CHAPTER 4

上一章已对TM的能力作了初步刻画。我们已说明图灵可识别语言类及图灵可判定语言类远大于正则语言类和CFL类，也看见了TM相比DFA/NFA、PDA的强大之处。此外，我们还讨论了各运算下语言类的封闭性。

本章的核心议题承接上章，希望弄清：

1° TM是否有能力边界？换言之，图灵可识别的语言是否囊括了所有语言？

2° 判定器是否有能力边界？若有，它与TM的能力边界是否重合？

用图来表示，我们已知图灵可判定语言类⊆图灵可识别语言类⊆全体语言，而希望弄清三者边界的关系。

首先，我们引入一种证明工具：对角线方法。先回忆一个经典例子。

Proposition 1 $\mathbb{N} \ncong \mathbb{R}$.

proof. 假设$\mathbb{N} \cong \mathbb{R}$。那么，由于任意实数可唯一对应一串无限长的01串，故$\mathbb{N} \cong \mathbb{R} \cong \{0,1\}^\infty$。从而存在双射$f: \mathbb{N} \to \{0,1\}^\infty$。我们可列举$f$的原象$n \in \mathbb{N}$及象$f(n) \in \{0,1\}^\infty$如下：

n	$f(n)$
1	$b_{11}\ b_{12}\ b_{13}\ b_{14} \cdots$
2	$b_{21}\ b_{22}\ b_{23}\ b_{24} \cdots$
3	$b_{31}\ b_{32}\ b_{33}\ b_{34} \cdots$
4	$b_{41}\ b_{42}\ b_{43}\ b_{44} \cdots$
$\vdots$	$\cdots\ \cdots$

$(\forall i,j: b_{ij} \in \{0,1\})$

然后构造 $s := (\overline{b_{11}}\ \overline{b_{22}}\ \overline{b_{33}}\ \overline{b_{44}} \cdots) \in \{0,1\}^\infty$.

由于f是双射，故$\exists i: f(i) = s$。根据我们的列表规则，$f(i) = (b_{i1}\ b_{i2}\ b_{i3} \cdots)$。但我们又要求$f(i) = s = (\overline{b_{11}}\ \overline{b_{22}}\ \overline{b_{33}} \cdots)$。于是在第$i$位必将产生$b_{ii} = \overline{b_{ii}}$的矛盾。↯

在上面的证明中，我们选取了表中的对角线来构造出新元素s，使得s实质上与每一行 …

都不可能与相同，从而产生矛盾。

那么，「对角线方法」的~~精髓~~精髓在哪？是在于对角线吗？非也。即使我们不选对角线，同样能完成证明：

proof'. ∵ 存在双射 $f: \mathbb{N} \to \{0,1\}^\infty$. 设 $g: \mathbb{N} \to \mathbb{N}$ 是一个满射。我们构造

$$S := (\overline{b_{g(1),1}}, \overline{b_{g(2),2}}, \overline{b_{g(3),3}}, \cdots) \in \{0,1\}^\infty.$$

因 f 是双射，故 $\exists i: f(i) = S$，即

$$\begin{cases} b_{i1} = \overline{b_{g(1),1}} \\ b_{i2} = \overline{b_{g(2),2}} \\ b_{i3} = \overline{b_{g(3),3}} \\ \cdots \end{cases}$$

或者简写成 $b_{ij} = \overline{b_{g(j),j}}$ $(\forall j \in \mathbb{N})$.

取 $j: g(j) = i$，即得到 $b_{ij} = \overline{b_{ij}}$ 的矛盾. ⬛

在这一份反的证明中，满射 g 是随便选取的，例如 $g(n) = \begin{cases} n-1, & n \text{是偶数} \\ n+1, & n \text{是奇数} \end{cases}$ 就是一个合理选择。最终矛盾的焦点不出现在对角线上，而是出现在某个奇特的方位。由此可见：对角线并非「对角线方法」的核心。想要把握其精髓，应当着手分析证明二中蕴含的思想。

首先关注函数 g。我们要求它是一个满射就够了，无需令其为单射。这是因为我们在取 j 的时候，不要求 $j: g(j) = i$ 是唯一的，而只希望它存在即可。

其次关注 S 的构造。我们要求 S 与所有行均有至少一处不同（因为 g 是满射，故 $g(1), g(2), \cdots$ 取遍 $\mathbb{N}$，且有可能重复）。如此一来，当我们将 S 视作表中对象时，就必定产生矛盾。或者说，如此构造出的 S 不应当出现在表中。

由此，我们将「对角线方法」的精髓提炼如下：

1° 提出反证假设 $\mathcal{H}$.

2° 依据 $\mathcal{H}$，构造出一个元素 $a \in A$，使得 a 与集合 A 中任意元素均有至少一处差别，造成矛盾.

~~再度依据H，说明$x\in A$，从而造成矛盾。~~

其中，第2°步的构造往往采用较为直观的「对角线构造」，也就是令满射g为恒等函数，从而将矛盾聚焦在对角线上，方便以图示向别人说清道理。

下面的命题同样用到对角线方法，只不过抽象得多。我们不必拘泥于「对角线」一词不放，而应尽量尝试用上面提炼的要点去理解。

Proposition2 对任意集合S，$S\ncong 2^S$.

proof. $|S|<\infty$的情形是显然的。若$|S|=\infty$，假设$S\cong 2^S$成立。那么存在双射$f:S\to 2^S$。我们现在来构造

$$T:=\{x\in S \mid x\notin f(x)\}\in 2^S$$

又因为f是双射，故$\exists y\in S: f(y)=T$.

从而有2种可能

(1) $y\in f(y)=T$，则据T的定义有$y\notin f(y)$

(2) $y\notin f(y)=T$，则据T的定义有$y\in f(y)$ 矛

remark. 对该证明作一点梳理：

1° 假设H: $S\cong 2^S$. 这等价于假设$\exists$双射$f:S\to 2^S$.

2° 构造$T\in 2^S$. 依据H，便无法有$T\in 2^S$. 产生矛盾。

（作为思考题，考虑将T定义为

$$T:=\{g(x)\mid x\in S \text{ 且 } g(x)\notin f(x)\}$$

其中$g:S\to S$是满射。试由此证得命题。）

图形化地来看，证明中的构造无非是把

$x\in S$	$f(x)\in 2^S$ $x_1\in S?$	$x_2\in S?$	$x_3\in S?$	……
x_1	0/1	0/1	0/1	
x_2	0/1	0/1	0/1	…
x_3	0/1	0/1	0/1	
x_4	0/1	0/1	0/1	⋱
⋮		…		

（0 = 不选
1 = 选）

对角线上的0/1翻转过来罢了。原本为1的（即$x\in f(x)$的），被排除在T以外；原本为0的（即$x\notin f(x)$的），被包含在T之内。如此构造的T，当然与$f(S)$中任一个集合均不同，故$T\notin f(S)\overset{H}{=}2^S$，矛盾。

翻来覆去地讲了这么多对角线方法，我们总算来到正题。我们将用该方法证明一个特别的可识别的语言是不可判定的。

def $A_{TM} := \{\langle M,w\rangle \mid M$是一台TM，$w\in\mathcal{L}(M)\}$

其中$\langle M,w\rangle$指的是M与w的编码。注意M与w都是变量。

Theorem 3 A_{TM}是图灵可识别的。

proof. 构造TM N，它能解析任意输入串$\langle M,w\rangle$，并依照M的规则去计算w。若某时刻M进入了接纳状态，则N接纳。■

remark. 注意N是在模拟M的行为，而非调用M —— 模拟意即M的所有状态都是虚拟出来的，而非物理嵌入于N里面。

Theorem 4 A_{TM}不是图灵可判定的。（简称不可判定。）

proof. 假若A_{TM}可判定，那么存在判定器N：$\mathcal{L}(N)=A_{TM}$. 我们现在来构造一个TM N'（实际上是判定器），其输入为任意TM M的编码。它将调用N并据此产生自身的输出。

$$N'(\langle M\rangle) := \begin{cases} 接纳. & 若N(\langle M,\langle M\rangle\rangle)拒绝 \\ 拒绝. & 若N(\langle M,\langle M\rangle\rangle)接纳 \end{cases}$$

现考虑将N'自身的编码输入N'，则

$$N'(\langle N'\rangle) = \begin{cases} 接纳. & 若N(\langle N',\langle N'\rangle\rangle)拒绝 \\ 拒绝. & 若N(\langle N',\langle N'\rangle\rangle)接纳 \end{cases}$$

$$\overset{据N}{\underset{的定义}{=}} \begin{cases} 接纳. & 若\langle N'\rangle\notin\mathcal{L}(N') \\ 拒绝. & 若\langle N'\rangle\in\mathcal{L}(N') \end{cases}$$

$$= \begin{cases} 接纳. & 若N'拒绝\langle N'\rangle \\ 拒绝. & 若N'接纳\langle N'\rangle \end{cases}$$

remark. 图形化地来看，

机器M \ 输入⟨M⟩	⟨M₁⟩	⟨M₂⟩	⟨M₃⟩	⟨M₄⟩	...
M_1	a/r/l	a/r/l	a/r/l	a/r/l	
M_2	a/r/l	a/r/l	a/r/l	a/r/l	
M_3	a/r/l	a/r/l	a/r/l	a/r/l	
M_4	a/r/l	a/r/l	a/r/l	a/r/l	
⋮					

(a=accept
r=reject
l=loop)

我们构造的N'在面对每种输入$\langle M_i\rangle$时表现均与$M_1, M_2, \ldots$不同，从而N'无法出现在表中，与N'是TM矛盾。当然，为了

保证N'真的能与$M_1, M_2, \cdots$不同，必须借助于N的「超能力」，把迷途的情形纠出来，以防N'同样地也迷途了。

problem

请考虑该定理另外的证法。

(提示：将所有{0,1}上的TM编号为$M_1, M_2, \cdots$；将所有01串编号为$w_1, w_2, \cdots$。仍用对角线方法。)

由Theorem 3.4，我们知道：可识别未必可判定，于是 可判定语言类 $\subset$ 可识别语言类。

判定一个问题（即永远给出非黑即白的回答）確比识别一个问题（即仅给出肯定回答）困难。

Theorem 5 若语言A与$\overline{A}$均可识别，则A可判定.

proof. 因为A与$\overline{A}$均可识别，故存在TM M_1与M_2：$\mathcal{L}(M_1) = A$，$\mathcal{L}(M_2) = \overline{A}$。从而

$$\forall w, \begin{cases} w \in A \Leftrightarrow M_1 \text{接纳} w \\ w \notin A \Leftrightarrow M_2 \text{接纳} w. \end{cases}$$

于是可制造出判定器M：

1° 轮番运行M_1与M_2

2° 若M_1某时刻接纳，则接纳；
若M_2某时刻接纳，则拒绝.

显然M能够终止，不会迷途，故M是判定器，且$\mathcal{L}(M) = A$. ■

由Theorem 3.4.5可推知：A_{TM}必定不可识别。这也便证明了可识别语言类 $\subset$ 全体语言。至此，本章的核心议题得以完满解决。

作为补充，我们来讨论一下图灵可识别语言类与全体语言的差距有多大。给定字符集Σ，全体语言

$$U = 2^{\Sigma^*}$$

而图灵可识别语言类

$$\begin{aligned} TMR &\preceq \{M \mid M\text{是图灵机}\} \\ &\cong \{\langle M\rangle \mid M\text{是图灵机}\} \\ &\cong \{0,1\}^* \\ &\cong \Sigma^* \end{aligned}$$

因此 ~~A_{TM}~~ TMR $<$ U，且二者之间鸿沟巨大。

(类似于 $\mathbb{N}$ 与 $\mathbb{R}$ 的差距).

本章末尾，我们介绍几类在理论上重要的问题。

1° 接纳问题 (A)：给定机器 M 和输入 w，问 M 是否接纳 w？

2° 空语言问题 (E)：给定机器 M，问 $\mathcal{L}(M)$ 是否为 $\emptyset$？

3° 等价问题 (EQ)：给定机器 M_1 与 M_2，问 $\mathcal{L}(M_1)$ 是否等于 $\mathcal{L}(M_2)$？

我们前面已讨论过图灵机的接纳问题 A_{TM}。那么 A_{DFA}、A_{NFA}、A_{PDA} 等等问题如何呢？下表给出了解答。

	DFA/NFA	PDA	TM
A	可判定	可判定	可识别
E	可判定	可判定	可识别
EQ	可判定	可识别	可识别

其中 $A_{DFA/NFA}$，$E_{DFA/NFA}$，A_{PDA} 是简单的，证明留作习题。

Proposition 6 E_{PDA} 可判定。

proof. 构造判定器 M 如下。读入 $\langle P \rangle$ 时，

1° 将 PDA P 转换为 CFG G (见 Chap 2)

2° 标记 G 的所有终结符

3° 重复以下步骤直至无法再更新：

对于任意规则 $A \to w$，若 A 未被打标记，且 w 中每个变量与终结符均有标记，则给 A 打上标记。

4° 若初始变量无标记，则接纳；否则，拒绝。

一言以蔽之，M 通过「自底向上」方法判定 S 能否逐步展开为终结符串。■

Proposition 7 $EQ_{DFA/NFA}$ 可判定。

proof 构造判定器 M 如下。读入 $\langle N_1, N_2 \rangle$ 时，

1° 若 N_1, N_2 是 NFA，则将其转换为 DFA M_1, M_2.

2° 构造 DFA D，使

$$\mathcal{L}(D) = [\mathcal{L}(M_1) \cap \overline{\mathcal{L}(M_2)}] \cup [\overline{\mathcal{L}(M_1)} \cap \mathcal{L}(M_2)]$$

3° 判定$L(D)$?$\emptyset$。若是，则意味着$L(M_1)=L(M_2)$，接纳；若否，则意味着$L(M_1)\neq L(M_2)$，拒绝。∎

remark. 该证明不能移植到EQ_{PDA}上，因为CFL对于$\cap$、补操作不封闭。

至于EQ_{PDA}，E_{TM}以及EQ_{TM}的不可识别性，需要借助下一章的技巧方得证明，此处暂不讨论。

由表中可以看出：越是通用的模型，解决A、E、EQ问题就越是困难。而通常而言，解决A问题是相对容易的，解决EQ问题是最困难的。正如常识告诉我们的一样，限定愈强，搜索空间愈小，问题愈简单。

了解到问题的可识别性/可判定性是十分有益的。一方面，它能避免我们在某些不可计算的问题上耗散精力，也能指导我们对过难的问题作出简化。