第九讲 （不）可计算问题

罗 欢

Email：hluo@fzu.edu.cn

(不)可计算问题

第4章 丘奇-图灵论题
第5章 可判定性
第6章 可归约性
第7章 可计算性理论的高级专题

第5章　可判定性

- 可判定语言
  - 关于正则语言的可判定问题
  - 关于上下文无关语言的可判定问题
- 停机问题
  - 对角化方法
  - 停机问题是不可判定的
  - 非图灵可识别语言

第6章 可归约性

- 语言理论中的不可判定问题
  - 利用计算历史的归约
  - 线性界限自动机
- 波斯特对应问题
- 映射归约(多一归约,m归约)

算法可解性的局限

- 算法可解
  - 存在处处停机的算法
- 按照算法可解性给问题分类
  - 证明有些问题能用算法求解
  - 证明另一些问题不能用算法求解
- 研究不可解性的意义
  - 避免做无用功
  - 激发想象力,全面透彻地理解什么是计算

可判定语言

- 问题与语言
  - 编码
- 可判定性
  - 存在处处停机的TM程序

关于正则语言的
可计算问题

关于正则语言的可判定问题

- DFA接受性问题
- NFA接受性问题
- 正则表达式派生性问题
- DFA空性问题
- DFA等价性问题

DFA接受性问题

DFA接受性问题

检测一个给定的确定型有穷自动机是否接受一个给定的串

语言

A_{DFA}={<B,w>| DFA B接受串w}

DFA B接受串w $\Leftrightarrow$ $\langle B,w\rangle \in A_{DFA}$.

定理5.1

定理5.1: A_{DFA}是可判定语言

证明思路:

设计一个判定A_{DFA}的TM M.

M模拟B在w上的计算

想象一下写上述程序所需要的细节

定理5.1证明

证明：设计一个判定A_{DFA}的TM M.

M="对输入<B,w>,

其中B是DFA,w是串:

1) 在输入w上模拟B.

2) 若模拟以接受状态结束,

则接受;

若模拟以非接受状态结束,

则拒绝."

定理5.1证明

证明:(续) TM M首先检查输入<B,w>,

若w不是字符串，或B不是

(Q,Σ,δ,q_0,F)形式, 则拒绝.

然后M执行模拟.

M通过在带上写下信息, 来跟踪B

在w上运行时当前状态和当前位置.

状态和位置的更新

由B的转移函数确定.

当M处理完w最后一个符号时, 如果B

处于接受状态, 则M接受, 否则拒绝. #

NFA接受性问题

NFA接受性问题

检测一个给定的非确定型有穷自动机是否接受一个给定的串

语言

A_{NFA}={<B,w>|NFA B接受串w}

定理5.2

定理5.2: A_{NFA}是可判定语言

证明思路:

证法一:

用NTM N模拟NFA B在w上计算

证法二:

先把NFA B转化为等价的DFA C,

再用TM M模拟C在w上计算

定理5.2证明

证明：构造判定A_{NFA}的TM N.

N="对于输入<B,w>,

B是NFA,w是串:

1) 把NFA B转化成等价DFA C (书本定理2.19).

2) 在输入<C,w>上运行TM M (书本定理5.1).

3) 如果M接受,则接受, 否则拒绝." #

正则表达式派生性问题

- 正则表达式派生性问题
  - 一个正则表达式是否派生一个给定的串
- 语言
  - A_{REX}={<R,w>|正则表达式R派生串w}

定理5.3证明

证明：下面的TM P判定A_{REX}.

P="对于输入<R,w>,

R是RE, w是串:

1) 把REX R转化成等价DFA A (定理2.28).

2) 在输入<A,w>上运行TM M (定理5.1).

3) 如果M接受,则接受,

否则拒绝." #

说明

对于可判定性问题,

把**DFA, NFA, REX**

提供给**TM**都是等价的,

因为**TM**能在这三种编码

之间进行互相转换.

DFA空性问题

DFA空性问题

一个DFA是否根本不接受任何串?

语言

E_{DFA}={<A>|A是DFA且 L(A)=∅}

定理5.4

定理5.4: E_{DFA} 是可判定语言

证明思路:

- 逐个检查所有的串
  - 有无穷多个串
- DFA接受一个串当且仅当:
  - 从初始状态出发,
    - 沿着此DFA的箭头方向,
    - 能够到达一个接受状态
  - 采用例4.14的标记算法

定理5.4证明

证明: TM T="对于输入<A>, A是DFA:

1) 标记初始状态.
2) 重复下列步骤, 直到所有状态都被标记.
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, 则将它标记.
4) 如果没有接受状态被标记, 则接受, 否则拒绝." #

DFA等价性问题

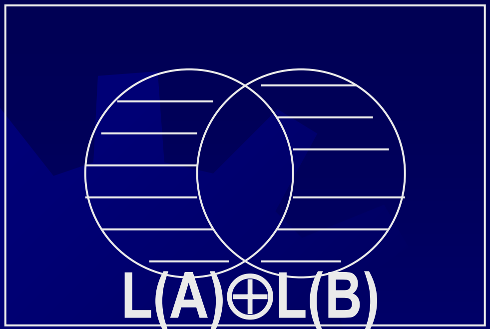
DFA等价性问题

检查两个DFA是否识别同一个语言

语言

EQ_{DFA}={<A,B>|A和B是DFA且L(A)=L(B)}

定理5.5



定理5.5: EQ_{DFA} 是可判定语言

证明思路:

正则语言对于布尔运算封闭,

因此对于对称差运算封闭

两个语言相等当且仅当

其对称差为空语言

正则语言是否为空,

这是可判定的(定理5.4)

定理5.5证明

- 证明:

TM F="对于输入<A,B>,

A和B都是DFA:

1) 构造DFA C使得

$L(C)=L(A)\oplus L(B)$

(定理2.12等);

2) 在输入<C>上运行TM T

(定理5.4)

3) 如果T接受, 则接受,

否则拒绝." #

关于上下文无关语言的可计算问题

关于CFL的可判定问题

- CFG派生性问题
- CFG空性问题
- CFG等价性问题
  - 不可判定
  - DCFG等价性问题可判定
- 上下文无关语言

CFG派生性问题

CFG派生性问题

- **检查一个CFG是否派生**
  - **一个特定的串**

语言

A_{CFG}={<G,w>|CFG G派生串w}

定理5.6

定理5.6: A_{CFG}是可判定语言

证明思路:

- 让G遍历所有派生,
  - 以确定哪一个是w的派生
    - 如果G产生w, 这个过程将终止
    - 如果G不产生w, 这个过程将不终止
    - 这样只能证明A_{CFG}是图灵可识别语言
- 采用CNF, 派生长度为n的串恰好
  - 需要2n-1步
    - 先把G转换成等价的CNF
    - 再检查所有长度为2n-1的派生

定理5.6证明

证明：TM S=“对于输入<G,w>,

G是CFG:

1) 把G转化成等价的CNF
(定理3.6).

2) 列出max{1,2|w|-1}步的
所有派生

3) 如果这些派生中有一个
产生w, 则接受, 否则拒绝.” #

CFG空性问题

CFG空性问题

检查一个CFG是否不派生任何串

语言

E_{CFG}={<G>|G是CFG且 L(G)=∅}

定理5.7

- 定理5.7: E_{CFG}是可判定语言
- 证明思路:

让TM S逐个检查所有可能的w

w有无限多个,这个过程将不终止

检查初始变元是否产生一个终结符串

检查每个变元是否产生一个终结符串

类似于DFA的标记算法

定理5.7证明

证明：TM R="对于输入<G>,

G是CFG:

1) 将G中所有终结符做上标记;

2) 重复下列步骤,

直到找不到可以做标记的变元

3) 如果G有规则$A\rightarrow U_1U_2\ldots U_k$,

且$U_1,U_2,\ldots, U_k$都已做过标记,

则将变元A做上标记.

4) 如果初始变元没有被标记,

则接受，否则拒绝."　　#

CFG等价性问题

- CFG等价性问题
  - 两个CFG是否派生同一个语言
- 语言
  - EQ_{CFG}={<G,H>|G和H都是CFG 且L(G)=L(H)}
- EQ_{CFG}不可判定! (下一章)

定理5.8

定理5.8: 每个CFL是可判定的.

证明思路:

设A是CFL,

把A的PDA转化成NTM

PDA某些计算分支可能不停机,

NTM的计算分支也可能不停机,

所以不是判定器

使用判定A_{CFG}的TM S(定理5.6).

定理5.8证明

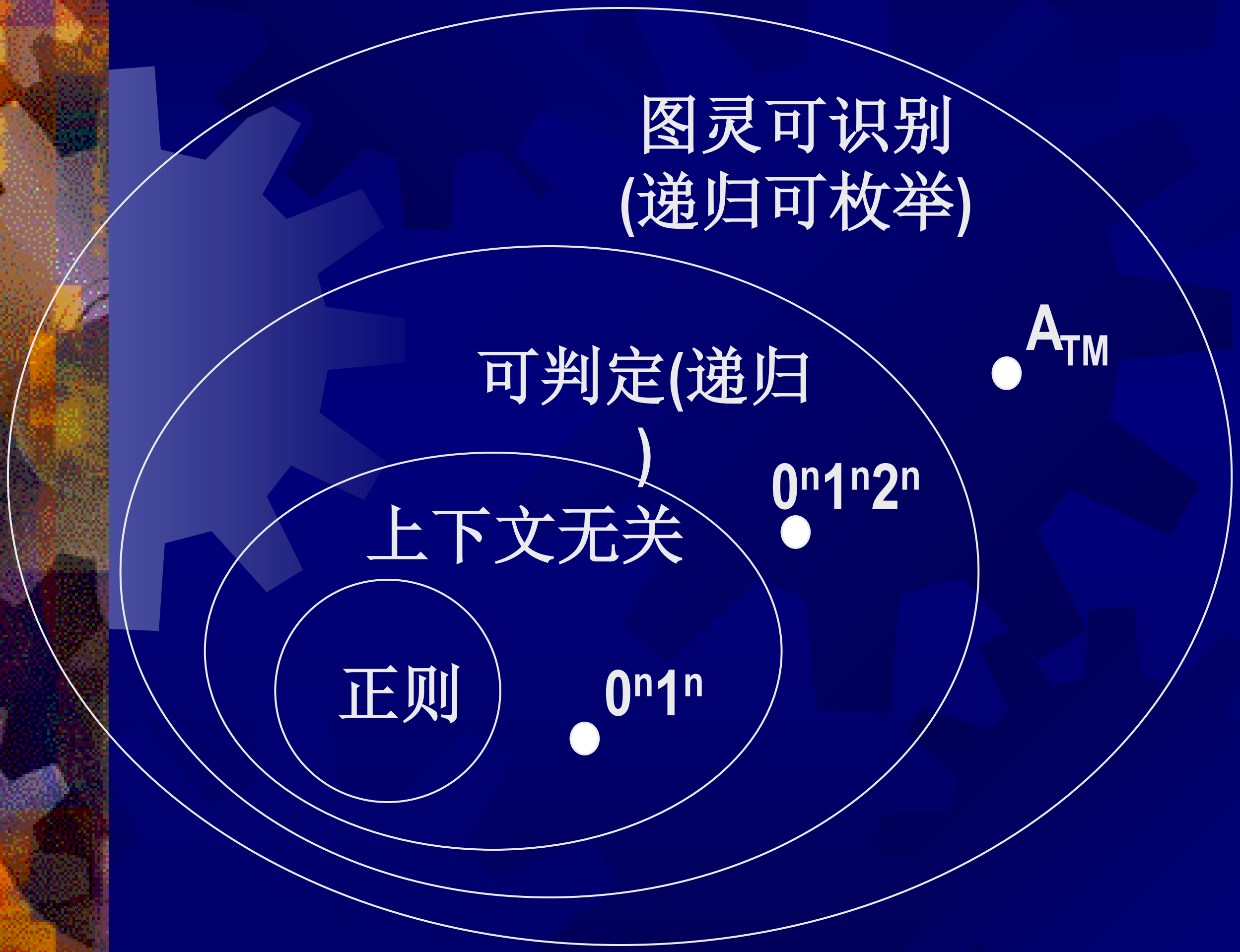
- 定理5.8: 每个CFL是可判定的.
- 证明: 设A是CFL, G是A的CFG,

下面设计判定A的TM M_G.

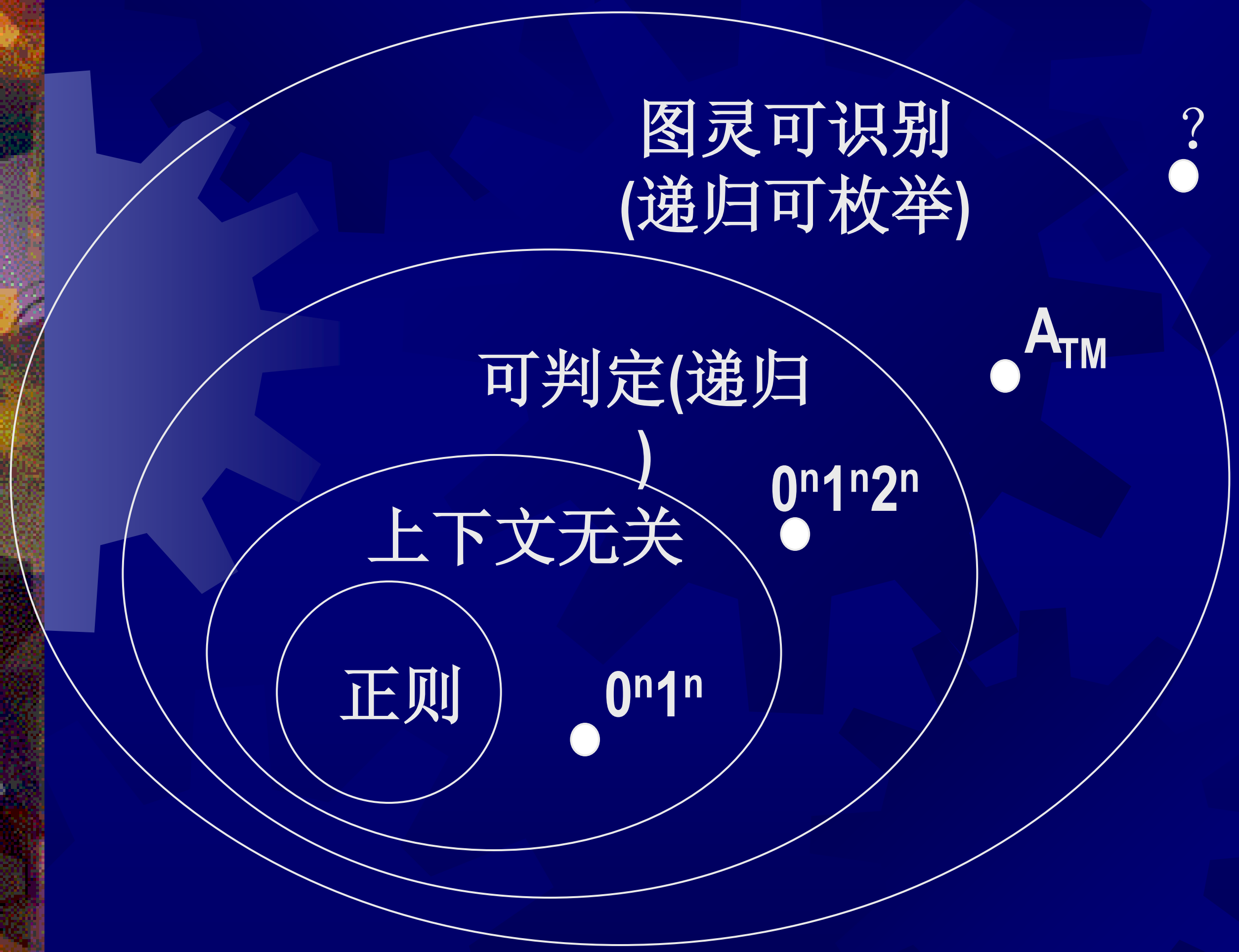
M_G = “对于输入w:

1) 在输入<G,w>上运行TM S
   (定理5.6).
2) 如果这个机器接受,则接受,
   否则拒绝.” #

语言类之间的关系



语言类之间的关系



对角化方法

图灵机接受性问题

- **TM接受性问题**
  - 检查一个图灵机是否接受一个给定的串
- 语言

A_{TM}={<M,w>|TM M接受串w}

定理5.9

定理5.9: A_{TM}是不可判定的

证明思路:

A_{TM}={ <M,w> | TM M接受串w }

D_{TM}={ <M> | TM M接受串<M> }

D_{TM}是A_{TM}的特例

用对角化法证明D_{TM}不可判定

定理5.9证明

证明: (反证) 假设TM H判定A_{TM}.

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

利用H, 构造TM D.

D="对于输入<M>, M是TM:

1) 在输入<M,<M>>上运行H.

2) 如果H接受, 就拒绝;

若H拒绝,就接受." 则

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

定理5.9证明

证明: (续) D在输入<D>上结果怎样?

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

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

D(<D>)=接受 ⇔ D(<D>)=拒绝

U(<M,w>)结果(可能不停机)

	$\langle M_1\rangle$	$\langle M_2\rangle$	$\langle M_3\rangle$	$\langle M_4\rangle$	$\langle M_5\rangle$	$\langle M_6\rangle$	$\cdots$
M_1	接受		接受		接受	接受	$\cdots$
M_2		接受			接受		$\cdots$
M_3							$\cdots$
M_4	接受		接受		接受		$\cdots$
M_5	接受	接受	接受	接受	接受	接受	$\cdots$
M_6		接受				接受	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

H(<M,w>)结果(处处停机)

	$<M_1>$	$<M_2>$	$<M_3>$	$<M_4>$	$<M_5>$	$<M_6>$	…
M_1	接受	拒绝	接受	拒绝	接受	接受	…
M_2	拒绝	接受	拒绝	拒绝	接受	拒绝	…
M_3	拒绝	拒绝	拒绝	拒绝	拒绝	拒绝	…
M_4	接受	拒绝	接受	拒绝	接受	拒绝	…
M_5	接受	接受	接受	接受	接受	接受	…
M_6	拒绝	接受	拒绝	拒绝	拒绝	接受	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

H(<M,<M>>)结果(处处停机)

	$\langle M_1\rangle$	$\langle M_2\rangle$	$\langle M_3\rangle$	$\langle M_4\rangle$	$\langle M_5\rangle$	$\langle M_6\rangle$	…
M_1	接受	拒绝	接受	拒绝	接受	接受	…
M_2	拒绝	接受	拒绝	拒绝	接受	拒绝	…
M_3	拒绝	拒绝	拒绝	拒绝	拒绝	拒绝	…
M_4	接受	拒绝	接受	拒绝	接受	拒绝	…
M_5	接受	接受	接受	接受	接受	接受	…
M_6	拒绝	接受	拒绝	拒绝	拒绝	接受	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

$D(\langle M\rangle)=\neg H(M,\langle M\rangle)$(处处停机)

	$\langle M_1\rangle$	$\langle M_2\rangle$	$\langle M_3\rangle$	$\langle M_4\rangle$	$\langle M_5\rangle$	$\langle M_6\rangle$	…
M_1	拒绝	拒绝	接受	拒绝	接受	接受	…
M_2	拒绝	拒绝	拒绝	拒绝	接受	拒绝	…
M_3	拒绝	拒绝	接受	拒绝	拒绝	拒绝	…
M_4	接受	拒绝	接受	接受	接受	拒绝	…
M_5	接受	接受	接受	接受	拒绝	接受	…
M_6	拒绝	接受	拒绝	拒绝	拒绝	拒绝	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

D(<D>)结果(矛盾)

	$<M_1>$	$<M_2>$	$<M_3>$	$<M_4>$	…	<D>	…
M_1	接受	拒绝	接受	拒绝	…	接受	…
M_2	拒绝	接受	拒绝	拒绝	…	拒绝	…
M_3	拒绝	拒绝	拒绝	拒绝	…	拒绝	…
M_4	接受	拒绝	接受	拒绝	…	拒绝	…
⋮	⋮	⋮	⋮	⋮	⋱	⋮	…
D	拒绝	拒绝	接受	接受	…	?	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

一个非图灵可识别语言

定理5.16

- A的补：$A^c=\Sigma^*-A$
  (课本记作 $\overline{A}$)
- 定理5.16:
- A可判定 ⇔
- A和A^c图灵可识别

定理5.16证明

证明: (⇒) 设A是可判定的.

可判定语言对布尔运算封闭

所以A^c是可判定的.

可判定语言都是图灵可识别的 (定义),

所以, A和A^c都是图灵可识别的.

定理5.16证明

证明: (⇐) 设A和A^c都是图灵可识别的.

设TM M_1识别A, TM M_2识别A^c.

下面构造TM M判定A.

M="对输入w:

1) 在输入w上并行运行M_1和M_2.

M有两个带,

一个模拟M_1,一个模拟M_2,

M交替地模拟两个机器的一步,

直到其中一个停机

2) 若M_1接受, 就接受;

若M_2接受, 就拒绝. "

定理5.16证明

证明: (续) 下面证明M确实判定A.

1) 任何一个串w要么在A中,

要么在A^c中.

所以M_1和M_2必有一个接受w.

因为当M_1或M_2接受时,

M就停机, 所以M总会停机.

2) M接受所有A中的串,

拒绝所有不在A中的串,

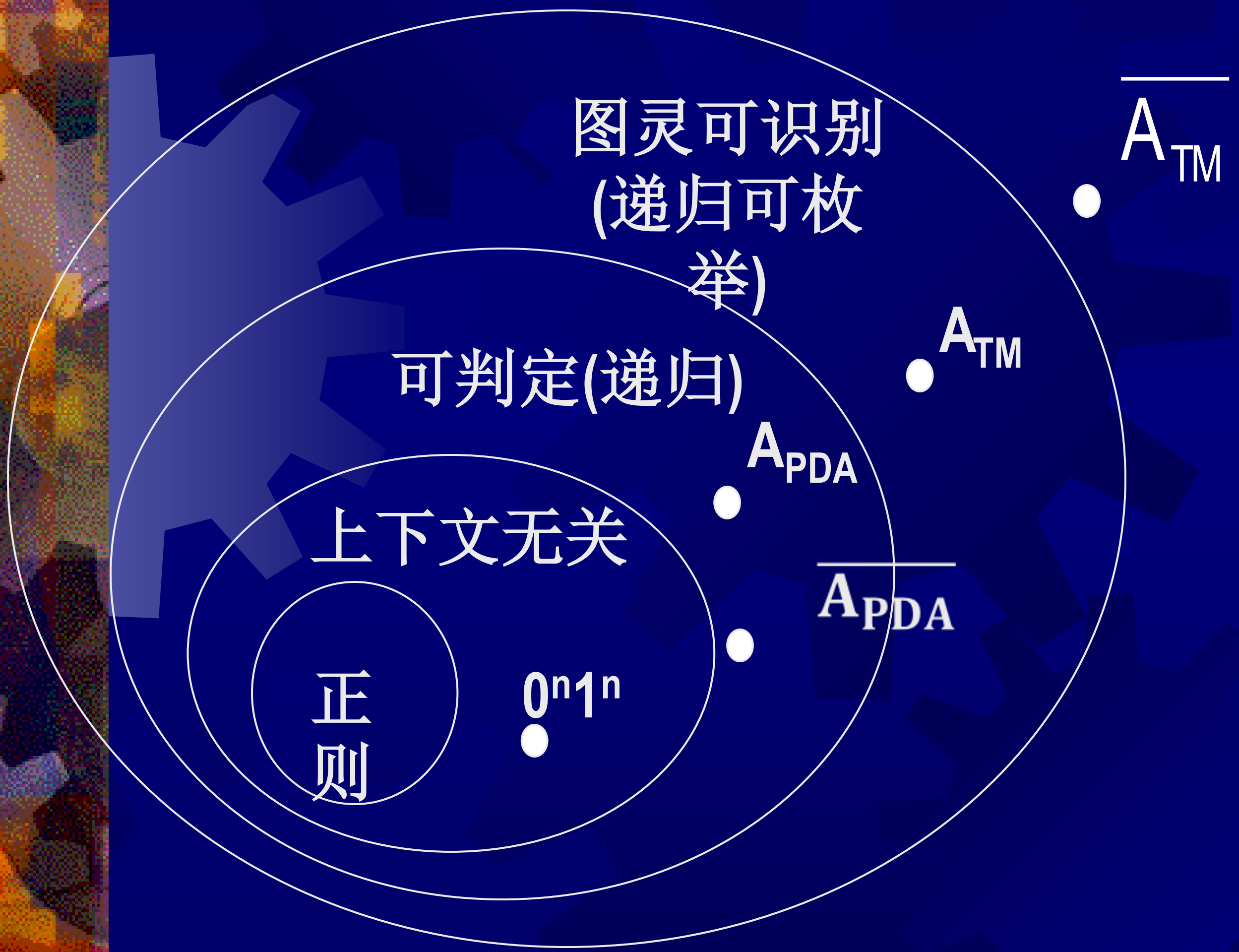
所以，M判定A. #

推论5.17

- **推论5.17**: $\overline{A_{TM}}$ 不是
- 图灵可识别的.
- **证明**: (反证)

假设 $\overline{A_{TM}}$ 是图灵可识别的,
因为A_{TM}是图灵可识别的(引理),
所以A_{TM}是可判定的(定理5.16).
但A_{TM}是不可判定的(定理5.9),
矛盾! #

语言类之间的关系



图灵可识别
(递归可枚举)
可判定(递归)
上下文无关
正则
0^n1^n
A_{PDA}
$\overline{A_{PDA}}$
A_{TM}
$\overline{A_{TM}}$

(不)可判定性总结

- 可判定性结论

	DFA	CFG	TM
接受性	√	√	×
空性	√	√	×
等价性	√	×	×

- 用对角化法证明不可判定语言
- 非图灵可识别语言

作业

- 4.1 4.2 4.3 4.4 4.5