

第8章 图灵机

形式化定义

- A TM is described by:
 1. A finite set of *states* (Q , typically).
 2. An *input alphabet* (Σ , typically).
 3. A *tape alphabet* (Γ , typically; contains Σ).
 4. A *transition function* (δ , typically).
 5. A *start state* (q_0 , in Q , typically).
 6. A *blank symbol* (B , in $\Gamma - \Sigma$, typically).
 - u All tape except for the input is blank initially.
 7. A set of *final states* ($F \subseteq Q$, typically).

迁移函数

- Takes two arguments:
 1. A state, in Q .
 2. A tape symbol in Γ .
- $\delta(q, Z)$ is either undefined or a triple of the form (p, Y, D) .
 - p is a state.
 - Y is the new tape symbol.
 - D is a *direction*, L or R.

瞬时描述ID

$\alpha q \beta$: q 指向的是 β 的第一个符号；如果 q 正在扫描左端的 B ，那么 q 处和 q 右侧的连续 B 是 α 的一部分。

推导的符号同PDA： \vdash 和 \vdash^*

图灵机接受的语言

有两种方式，接受状态、停机。从语言上来说，以两种方式定义的语言集合是等价的——递归可枚举语言

接收->停机

Modify M to become M' as follows:

1. For each final state of M , remove any moves, so M' halts in that state.
2. Avoid having M' accidentally halt.
 - u Introduce a new state s , which runs to the right forever; that is $\delta(s, X) = (s, X, R)$ for all symbols X .
 - u If q is not a final state, and $\delta(q, X)$ is undefined, let $\delta(q, X) = (s, X, R)$.

停机->接收

- Modify M to become M'' as follows:
 1. Introduce a new state f , the only final state of M'' .
 2. f has no moves.
 3. If $\delta(q, X)$ is undefined for any state q and symbol X , define it by $\delta(q, X) = (f, X, R)$.

递归语言

一个算法是一个由最终状态接受，并且无论是否接受，都保证停止的图灵机。

但是并不是每一个图灵机都是一种算法，因为图灵机可能不可以停止。

如果 L 是某个作为算法的图灵机 M 以终止状态接受的语言 $L(M)$ ，我们说 L 是一种递归语言。

对于某一种语言，如果可以判断其成员性，那么他就是递归语言。

图灵机的扩展和限制

多磁道：通常用来“标记”，注意每个磁道只能一起动；

状态缓存：在状态中加入缓冲区，将某个符号或串放在缓冲区中；

半无穷的磁带：实际情况下用2条半无穷磁带便于实现“双向无穷”；

多磁带： k 磁带可以用 $2k$ 磁道模拟，不同磁带的磁头可以有不同的移动。注意：讨论P和NP的时候不要多用多磁带。

非确定性图灵机

在每一步的移动有多种可能，可以用确定性图灵机来模拟：类似于广度优先搜索，将磁带看成一个队列，每次如果有 m 种可能，将当前ID复制 m 份在队列尾；对复制的 m 个ID分别做 m 种可能的迁移：如果到达了终止状态，停机并接收；否则返回队列头，做下一次的模拟。

NTM不是一个现实的模型，可以用TM模拟NTM，模拟的时候 k 步可以最多产生 $O(k^n)$ 个ID。但是直接去看一个NTM的时候，可以假设它每一步都猜对，猜 k 步的复杂度是 $O(k)$ 的。

RE和Recursive的封闭性

- Both closed under union, concatenation, star, reversal, intersection, inverse homomorphism.
- Recursive closed under difference, complementation.
- RE closed under homomorphism.

\cup , \cap 用构造图灵机的方法；

连接, $*$: RE可以用NTM来guess, Recursive只能穷举每一种可能

reversal只需要将输入逆转, 逆同态只需要将函数 h 作用于输入 w