第4章 可判定性

朴秀峰

xfpiao@126.com

主要内容

4.1 可判定语言

4.1.1 与正则语言相关的可判定问题

4.1.2 与上下文无关语言相关的可判定问题

4.2 停机问题

4.2.1 对角化方法

4.2.2 停机问题是不可判定的

4.2.3 一个图灵不可识别语言

与正则语言相关的可判定问题

$A_{DFA} = \{ <B, w> \mid B$ 是 DFA，w 是串，B 接受 $w \}$

定理 4.1

A_{DFA}是一个可判定语言。

设计一个判定 A_{DFA} 的 TM M。

M = “ 对于输入 < B, w >，其中 B 是 DFA，w 是串：

1) 在输入w上模拟B。

2) 如果模拟以接受状态结束，则接受；
   如果以非接受状态结束，则拒绝。”

A_{DFA}是一个可判定语言

- **Step 1:** 检查输入 <***B***, ***w***>，它表示输入串 ***w*** 和 **DFA** ***B***。

  B 的一个合理的表示方法是简单的列出它的五个元素 **Q, Σ, δ,** q_0 及 **F**。当 ***M*** 收到输入时，首先检查它是否正确的表示了 **DFA** ***B*** 和串 ***w***。如果不是，则拒绝

- **Step 2:** ***M*** 直接执行模拟。用在带上写下信息的方法，它可以跟踪 ***B*** 在输入 ***M*** 上运行时的当前状态和当前位置。

  运行开始时，***B*** 的当前状态时 q_0，读写头的当前位置是 ***w*** 的最左端符号。状态和位置的更新是由转移函数决定的。

  当 ***M*** 处理完 ***w*** 的最后一个符号时，如果 ***B*** 处于接收状态，则 ***M*** 接受这个输入；如果不是，则 ***M*** 拒绝。

与正则语言相关的可判定问题

$A_{\text{NFA}} = \{ <B, w> \mid B$ 是 NFA，w 是串，B 接受 $w \}$

定理 4.2

A_{NFA}是一个可判定语言。

构造一个判定 A_{NFA}的 TM N。用 M 作为 N 的子程序。

N= “对于输入$<B, w>$，其中 B 是 NFA，w 是串：

1) 将 NFA B 转换成一个等价的 DFA C。

2) 在输入$<C, w>$ 上运行 TM M。

3) 如果 M 接受，则接受，否则拒绝。”

与正则语言相关的可判定问题

A_{REX}={ <R,*w*> | R是正则表达式， *w*是串， *R* 派生*w*}

定理 4.3

A_{REX} 是一个可判定语言。

P= “对于输入 <*R*, *w*> 上，其中 R 是正则表达式，*w* 是串：

1) 将正则表达式 *R* 转换成一个等价的 NFA *A*。

2) 在输入 <*A*, *w* >上运行 TM *M*。

3) 如果 *N* 接受，则接受，否则拒绝。”

有穷自动机的空性质测试

$E_{\mathrm{DFA}} = \{ \langle A \rangle \mid A$ 是 DFA，且 $L(A) = \varnothing \}$

定理 4.4　E_{DFA} 是一个可判定语言。

T= “对于输入 $\langle A \rangle$，其中 A 是 DFA：

1) 标记 A 的起始状态。
2) 重复下列步骤，直至所有状态都被标记。
3) 对于一个状态，如果有一个到达它的转移是从某个已经标记过的状态出发的，则将其标记。
4) 如果没有接受状态被标记，则接受，否则拒绝。”

与正则语言相关的可判定问题

$EQ_{\mathrm{DFA}} = \{ <A, B> \mid A$ 和 B 都是 DFA，且 $L(A)=L(B) \}$

定理 4.5 EQ_{DFA}是一个可判定语言。

构造 DFA C，使得 $L(C) = (L(A) \cap \sim L(B)) \cup (\sim L(A) \cap L(B))$

$L(C) = \varnothing$ 当且仅当 $L(A)=L(B)$

F= “对于输入$<A, B>$，其中 A 和 B 都是 DFA：

1) 构造 DFA C。

2) 再输入 $<C>$ 上运行定理4.4中的 TM T。

3) 如果 T 接受，则接受，否则拒绝。”

与上下文无关语言相关的可判定问题

A_{CFG}={ <G, w> | G 是 CFG，w 是串，G 派生 w }

定理 4.6 A_{CFG}是一个可判定语言。

如果 G 是乔姆斯基范式，w 的任意派生都是 $2n-1$ 步，其中 n 是 w 的长度。（见书上的问题2.26）

S= “对于输入 <G, w>上，其中 R 是一个 CFG，w 是串：

1) 将 G 转换成与之等价的乔姆斯基文法。

2) 列出所有 $2n-1$ 步的派生，其中n 是 w 的长度，除非 $n=0$，此时列出一部步以内的派生。

3) 如果这些派生中有一个产生 w，则接受，否则拒绝。”

与上下文无关语言相关的可判定问题

E_{CFG} = { <G> | G 是 CFG， 且 $L(A) = \varnothing$ }

定理 4.7 E_{CFG}是一个可判定语言。

要点：检查每一个变元，以确定该变元能否产生终极符。

R= “ 对于输入 <G> 上，其中 G 是一个 CFG：

1) 将 G 中所有的终结符都作上标记。
2) 重复下列步骤，直至找不到可以作标记的变元。
3) 如果 G 有规则 $A \rightarrow U_1U_2 \ldots U_K$，且 $U_1, U_2, \ldots, U_K$ 中的每个符号都已被作过标记，则将变元 A 作标记。
4) 如果起始变元没有被标记，则接受，否则拒绝。”

与上下文无关语言相关的可判定问题

定理 4.8 每个上下文无关语言都是可判定的。

- 设 G 是 A 的一个 **CFG**。设计一个判定 A 的 **TM** M_G，它在自己内部建立 G 的一个备份。其工作方式如下：

M_G = " 对于输入 w:

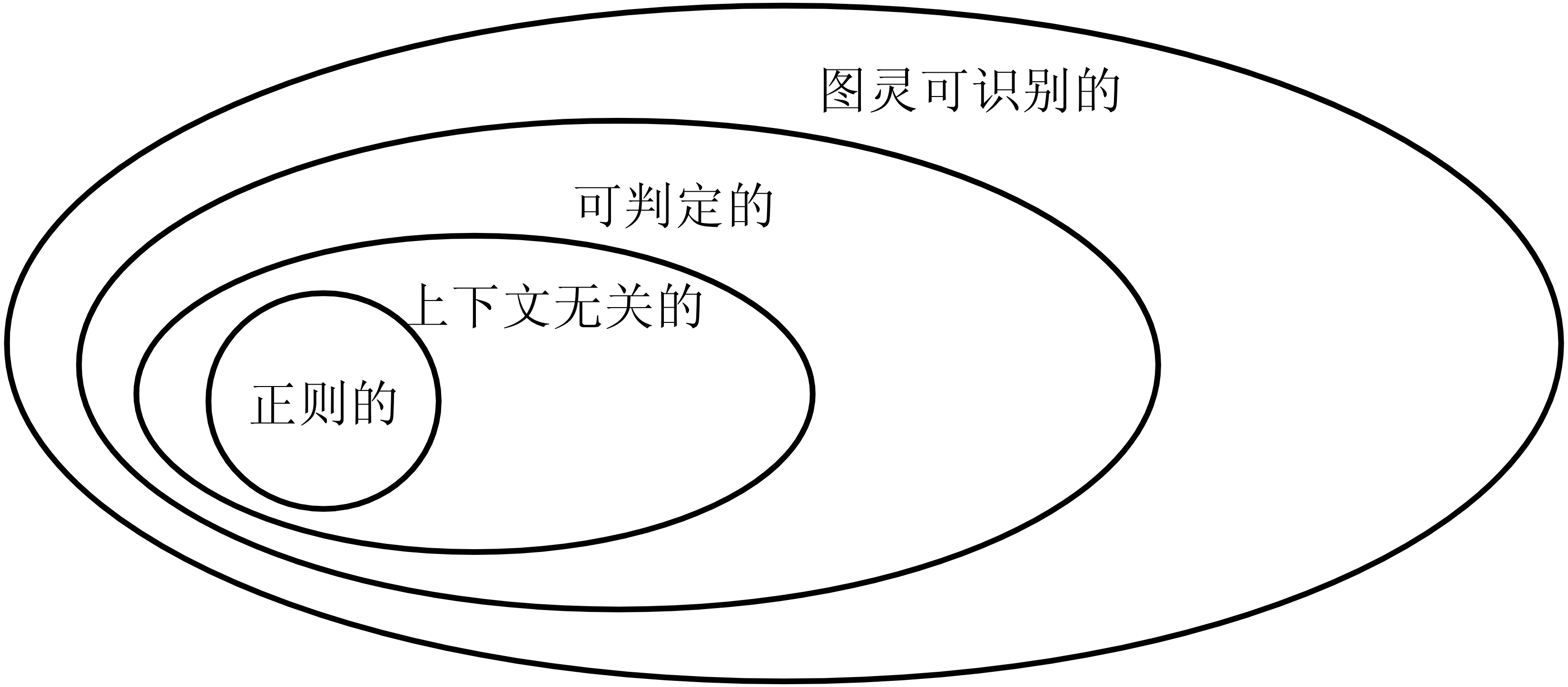
1) 在输入 $\langle G, w \rangle$ 上运行 **TM** S。

2) 如果该机器接收，则接受；否则拒绝。"

与上下文无关语言相关的不可判定问题

EQ_{CFG} = { <G, H> | G 和 H 是 CFG，且 $L(\mathrm{G}) = L(\mathrm{H})$ }

四个语言类



主要内容

4.1 可判定语言

- 4.1.1 与正则语言相关的可判定问题
- 4.1.2 与上下文无关语言相关的可判定问题

4.2 停机问题

- 4.2.1 对角化方法
- 4.2.2 停机问题是不可判定的
- 4.2.3 一个图灵不可识别语言

停机问题

A_{TM}={ <M, w> | M 是一个 TM，且接受 w }

定理 4.9 A_{TM} 是不可判定的。

U= “ 对于输入 <M, w>上，其中 M 是一个 TM，w 是串：

1) 在输入 w 上模拟 M。

2) 如果 M 进入接受状态，则接受；如果 M 进入拒绝状态，则拒绝。”

如果 M 在 w 上循环，则机器 U 在输入 <M, w>上循环。

因此，U 不能判定 A_{TM}。

对角化方法

定义 4.10

设 A 和 B 是两个集合，f 是从 A 到 B 的函数。如果 f 从不将 两个不同元素映射到同一个对象，即：只要 $a \neq b$ 就有 $f(a) \neq f(b)$，则称 f 是一对一映射的。

如果 f 能击中 B 的每个元素，即：对 B 的每个元素 b，都存在 $a \in A$，使得 $f(a) = b$，则称 f 是满映射。

如果存在函数 f：$A \rightarrow B$，f 是一对一映射又是满映射，则称集合 A 和 B 有相同规模。

而既是一对一映射又是满映射的函数称为对应。

在对应中，A 的每个元素映射到 B 的唯一一个元素，且 B 的每个元素都有A的唯一一个元素映射到它。对应就是将 A 的元素与 B 的元素进行配对的方法。

自然数集合和偶数集合

例4.11 设 N 是自然数集合{1, 2, 3, …}，ε是偶自然数集合{2, 4, 6, …}。用康托的关于进和规模的定义可以 看到：

N 和 ε 有相同的规模。

从 N 映射到 ε 的对应 f 是 $f(n)=2n$。

可数集合

定义 4.12

如果一个集合 **A** 是有限的或者与 **N** 有相同的规模，则 **A** 是可数的。

有理数集合和自然数集合

例4.13 设 $\mathbf{Q} = \{m/n \mid m,n \in \mathbf{N}\}$是正有理数集合，$\mathbf{Q}$ 和 $\mathbf{N}$ 的规模相同。

对角化方法

还是有—些集合，因为它们太大了，故没有与**N**的对应。这样的集合称为不可数的。

实数集是一个不可数集合的例子。实数是用十进制表示的数。数**n＝3.1415926…**和√**2＝1.414235…**都是实数。设**R**是实数集合。康托证明了**R**是不可数的。下面用对角线方法证明之。

对角化方法

定理 4.14

R 是不可数的。

证明：假设在**R**与**N**间存在着对应**f**，现在的任务是证明它没有应有的性质。因为它是一个对应，必须能将**N**的所有元素与**R**的所有元素进行配对。如果能找到**R**中的一个**x**，它和**N**中的任何元素都不能配对，则找到矛盾。

为此，实际构造出这样一个**x**。方法为：在选择它的每一位数时，都使得**x**不同于某个实数，且此实数已与**N**中的一个元素配对。这样就能保证**x**不同于任何已配对的实数。

对角化方法

定理 4.14　$\boldsymbol{R}$ 是不可数的。

用一个例子来说明这个思路。假设对应**f**存在，且设**f(1)=3.14159…**，**f(2)=55.55555…**，**f(3)=…**等等。则**f**将自然数**1**与**3.14159…**配对，将**2**与**55.55555…**配对，依此类推。表**4-2**给出了此假定存在的**f**的一些值，**f**联系了**N**和**R**。

只要给出**x**的十进制表示，则**x**就可以构造出来。所构造的**x**是在**0**与**1**之间的 一个数，所以重要的是小数点后面的数字。要保证对每个**n**都有**x≠f(n)**。为保证**x≠f(1)**，只要保证**x**的第一位小数不同于**f(1)**＝**3.14159…**的第一位小数，即不

对角化方法

定理 4.14　R 是不可数的。

是数字**1**，随意地令它为**4**。为保证**x**$\neq$**f(2)**，只要保证**x**的第二位小数不同于**f(2)**＝**55.555555….**的第二位小数。即不是数字**5**，任意地令它为**6**。**f(3)**＝**0.12345…**的第三个小数是**3**，故可取**x**的第三位小数是任一个不为**3**数字，比如**4**。沿着表**4-2**中的对角线，以这种方法继续下去，就能够得到**f**的所有数字。如表**4-3**所示。不难知道，对任意**n**，**x**都不是**f(n)**，因为**x**与**f(n)**在第**n**个小数位上不同。

对角化方法

定理 4.14 $\boldsymbol{R}$ 是不可数的。

上述定理对计算理论有着重要的应用，它表明有些语言是不可判定的．甚至不是图灵可识别的，原因是：有不可数个语言，却只有可数个图灵机。由于一个图灵机只能识别一个语言，而语言比图灵机更多，故有些语言不能用任何的图灵机识别。这样的语言就不是图灵可识别的，正如下面推论所说。

对角化方法

推论 4.15　存在不能被任何图灵可识别的语言。

证明：为证明所有图灵机构成的集合是可数的，首先证明：对仟意的字母表Σ，其上所有串的集合Σ*是可数的。这是因为，对每个自然数**n**，长度为**n**的串只有有限多个，我们先写下长度为**0**的所有串，再写下长度为**1**的所有串。再写下长度为**2**的所有串，依此类推，这样就能构造出Σ的序列。

由所有图灵机构成的集合是可数的，原因是：每个图灵机有一个编码．它是一个串<**M**>。只要去掉那些不是图灵机合法编码的串，就得到了所有图灵机的序列。

对角化方法

推论 4.15 存在不能被任何图灵可识别的语言。

为证明由所有语言构成的集合是不可数的，首先证明由所有无限二进制序列构成的集合是不可数的。所谓的无限二进制序列是指由**0**或**l**构成的无限序列。以**β**记所有无限二进制序列构成的集合，可以通过对角化方法来证明**β**是不可数的。此法类似于定理**4.14**所用的方法，只不过那时是证明**R**是个可数的。

设**£**是字母表Σ所有语言的集合。只要给出**£**与Σ的一个对应，就证明了此这两个集合有相同的规模，也就证明**£**是不可数的。设Σ* =**{s1，s2，s3，….}**。每个语言**A** ∈ **£**在**β**中都

对角化方法

推论 4.15 存在不能被任何图灵可识别的语言。

有唯一的一个相应序列：如果**si** ∈ **A**，则此序列的第**i**位为**1**，如果**si** ∈（错）**A**，则此序列的第**i**位为**0**。此序列被称为**A**的特征序列。例如，如果**A**是字母表{**0**，**1**}上以**0**开始的串构成的语言，则其特征序列X_A是：

$\varepsilon\Sigma^*$ ={ ， **0**， **1**， **00**， **01**， **10**， **11**， **000**， **001**， **….**}

A={ **0**， **00**， **01**， **10** **000**， **001**， **….**}

X_A = **0 1 0 1 1 0 0 1 1….**

令函数集： **£** → **β**为；**f(A)**是**A**的特征序列，则**f**是一对一且满射的，即是一个对应。因为**β**是不可数的，故**£**也是不可数的。

停机问题是不可判定的

A_{TM}={ $<M, w>$ | M 是一个 TM，且接受 w }

定理 4.9 A_{TM} 是不可判定的。

A_{TM}是可识别的。

U= “ 对于输入 $<M, w>$上，其中 M 是一个 TM，w 是串：

1) 在输入 w 上模拟 M。

2) 如果 M 进入接受状态，则接受；

如果 M 进入拒绝状态，则拒绝。”

如果 M 在 w 上循环，则机器 U 在输入 $<M, w>$上循环。

因此，U 不能判定 A_{TM}。

停机问题是不可判定的

现在证明下列语言的不可判定性：

A_{TM} ={ <M, *w*> | M 是一个 TM，且 M 接受 *w* }

假没 A_{TM} 是可判定的，下面将由之导出矛盾。

设 **H** 是 A_{TM} 的判定器。令 **M** 是一个 **TM**，*w* 是一个串。

在输入<**M**, *w*>上，如果 **M** 接受 *w*，则 **H** 就停机且接受 *w*;

如果 **M** 不接受 **w**，则 **H** 也会停机，但拒绝 **w**。

换句话说，**H** 是一个 **TM** 使得：

$$H(<M,w>)=\begin{cases}\text{接受} & \text{如果} M \text{ 接受 } w\\ \text{拒绝} & \text{如果} M \text{ 不接受 } w\end{cases}$$

停机问题是不可判定的

现在来构造一个新的图灵机 **D**，它以 **H** 作为了程序。

当 **M** 被输入它自己的描述 <**M**> 时，**TM D** 就调用 **H**，以了解 **M** 将做什么。

一旦得到这个信息，**D**就反着做，

即：如果**M**接受，它就拒绝；如果**M**不接受，它就接受。

下面是 **D** 的描述。

D＝ "对于输入**<M>**，其中 **M** 是 一 个 **TM**：

1) 在输入**<M, <M>>**上运行 **H**。

2) 输出 **H** 输出的相反结论，即，

如果 **H** 接受，就拒绝；如果 **H** 拒绝，就接受。"

停机问题是不可判定的

"在一个机器上运行它自己的描述"类似于以一个程序本身作为输入来运行这个程序，是有意义的。总而言之，

$$D(<M>) = \begin{cases} \text{接受} & \text{如果}\, M\, \text{不接受} <M> \\ \text{拒绝} & \text{如果}\, M\, \text{接受} <M> \end{cases}$$

当以 **D** 的描述 **<D>** 作为输入来运行 **D** 自身时，结果是：

$$D(<D>) = \begin{cases} \text{接受} & \text{如果}\, D\, \text{不接受} <D> \\ \text{拒绝} & \text{如果}\, D\, \text{接受} <D> \end{cases}$$

这显然是一个矛盾。所以，**TM D** 和 **TM H** 都不存在。

停机问题是不可判定的

‘Acceptance behavior’ of M_i on $<M_j>$

图灵机　　输入串

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$
M_1	accept		accept		
M_2	accept	accept	accept	accept	
M_3					$\cdots$
M_4	accept	accept			
$\vdots$			$\vdots$		$\ddots$

停机问题是不可判定的

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$
M_1	accept	reject	accept	reject	
M_2	accept	accept	accept	accept	
M_3	reject	reject	reject	reject	$\cdots$
M_4	accept	accept	reject	reject	
$\vdots$			$\vdots$		$\ddots$

'Deciding behavior' of G on $<M_i,<M_j>>$，拟用对角线上反码构造图灵机D

停机问题是不可判定的

相反　相反　相反　相反

	$\langle M_1\rangle$	$\langle M_2\rangle$	$\langle M_3\rangle$	$\langle M_4\rangle$	…	$\langle D\rangle$	…
M_1	accept	reject	accept	reject			
M_2	accept	accept	accept	accept			
M_3	reject	reject	reject	reject	…		
M_4	accept	accept	reject	reject			
⋮			⋮		⋱		
D	reject	reject	accept	accept	…		

D也是一切合乎条件的TM 中的一个，所以也在其中，

下页

停机问题是不可判定的

相反 相反 相反 相反

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$	$\langle D \rangle$	$\cdots$
M_1	*accept*	*reject*	*accept*	*reject*			
M_2	*accept*	*accept*	*accept*	*accept*			
M_3	*reject*	*reject*	*reject*	*reject*	$\cdots$		
M_4	*accept*	*accept*	*reject*	*reject*			
$\vdots$			$\vdots$		$\ddots$		
D	*reject*	*reject*	*accept*	*accept*	$\cdots$	$\boxed{?}$	
$\vdots$							

Contradiction for D on input <D>. 接受、拒绝都矛盾

一个图灵不可识别语言

- A_{TM}是一个不可判定的语言。但它是图灵可识别的。
- 现在介绍另一个语言，此语言甚至是不可识别的。
- 可以证明：如果一个语言和它的补都是图灵可识别的，则此语言也是可判定的。
- 这样，对任何不可判定语言，它或它的补至少有一个不是图灵可识别的。
- 一个语言的补是由不在此语言中的所有串构成的语言。
- 如果一个语言是一个图灵可识别语言的补集，则称它是补图灵可识别的 (co-TM recognizable)。

一个图灵不可识别语言

定理 4.16 一个语言是可判定的，当且仅当它既是图灵可识别的，也是补图灵可识别的。

$\Rightarrow$ 如果 **A** 是可判定的，很容易看出 **A** 和它的补 $\bar{\mathbf{A}}$ 都是图灵可识别的，因为任何可判定语言都是图灵可识别的，且任何可判定语言的补也是可判定的。

$\Leftarrow$ 如果 **A** 和 $\bar{\mathbf{A}}$ 都是图灵可识别的，

令 $\mathbf{M_1}$ 是 **A** 的识别器， $\mathbf{M_2}$是 $\bar{\mathbf{A}}$ 的识别器。

下列图灵机 **M** 是**A** 的判定：

M＝ " 对于输入 w:

1) 在输入 w 上并行运行 $\mathbf{M_1}$ 和 $\mathbf{M_2}$。

2) 如果 $\mathbf{M_1}$ 接受，就接受；如果 $\mathbf{M_2}$ 接受，就拒绝。"

一个图灵不可识别语言

定理 4.16 一个语言是可判定的，当且仅当它既是图灵可识别的，也是补图灵可识别的。

并行地运行两个机器指的是：**M** 有两个带，一个模拟 $\mathbf{M_1}$，另一个模拟 $\mathbf{M_2}$。此时 **M** 交替的模拟两个机器的一步，一直持续到其中之一接受。

现在证明 **M** 确实判定 **A**。任一个串 w 要么在 **A** 中，要么在 $\bar{\mathbf{A}}$中。所以，$\mathbf{M_1}$ 和 $\mathbf{M_2}$ 必定有一个接受 w。因为只要 $\mathbf{M_1}$ 或 $\mathbf{M_2}$ 接受，**M** 就停机，所以 **M** 总会停机，因而它是个判定器。

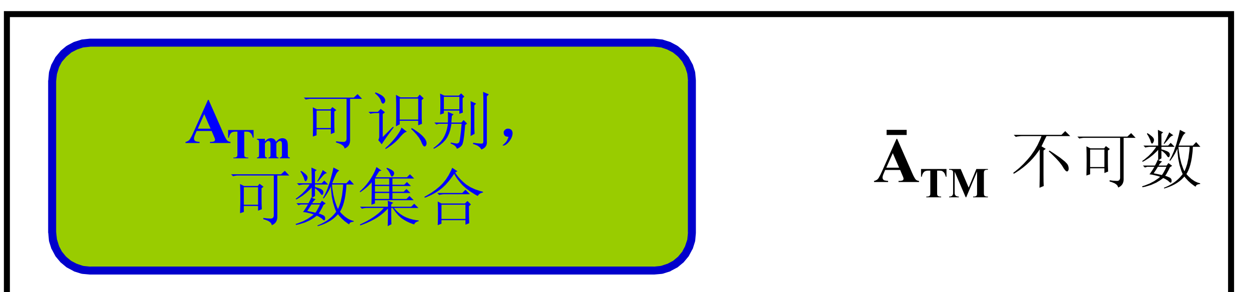
此外，**M** 接受所有在 **A** 中的串，拒绝所有不在 **A** 中的串，故 **M** 是 **A** 的判定器，因而 **A** 是可判定的。

一个图灵不可识别语言

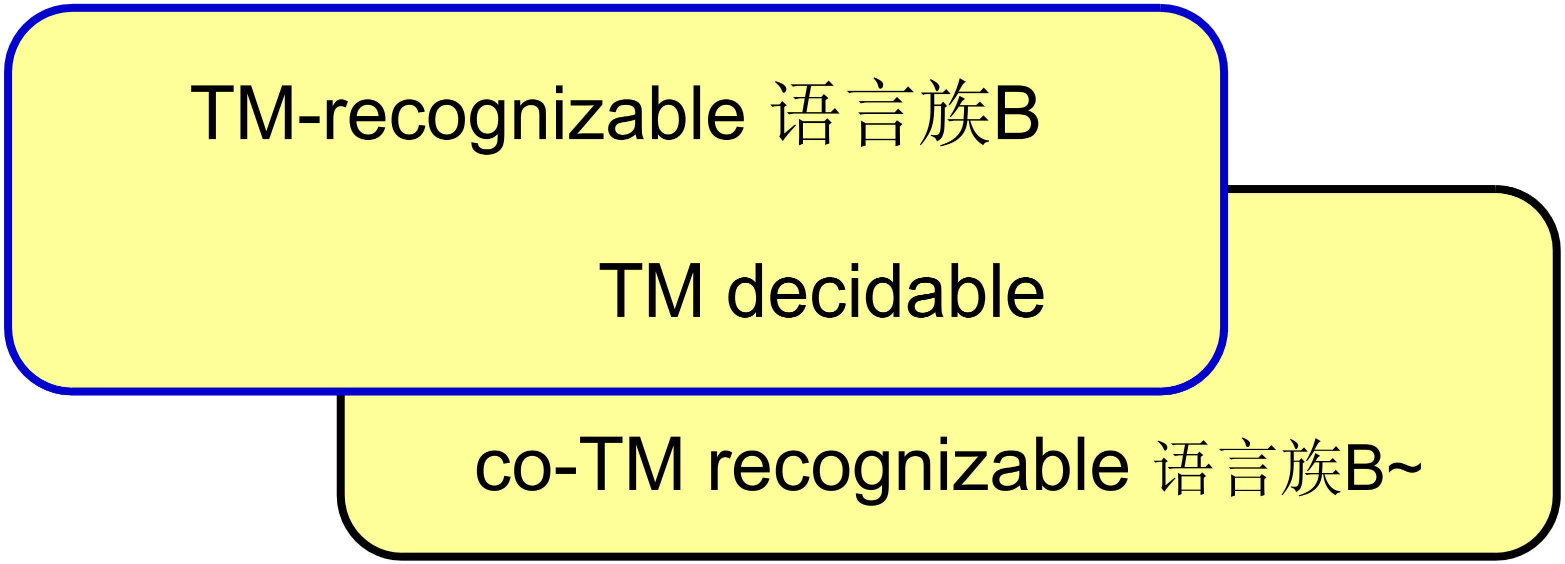
推论 4.17 $\bar{A}_{TM}$不是图灵可识别的。

反证法。因为 A_{TM} 是图灵可识别的。如果$\bar{A}_{TM}$也是图灵可识别的，则 A_{TM} 将是可判定的。但定理**4.9** 说 A_{TM} 不是可判定的，所以 $\bar{A}_{TM}$肯定不是图灵可识别的。

直观理解。语言总集不可数，可识别的集合 **A** 是可数集合，其补集是不可数的，集合太大，当然不可识。



其它问题



$E_{TM} = \{ \langle M \rangle \mid M$ 是一个TM，且$L(M)=\varnothing \}$ 空问题

$EQ_{TM} = \{ \langle M_1, M_2 \rangle \mid M_1$和$M_2$都是TM，且 $L(M_1)=L(M_2) \}$ 相等问题

本章小结

作业

- **4.1, 4.3, 4.6, 4.9, 4.12**