Cook定理的证明

陆天航

2022.3.12

定义：设A为判定性问题，称A为NP问题，如果：
存在一个多项式p以及认证算法(certifier)$\mathcal{B}=\mathcal{B}(\cdot,\cdot)$，使得对于$A$的任何一个实例$x$，$x$是一个yes实例当且仅当存在$x$的一个认证(certificate)$c$，使得$|c|\leq p(|x|)$，且$\mathcal{B}(x,c)$在至多$p(|x|)$步内输出yes.

定义：设A_1,A_2为判定性问题，称A_1可多项式转化(polynomially transforms)为A_2，记做$A_1 \propto A_2$，如果：
对于A_1的任何一个实例x，存在多项式p，以及A_2的实例y，使得$|y|\leq p(|x|)$，且x是A_1的yes实例当且仅当y是A_2的yes实例.

定义：设A是一个NP问题，若其他任何NP问题可多项式转化为A，则称A是NP完备的(NP-complete).

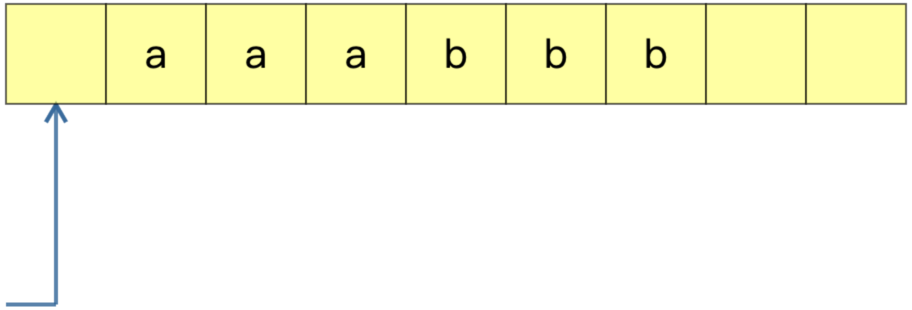
确定性图灵机

定义：称$\mathcal{T}(\Gamma, \Sigma, \#, s, q_a, q_r, h, \delta)$为图灵机(turing machine)，如果：

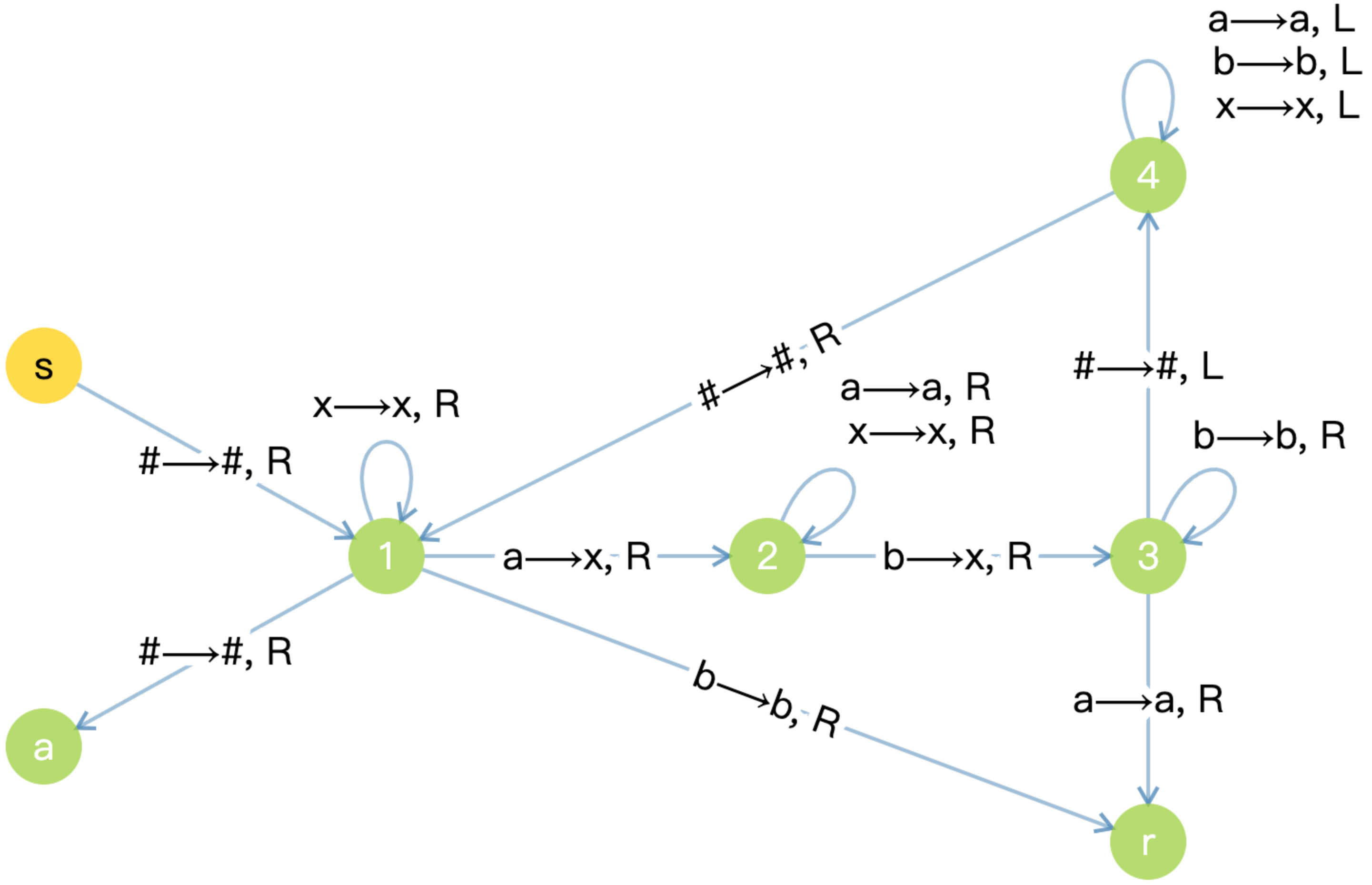
- Γ是有限集合，称为图灵机的状态集(states set).
- Σ是有限集合，称为图灵机的符号集(symbols set).
- $\#$是Σ中的元素，称为空白符号(blank symbol).
- s是Γ中的元素，称为初始状态(initial state).
- q_a是Γ中的元素，称为接受状态(accept state).
- q_r是Γ中的元素，称为拒绝状态(reject state).
- h是Γ中的元素，称为停机状态(halt state).
- $\delta : \Gamma \times \Sigma \longmapsto \Gamma \times \Sigma \times \{L, R\}$称为状态转移函数(transition function).

确定性图灵机

问题：判断一个字符串是否具有形式 $a^n b^n$，n为正整数.



STEP



定理 (Turing, 1936): 设A为NP问题，$\mathcal{B}$为其认证算法，则存在图灵机$\mathcal{T}$，满足对于问题任何实例x，存在纸带上的一个输入$a(x)$，$\mathcal{T}$在$|x|$的多项式步内终止，且若x为yes实例，图灵机$\mathcal{T}$输出接受状态，否则输出拒绝状态.

SATISFIABILITY PROBLEM:

实例：变量集合U和由U组成的句子集C.

问题：是否存在U的一组真值分配，使C中所有句子都为真.

实例I的规模：$|I| = |C||U|$.

例：$U = \{A, B, C, D, E\}$,

$$C = \{A \vee B, \overline{A} \vee C \vee D, \overline{A} \vee E, \overline{B} \vee \overline{C} \vee \overline{E}, C \vee E\}.$$

$I = (U, C)$是一个yes实例.

规模：$|I| = 5 \times 5 = 25$.

认证：$c = \{A = true, B = false, C = true, D = false, E = true\}$.

定理 （Cook）： SAT问题是NP完备的.

任意NP问题D，实例x，认证算法B

↓

SAT问题的实例ϕ_x

↓

x是yes实例 $\Longleftrightarrow$ ϕ_x是yes实例

↓

问题D可多项式转化为SAT问题

1 SAT问题是NP问题

对于SAT问题的任意实例$I = (U, C)$，若I是yes实例，那么I的认证就是I的真值指派，并且存在一个运算时间为$O(|I|)$的认证算法.

如何构造SAT问题的认证算法?

2 任意NP问题认证算法的图灵机表示

考虑任意一个NP问题D，$\mathcal{B}$是认证算法，p是某个多项式，x是D的一个实例，c是规模不超过$p(|x|)$的输入.

$$\text{yes实例x+认证c} \Longrightarrow \text{B}(\text{x}, \text{c})\text{输出yes}$$

$$\text{no实例x+输入c} \Longrightarrow \text{B}(\text{x}, \text{c})\text{输出no}$$

Cook定理

假定给定一组输入(x,c)，算法$\mathcal{B}$可以在某个图灵机$\mathcal{T}$上得到实现。不妨令$\mathcal{B}(x,c)$在q步内终止（q的规模为$O(p(|x|))$）。图灵机$\mathcal{T}$由以下构成：

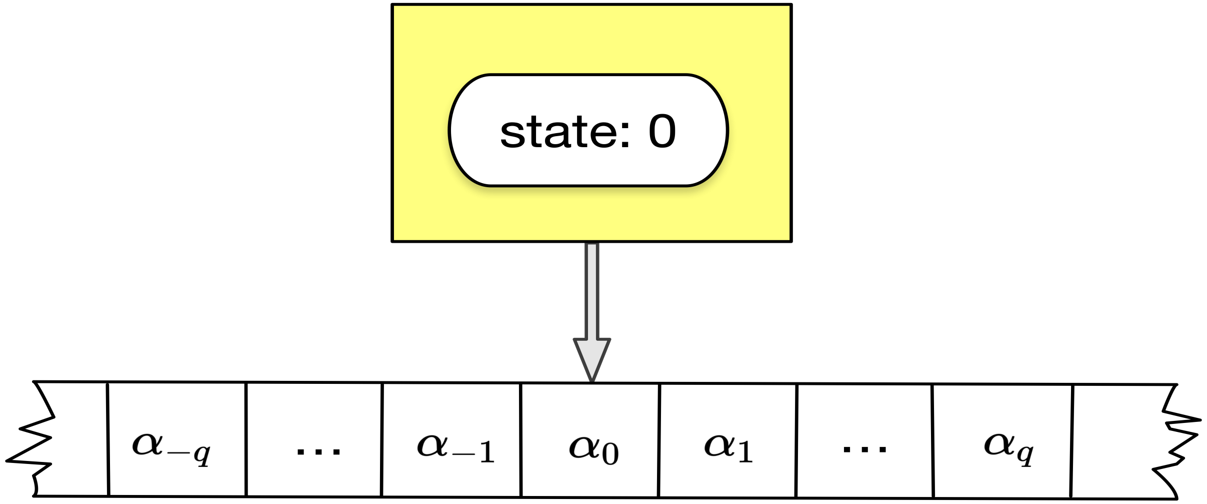
1. 状态集$\Gamma=\{0,1,2,\ldots,q\}$，假定初始状态为$0$，接受状态为$1$，拒绝状态为$2$。
2. 符号集$\Sigma=\{0,1\}$。（这里用0表示空格子）
3. 状态转移函数δ：$\delta:\Gamma\times\Sigma\longmapsto\Gamma\times\Sigma\times\{L,R\}$。

方便起见，将δ拆成三个函数：

$$T:\Gamma\times\Sigma\longmapsto\Gamma$$
$$U:\Gamma\times\Sigma\longmapsto\Sigma$$
$$D:\Gamma\times\Sigma\longmapsto\{L,R\}$$

Cook定理

在纸带上取一个格子表示0号格子，左边的格子编号为$-1,-2,\ldots$，右边的格子编号为$+1,+2,\ldots$．将0号格子作为分隔，在0号格子的左边放置c的编码，右边放置x的编码，并且指针初始和停机时位于0号格子．由于图灵机$\mathcal{T}$的运算时间为q步，所以指针的移动不会超出$-q\sim q$之外．记初始时刻纸带从$-q$到q号格子的符号为$\alpha_{-q},\ldots,\alpha_q$．



图灵机T 的初始状态

3 图灵机运行过程的表示

对于$t=0,1,\ldots,q,\ i=0,1,\ldots,q,\ j=-q,\ldots,q,\ k=0,1$.

Q_{ti}表示在第t步时，图灵机的状态是否为i;

T_{tj}表示在第t步时，指针是否位于格子j;

S_{tjk}表示在第t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(假定图灵机输出接受状态).

3.1 图灵机初始时的状态固定

1. 图灵机初始时处于状态0： $Q_{00}=1$.
2. 指针初始时处于0号格子： $T_{00}=1$.
3. 纸带上初始时有一组符号分配(0号格子左边为c的编码,右边为x的编码)：

$$\begin{cases} S_{0,-q,\alpha_{-q}}=1 \\ \dots \\ S_{0,q,\alpha_q}=1 \end{cases}$$

3.1中句子的数量为$O(p(|x|))$.

3.2 图灵机停机时的状态固定

1. 图灵机停机时处于状态0： $Q_{q0}=1$.
2. 指针在停机时处于0号格子： $T_{q0}=1$.
3. 纸带在停机时0号格子上的符号为1： $S_{q01}=1$.

3.2中句子的数量为$O(1)$.

3.3 每个时刻，图灵机的状态有且仅有一个

$$\begin{cases} Q_{t0} \vee Q_{t1} \vee \dots Q_{tq} = 1, & \forall\, t \in \Gamma, \\ \neg(Q_{ti} \wedge Q_{tj}) = \overline{Q}_{ti} \vee \overline{Q}_{tj} = 1, & \forall\, i \neq j \in \Gamma,\ t \in \Gamma. \end{cases}$$

3.3中句子的数量为$O(p^3(|x|))$.

3.4 每个时刻，指针只在纸带上的其中某一个格子出现

$$\begin{cases} T_{t,-q} \vee T_{t,-q+1} \vee \ldots T_{tq} = 1, & \forall\, t \in \Gamma, \\ \neg(T_{ti} \wedge T_{tj}) = \overline{T}_{ti} \vee \overline{T}_{tj} = 1, & \forall\, i \neq j \in \{-q, \ldots, q\},\ t \in \Gamma. \end{cases}$$

3.4中句子的数量为$O(p^3(|x|))$.

3.5 每个时刻，纸带上的每个格子有且仅有一个符号

$$\begin{cases} S_{ti0} \vee S_{ti1} = 1, & \forall\, i \in \{-q, \ldots, q\},\ t \in \Gamma, \\ \neg(S_{ti0} \wedge S_{ti1}) = \overline{S}_{ti0} \vee \overline{S}_{ti1} = 1, & \forall\, i \in \{-q, \ldots, q\},\ t \in \Gamma. \end{cases}$$

3.5中句子的数量为$O(p^2(|x|))$.

3.6 图灵机状态的变化满足状态转移函数

(a) 状态转移函数T(状态变化)：

$$Q_{ti} \wedge T_{tj} \wedge S_{tjk} \to Q_{t+1,T(i,k)},$$

即，

$$(Q_{ti} \wedge T_{tj} \wedge S_{tjk}) \vee \overline{Q}_{t+1,T(i,k)} = 1,$$

$\forall\, t \in \Gamma,\ i \in \Gamma,\ k \in \Sigma,\ j \in \{-q, \ldots, q\}$.

a中句子的数量为$O(p^3(|x|))$.

（b） 状态转移函数U（符号变化）：

$$Q_{ti} \wedge T_{tj} \wedge S_{tjk} \to S_{t+1,j,U(i,k)},$$

即，

$$(Q_{ti} \wedge T_{tj} \wedge S_{tjk}) \vee \overline{S}_{t+1,j,U(i,k)} = 1,$$

$\forall\, t \in \Gamma,\ i \in \Gamma,\ k \in \Sigma,\ j \in \{-q, \ldots, q\}$.

b中句子的数量为$O(p^3(|x|))$.

(c) 状态转移函数D(位置变化):

$$\begin{aligned} Q_{ti} \wedge T_{tj} \wedge S_{tjk} &\to T_{t+1,j-1}, \qquad & D(i,k) = L; \\ Q_{ti} \wedge T_{tj} \wedge S_{tjk} &\to T_{t+1,j+1}, \qquad & D(i,k) = R, \end{aligned}$$

即,

$$\begin{aligned} (Q_{ti} \wedge T_{tj} \wedge S_{tjk}) \vee \overline{T}_{t+1,j-1} &= 1, \qquad & D(i,k) = L; \\ (Q_{ti} \wedge T_{tj} \wedge S_{tjk}) \vee \overline{T}_{t+1,j+1} &= 1, \qquad & D(i,k) = R, \end{aligned}$$

$\forall\, t \in \Gamma,\ i \in \Gamma,\ k \in \Sigma,\ j \in \{-q, \ldots, q\}$.

c中句子的数量为$O(p^3(|x|))$.

3.6中句子的数量为$O(p^3(|x|))$.

4 Cook定理的证明

记以上所有句子的集合为C，所有变量的集合为U，$\varphi_x = (C, U)$．有$|C| = O(p^3(|x|))$，$|U| = O(p^2(|x|)$，$|\varphi_x| = |C||U| = O(p^5(|x|))$．

若x是yes实例，那么存在c为x的认证，则算法$\mathcal{B}$通过图灵机$\mathcal{T}$得到实现，且图灵机输出接受状态．记录图灵机的运行过程，这个运行过程对应SAT实例φ_x的一组真值指派．这说明φ_x是yes实例．

若x是no实例，那么对于任意规模不超过$O(p(|x|))$的输入c，$\mathcal{B}$输出no，所以图灵机$\mathcal{T}$不可能输出接受状态．这说明φ_x是no实例．

THANKS