$p(n)$步停机,所以c的encoding的长度控制在$p(n)$以内,$\therefore$ $|C||U|=O(p^4(n))$是polynomial size $\therefore$ f将x转化为φ_x的时间为polynomial size

2)由于x是Yes-Instance,$\therefore\exists$c为certificate,s.t.按照算法B=B(x,c)运行的图灵机TM输出Yes,由于TM被子句集C描述,且c的encoding使C中全部语句为True,$\therefore$c是φ_x的一个truth assignment,$\therefore\varphi_x$是**SAT**的一个Yes-Instance.

反之,若x是No-Instance,则x的certificate不存在,那么无论在文字集U中变量取何值,C中的子句都不能完全满足(否则,存在C中子句的一组truth assignment c',使C中子句全为True,那么可以通过C中的子句模拟算法B的运行过程,使算法B输出为True,那么c'是x的一个certificate,矛盾)$\therefore\varphi_x$是**SAT**的一个No-Instance.