计算理论

第4章 可判定性

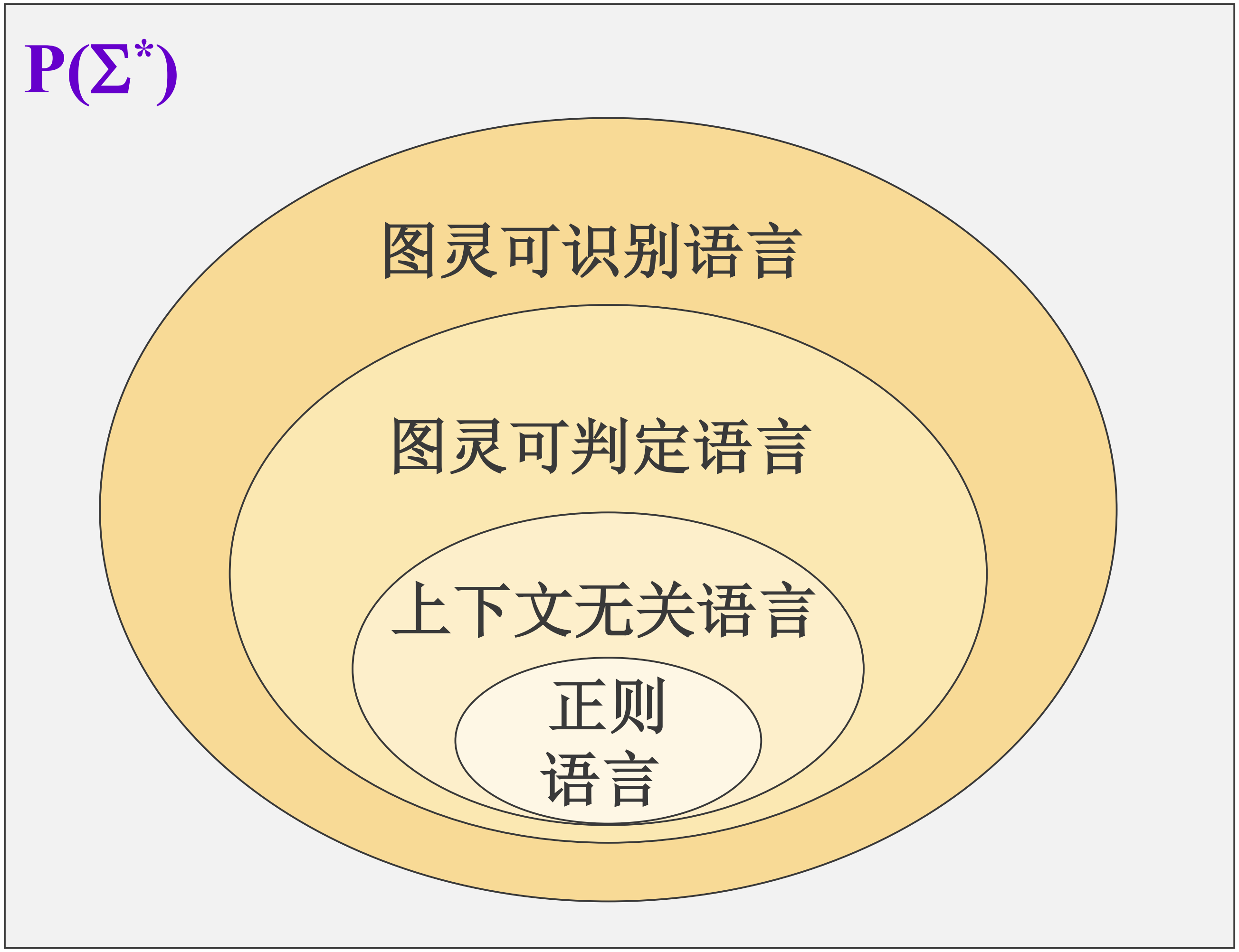
教材:

[S] 唐常杰等译, Sipser著, 计算理论导引, 机械工业.

参考资料:

[L] Lewis等著, 计算理论基础, 清华大学.

各种语言类的包含关系



A={0w1 : w∈Σ*}

正则语言

B={0^n1^n : n≥0}

上下文无关语言

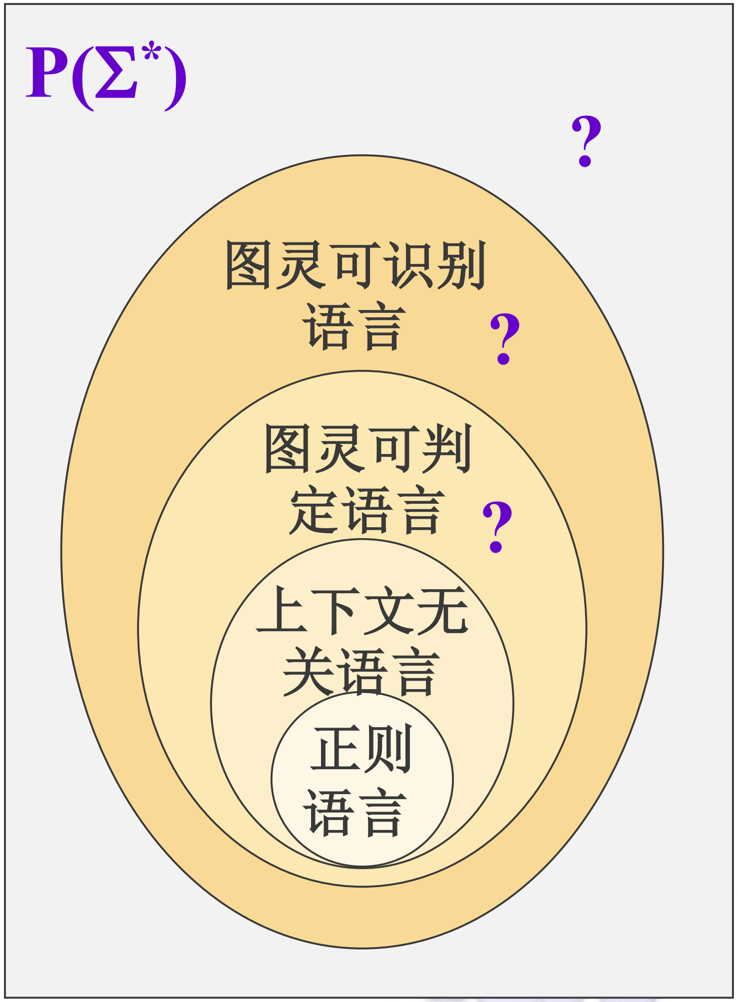
C={0^k:k=2^n,n≥0}

图灵可判定语言

第4章 可判定性

- 1. 算法及其可判定性
- 2. 关于正则语言的可判定性
- 3. 关于图灵机的可判定性

——第4章 可判定性
第5章 可归约性



算法及其可判定性

- 丘奇-图灵论题
  - 算法 ≅ 图灵判定器
- 问题可解
  - 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，理解什么是计算

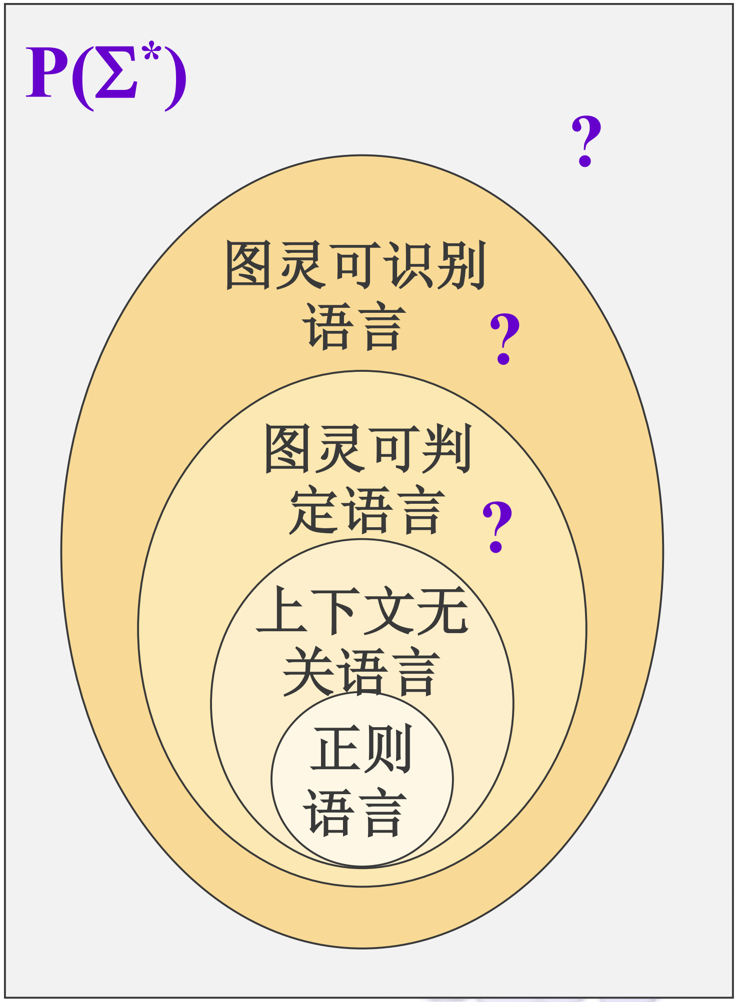
问题及语言

- **语言：** 给定字母表Σ，称Σ上一些字符串的集合为Σ上的**语言**.
- $\Sigma^*=\{x|\ x$是Σ上全体有限长度的字符串$\}$
- Σ上的任意语言A都是Σ^*的子集，及$A\subseteq\Sigma^*$.
- 判定性问题与{0,1}上的语言一一对应
  - $P(\Sigma^*)=P(\{0,1\}^*)$: 全体判定性问题

- 可判定性
  - 存在处处停机的TM程序

第4章 可判定性

- 1. 算法及其可判定性
- 2. 关于正则语言的可判定性
  - 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
- 3. 关于图灵机的可判定性



DFA接受性问题

- DFA接受性问题（Acceptance Problem）
  - 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
- 问题形式定义
  - A_{DFA}={<B, w>|DFA B接受串w}
  - DFA B接受串w$\Leftrightarrow$ <B,w>$\in A_{DFA}$.

定理:

A_{DFA}={<B,w>|DFA B接受串w}是一个**可判定语言**

证明：A_{DFA}是一个可判定语言

- 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. ”

- $L(M)=A_{DFA}$.
- 下面说明实现细节

证明：A_{DFA}是一个可判定语言

- 证明:(将B视为子程序或实现细节)
- 1. 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.
- 2. 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（Acceptance Problem）
  - 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
- 形式定义
  - A_{NFA}={<B, w>|NFA B接受串w}
  - NFA B接受串w⇔ <B,w>∈A_{NFA}.

定理:
A_{NFA}={<B,w>|NFA B接受串w}是一个可判定语言

证明：A_{NFA}是一个可判定语言

- 思路1: 直接模拟NFA
- 思路2: 先将NFA转换成DFA.
- 证明:如下构造A_{NFA}的判定器N:
- N="在输入<B,w>上, 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A_{DFA}的判定器M.

  3)如果M接受,则接受,否则拒绝."
- 运行图灵机M: M作为子程序加进N的设计中.
- $L(N) = A_{NFA}$.

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

- 形式定义：
- A_{REX}={<R, w>|正则表达式R派生串w}

定理:

A_{REX}={<R,w>|正则表达式R派生串w}是一个**可判定语言**

证明：A_{REX}是一个可判定语言

- 证明：构造图灵机 P判定A_{REX}.
- P= “对于输入<R, w>,
  - R是正则表达式, w是串：
- 1) 把正则表达式 R转化成等价DFA A
- 2) 在输入<A, w>上运行图灵机 M
- 3) 如果M接受,则接受,

  否则拒绝.” #

- 说明：对于可判定性问题, DFA, NFA, REX提供给图灵机都是等价的, 因为图灵机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?
-
- 定义语言
  - E_{DFA}={<A>|A是DFA, L(A)=∅}可判定

定理:
E_{DFA}={<A>|A是DFA, L(A)=∅}是一个可判定语言

证明：E_{DFA}是一个可判定语言

- E_{DFA}={<A>|A是DFA, L(A)=∅}是一个可判定语言
- 证明思路:
- 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。
- 若A为一个DFA, 则

  L(A)≠∅ ⇔ 存在从起始状态到某接受状态的路径.
- 证明思路：检查从初始状态是否有路径到达接受状态
  - 采用图连通性测试中的标记算法

证明：E_{DFA}是一个可判定语言

- 证明: 构造图灵机T

T="对于输入<A>, 其中A是一个DFA:

1)标记起始状态.

2)重复下列步骤, 直到没有新标记出现.

3) 对任一未标记状态, 若有从已标记状态到它的转移, 则将它标记.

4)如果无接受状态被标记,则接受;

否则拒绝." #

- $L(T) = E_{DFA}$.

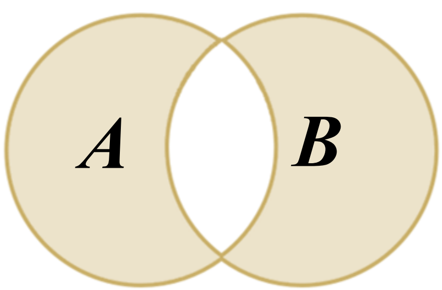
DFA等价性问题

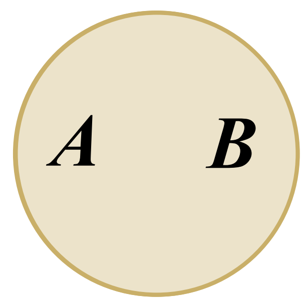
- **DFA等价性问题：**
  - 检查两个DFA是否识别同一个语言
- **形式定义：**
  - EQ_{DFA}={<A, B>|A和B是DFA且L(A)=L(B)}

定理:
EQ_{DFA}={<A, B>|A和B是DFA且L(A)=L(B)}
是一个可判定语言

证明：$\mathrm{EQ}_{\mathrm{DFA}}$是一个可判定语言

- 证明思路：
  - 两个语言相等当且仅当其对称差为空语言
  - 正则语言对于对称差运算封闭
  - 正则语言是否为空是可判定的





$A \oplus B$

$A = B \Leftrightarrow A \oplus B = \varnothing$

证明：EQ_{DFA}是一个可判定语言

- 证明：构造TM F判定EQ_{DFA}
- 图灵机 F="对于输入<A, B>, A和B都是DFA:

  1) 构造DFA C使得

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

  2) 在输入<C>上运行图灵机 T

  3) 如果T接受, 则接受,

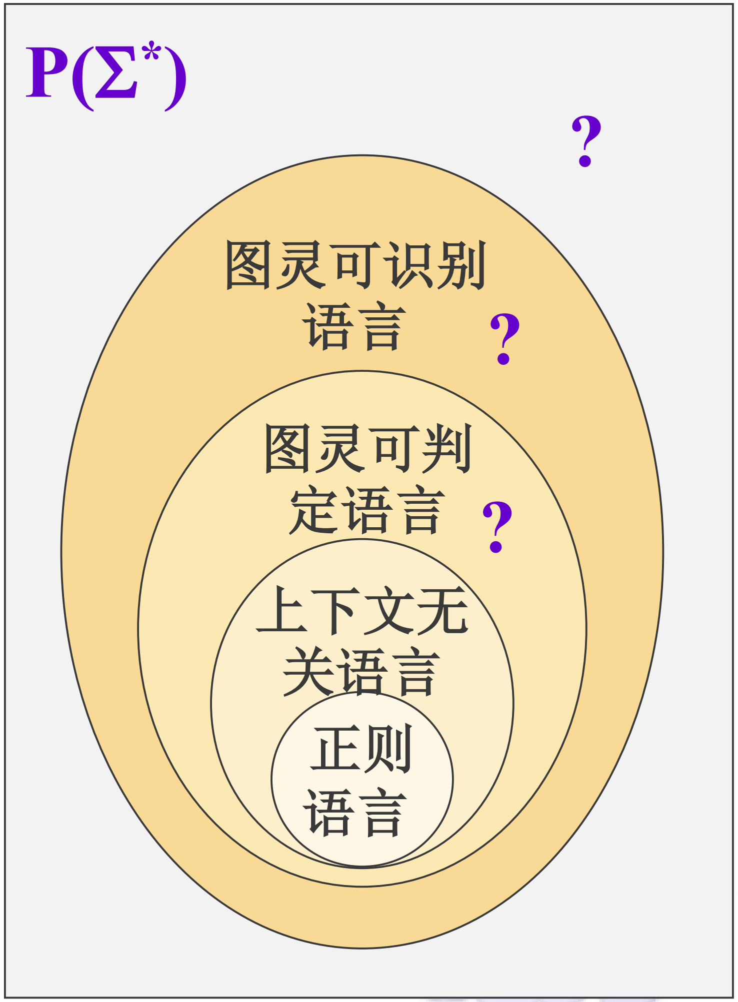
     否则拒绝."#

关于正则语言的问题总结

- **成员测试:**
  - ¶ A_{DFA} = {<B, w>|B是DFA,w是串,B接受w}可判定
  - ¶ A_{NFA} = {<B, w>|B是DFA,w是串,B接受w}可判定
- **正则表达式派生性测试:**
  - ¶ A_{REX}={<R, w>|正则表达式R派生串w}可判定
- **空性质测试:**
  - ¶ E_{DFA} = {<A>|A是DFA, L(A)=Φ}可判定
- **等价性质测试:**
  - ¶ EQ_{DFA} = {<A,B>|A和B都是DFA,且L(A)=L(B)}可判定

第4章 可判定性

- 1. 算法及其可判定性
- 2. 关于正则语言的可判定性
- 3. 关于图灵机的可判定性
  - A_{TM}可识别 不可判定
  - $HALT_{TM}$ 可识别 不可判定
  - E_{TM}图灵机空性问题不可判定
  - $HALT_{TM}$ 的补 非图灵可识别
  - LBA接受性问题
- 不可判定问题举例



图灵机接受性问题A_{TM}

- 图灵机接受性问题
  - 检查一个图灵机是否接受一个给定的串
- 语言
- A_{TM}={<M, w>|M是一个图灵机，且M接受串w}

定理:
A_{TM}是可识别的。
A_{TM}是不可判定的。

证明：A_{TM}是可识别的

- 证明：构造TM U识别A_{TM}。
- U = “对于输入<M, w>,
  M是一个图灵机，w是一个串
- 1. 在输入上模拟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。” #

- 说明：
  - 图灵机U是一个通用图灵机
  - U可以模拟任何其它图灵机

证明：A_{TM}是不可判定的

A_{TM}={<M, w>|M是一个图灵机，且M接受串w}

- 证明思路： 反证法
  - 假设A_{TM}可判定, 找矛盾：构造一个无法判定的特例。

- 假设A_{TM}可判定，且设H是其判定器。
- 构造TM D给出与H相反的结论。即：

D="对于输入<w>,其中w是一个串：
1)在串<M, w>上运行H.
2)若H接受(<M, w>), 则D拒绝(w);
若H拒绝(<M, w>), 则D接受(w)."

	$\langle M,w_1\rangle$	$\langle M,w_2\rangle$
H	ac	rej
D	rej	ac

Diagonal

证明：A_{TM}是不可判定的

用对角线法证明如下：

判定器H依次在下列图灵机及其编码串<**M, <M>**>上运行

All TM descriptions:

All TMs

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$		$\langle D \rangle$
M_1	acc						
M_2		rej					
M_3			acc				
M_4				acc			
$\vdots$							
D							

Diagonal

证明：A_{TM}是不可判定的

- A_{TM}={<M, w>|M是一个图灵机，且M接受串w}
- 证明：假设A_{TM}可判定，且设H是其判定器。

D="对于输入<M>,其中M是图灵机:

1)在串<M, <M>>上运行H.

2)若H接受(<M, <M>>), 则D拒绝(<M>);

若H拒绝(<M, <M>>), 则D接受(<M>)."

- 若在串<D, <D>>上运行H

  若H接受(<D, <D>>), 则D拒绝(<D>);

  若H拒绝(<D, <D>>), 则D接受(<D>)."
- 矛盾, 所以H不存在.

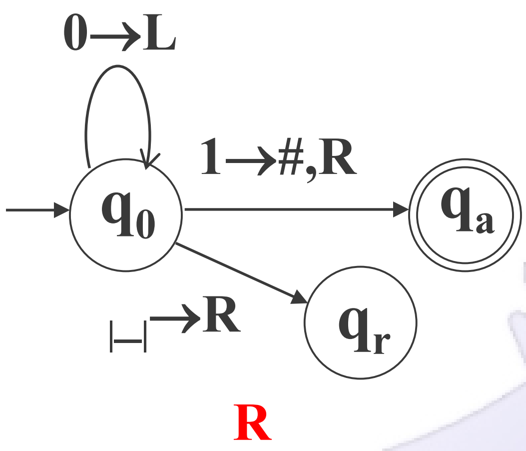
	<M,<M>>		<D,<D>>	
H	ac	rej	ac	rej
D	rej	ac	?	?

证明:停机问题$HALT_{TM}$是图灵可识别的

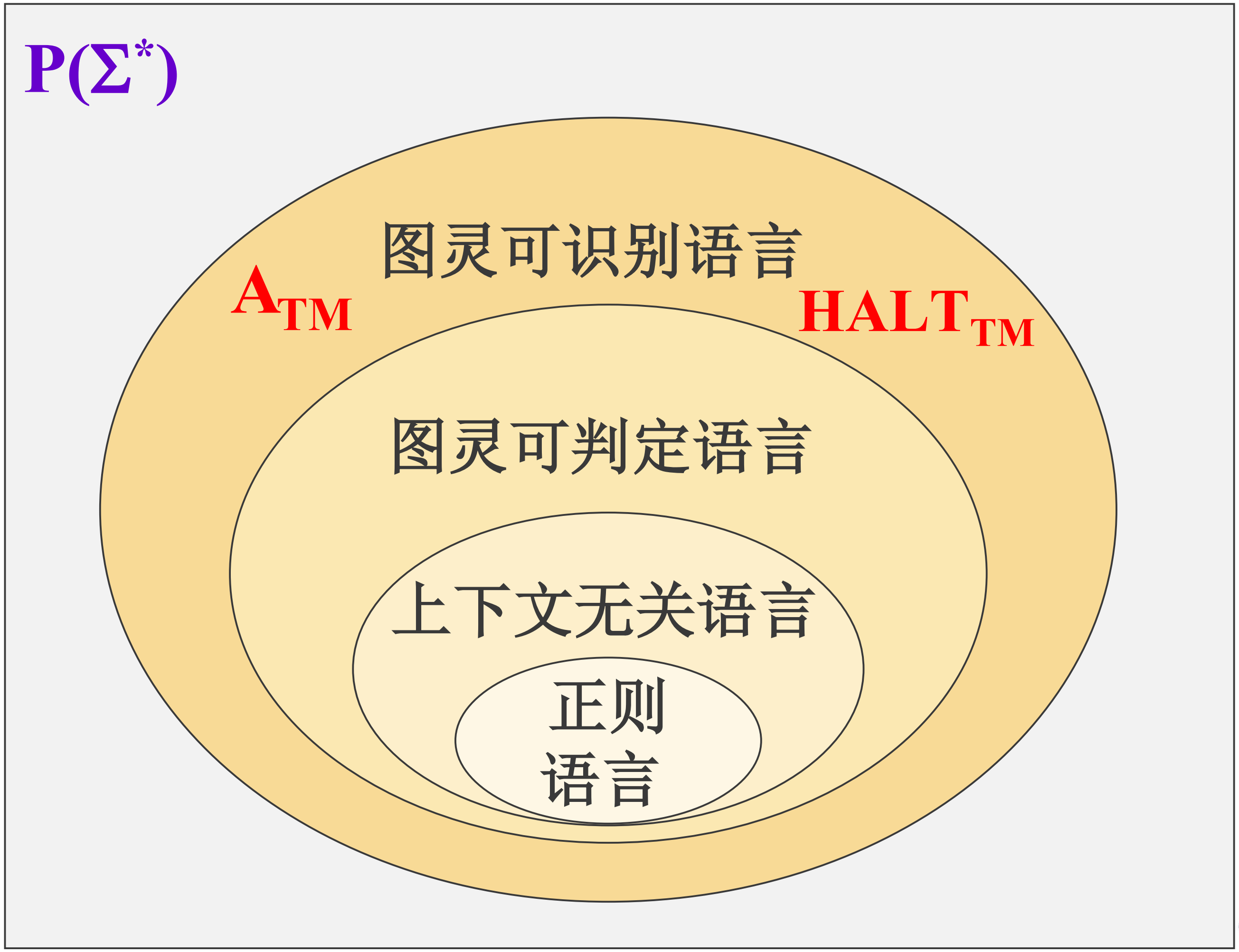
- $HALT_{TM}$={ <M, x> | 图灵机M在串x上会停机 }
- 证明: 构造识别$HALT_{TM}$的TM T,
- T = “ 对于输入<M, x>, M是图灵机, x是串

  1. 在x上模拟M,
  2. 若M停机(接受或拒绝), 则接受.” #
- T的语言是$HALT_{TM}$.
- 注：T是识别器，不是判定器

  例如：T上运行<R, 01>

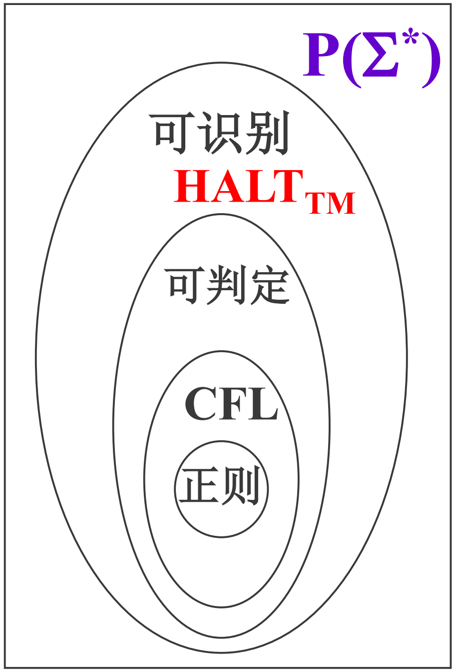


各种语言类的包含关系



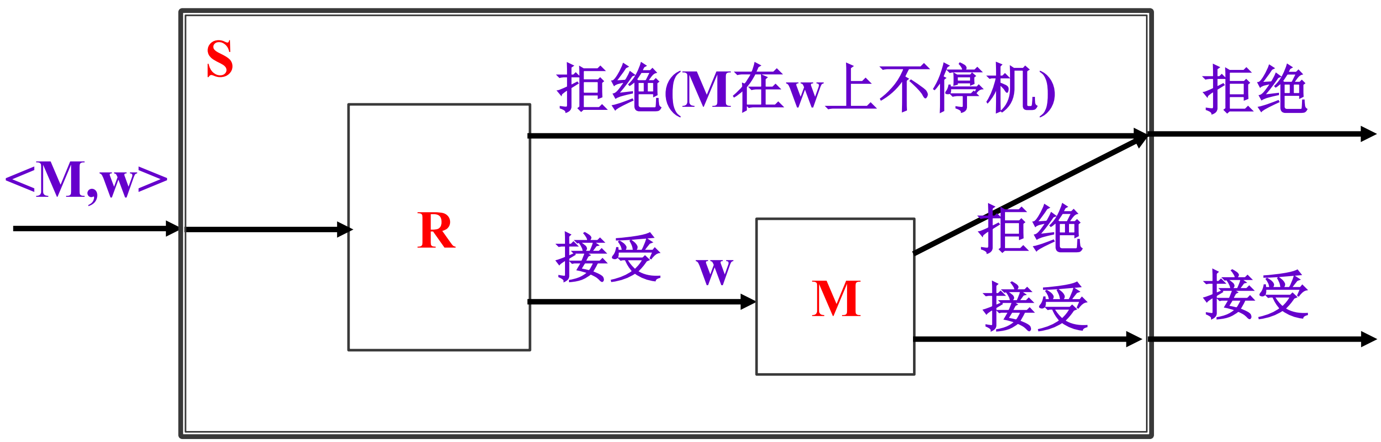
定理:停机问题$HALT_{TM}$不可判定

- $HALT_{TM}$={ <M, x> | 图灵机M在串x上会停机 }
- 证明1: 反证法, 对角线法
- 假设$HALT_{TM}$有判定器H, 构造D:
- D = “ 对于输入<M>, M是图灵机,
  1) 在<M,<M>>上运行H,
  2) 若H接受, 则返回1);
  3) 若H拒绝, 则停机.”
- 在D上输入串<D>”是否会停机?
  - 若D停机, H接受<D,<D>>, 则由2), D不停机
  - 若D不停机, H拒绝<D,<D>>, 则由3), D停机
- 矛盾, 所以H不存在.



证明:停机问题$HALT_{TM}$不可判定

- 证明思路2：反证法：与A_{TM}不可判定矛盾
- 假设$HALT_{TM}$可判定，TM R判定$HALT_{TM}$，
- 则可利用R构造TM S 判定A_{TM}，这与A_{TM}不可判定矛盾



证明:停机问题$HALT_{TM}$不可判定

- 证明2: (反证)假设图灵机R判定$HALT_{TM}$,
  则构造图灵机S判定A_{TM}, 这产生矛盾!
- S="在输入<M,w>上, 其中M是图灵机, w是串:
  1) 在输入<M,w>上运行图灵机 R.
  2) 如果R拒绝,则拒绝.
  3) 如果R接受,则在w上模拟M, 直到M停机.
  4) 若M接受,则接受;
     若M拒绝,则拒绝."
- 若R是$HALT_{TM}$的判定器, 则S是A_{TM}的判定器,
- 但是A_{TM}不可判定, 所以$HALT_{TM}$不可判定. #

证明方法：归约

- 归约性 reducibility
  - P可归约到Q, (如：映射归约 $P \leq_m Q$)
- 如果P可归约到Q，那么
  - 如果Q成立，则P也成立
  - 如果P不成立，则Q也不成立
- 例如：在“$HALT_{TM}$不可判定”的证明中
  - A_{TM}可归约到$HALT_{TM}$，则
  - 因为A_{TM}不可判定，所以$HALT_{TM}$也不可判定
  - 证明A_{TM}可归约到$HALT_{TM}$：若$HALT_{TM}$是可判定的，则A_{TM}是可判定的

证明方法：归约

- 例如：证明“E_{TM}不可判定”
  - 如果A_{TM}可归约到E_{TM}，则
  - 因为A_{TM}不可判定，所以E_{TM}也不可判定
  - 证明A_{TM}可归约到E_{TM}：若E_{TM}是可判定的，则A_{TM}是可判定的

图灵机的空性问题E_{TM}是不可判定的

- 图灵机的空性问题
  - 一个图灵机是否根本不接受任何串
- E_{TM} = {<M>|M是图灵机, L(M)=∅} 是不可判定的。

- 证明思路：(反证法-归约)
- 反证法-归约
  - 下面证明A_{TM}可归约到E_{TM}，即
  - 假设图灵机R判定E_{TM}，

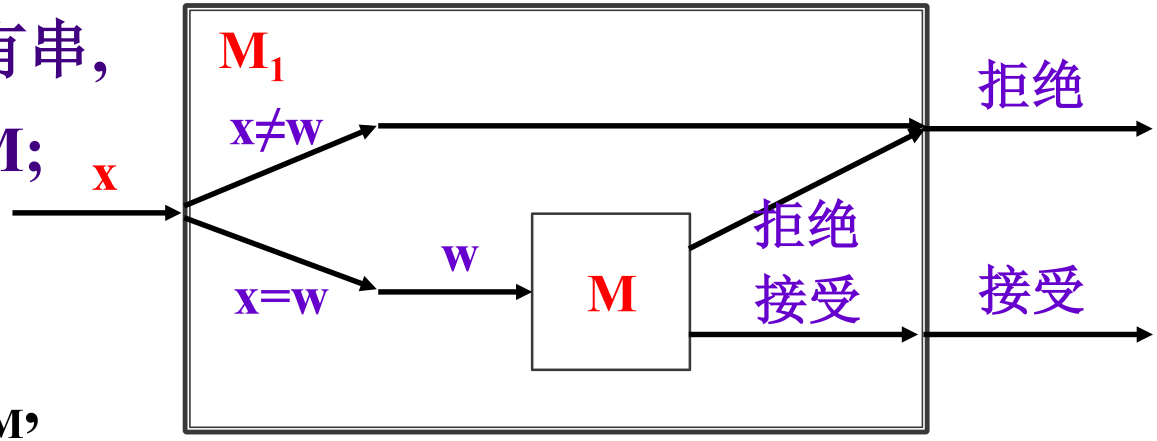
    则利用R构造图灵机S判定A_{TM}.
- 空性：如果图灵机M接受任意串w，则L(M)≠∅

证明：E_{TM}是不可判定的

- 先修改图灵机M为M_1.

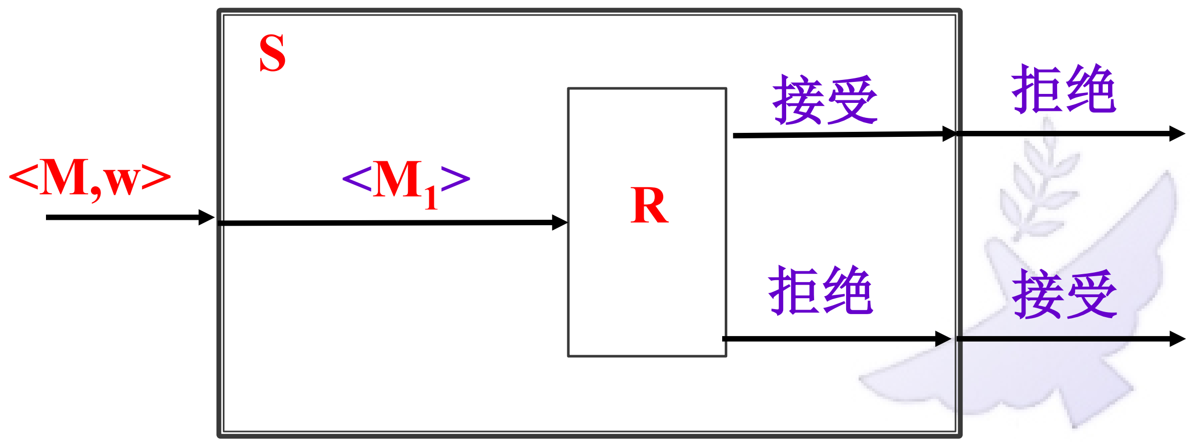
  M_1拒绝除w外的所有串，

  M_1在w上完全模拟M;



- 假设图灵机R判定E_{TM},

  则利用R构造图灵机S判定A_{TM}.



证明：E_{TM}是不可判定的

- 证明: (归约)假设图灵机R判定E_{TM}, 则构造图灵机 S判定A_{TM}.
- 修改型图灵机M_1. M_1= “ 在输入x上:

  1) 如果$x \neq w$, 则拒绝.

  2) 如果x=w, 则在x上运行M, M接受就接受， M拒绝就拒绝. ”
- S = “ 在输入<M, w>上:

  1) 用M和w的描述来构造上述图灵机 M_1.

  2) 在输入M_1上运行R.

  3) 如果R接受, 则拒绝;

     如果R拒绝, 则接受. ”
- 若R是E_{TM}的判定器, 则S是A_{TM}的判定器,
- 但是A_{TM}不可判定, 所以E_{TM}不可判定. #

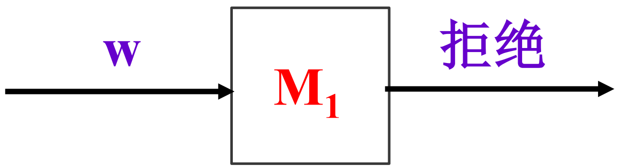
图灵机的等价性问题EQ_{TM}

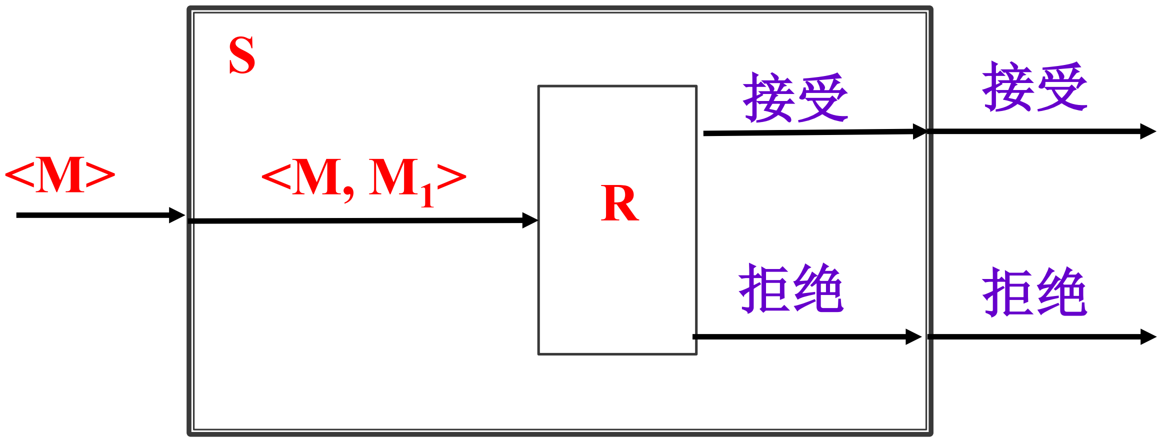
- 图灵机的等价性问题：
  - 检查两个给定的图灵机是否识别相同的语言
- EQ_{TM}= { <M_1,M_2> | M_1和M_2是图灵机，且$L(M_1)=L(M_2)$ }
- 定理：EQ_{TM}是不可判定的。

- 证明思路：使用从E_{TM}出发的归约，即E_{TM}归约到EQ_{TM}
- 若M_1, M_2中有一个识别空语言, 则
  - 若M_1和M_2等价当且仅当
  - 另一个机器也识别空语言
- E_{TM}是EQ_{TM}的特例 其中一个机器识别空语言

证明：EQ_{TM}是不可判定的

- **假设**图灵机**R判定EQ_{TM}**, 下面构造图灵机**S判定E_{TM}**.
- 令**$L(M_1)=\Phi$**.





证明：EQ_{TM}是不可判定的

- 证明: 假设图灵机R判定EQ_{TM},
- 下面构造图灵机S判定E_{TM}.
- S="对于输入<M>, M是图灵机:

  1) 在输入<M, M_1>上运行R,

  M_1是拒绝所有输入的图灵机.

  2) 如果R接受, 则接受;

  如果R拒绝, 则拒绝. "

  若R是EQ_{TM}的判定器, 则S是E_{TM}的判定器,

  但是E_{TM}不可判定, 所以EQ_{TM}不可判定. #

一个非图灵可识别语言

- HALT$_{TM}$的补（$HALT^{C}_{TM}$）不是图灵可识别的。
- 语言A的补：$A^c=\Sigma^*-A$

- 定理：A可判定⇔A和A^c都是图灵可识别
- 证明: (⇒) 设A是可判定的，则A和A^c都是图灵可识别.
- 设A是可判定的，

  因为可判定语言对补运算封闭，所以A^c是可判定的.

  又，可判定语言都是图灵可识别的

  所以, A和A^c都是图灵可识别的.

定理: $HALT_{TM}$的补不是图灵可识别的

- 证明: (⇐) 若A和A^c都是图灵可识别, 则A图灵可判定
- 设图灵机T和Q分别识别A和A^c, 构造图灵机R:

R = “ 对于输入x, x是串,

1. 在x上同步模拟T和Q, 直到有一个停机,
2. 若T接受x, 则接受x;
3. 若Q接受x, 则拒绝x.”

R是判定器, R的语言是A. #

$\forall x \in A \Rightarrow$ T一定停机接受 $\Rightarrow$ R停机接受

$\forall x \notin A \Rightarrow$ Q一定停机接受 $\Rightarrow$ R停机拒绝

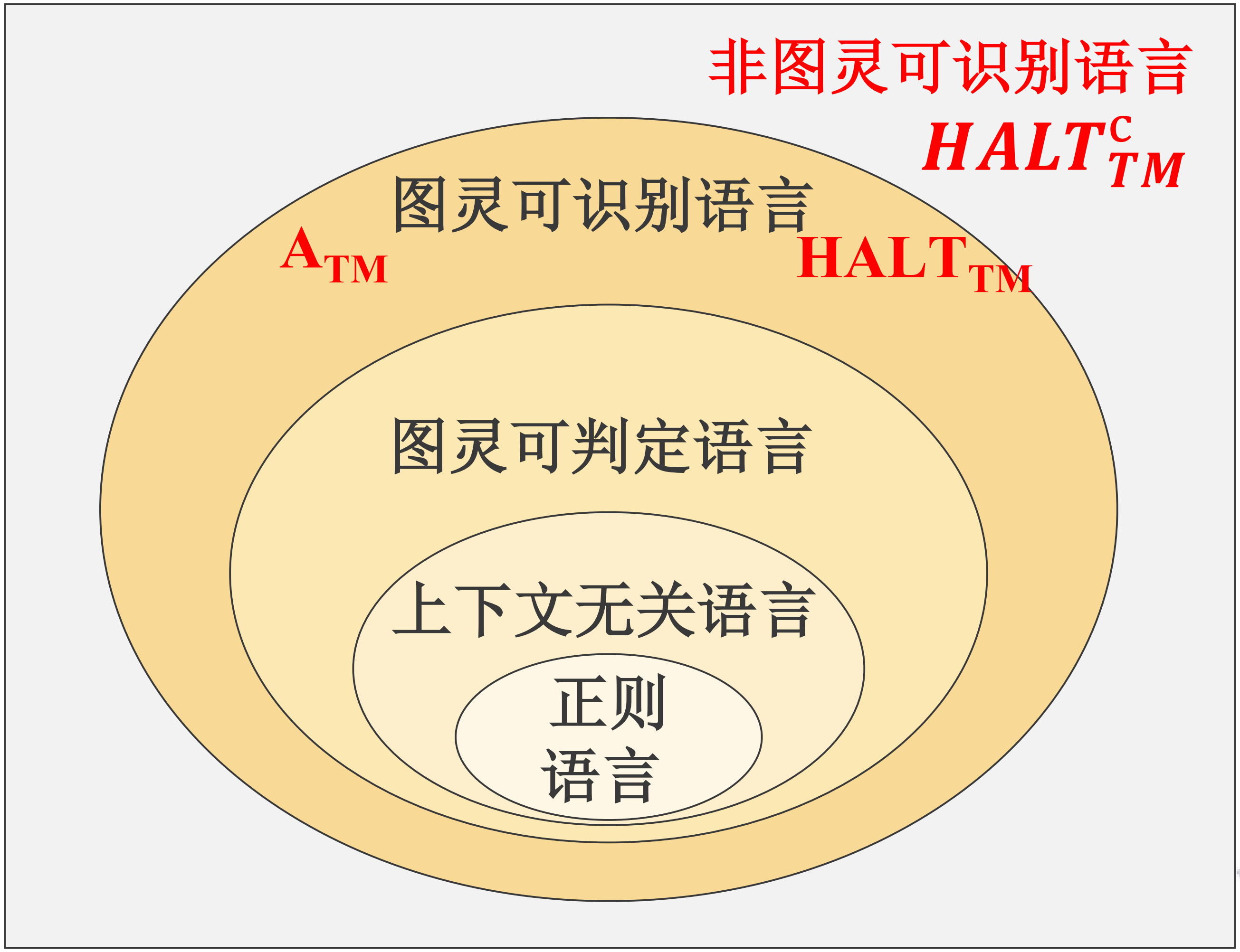
定理: $HALT_{TM}$的补不是图灵可识别的

- 定理：A可判定⇔A和A^c都是图灵可识别

- 推论: 停机问题$HALT_{TM}$的补不是图灵可识别的.
- 证明：归约，从A_{TM}开始的归约
  - 即，证明A_{TM}可归约到$HALT_{TM}$.
  - $HALT_{TM}$是可识别的，若$HALT^c_{TM}$可识别，则A_{TM}可判定。

各种语言类的包含关系

$P(\Sigma^*)$

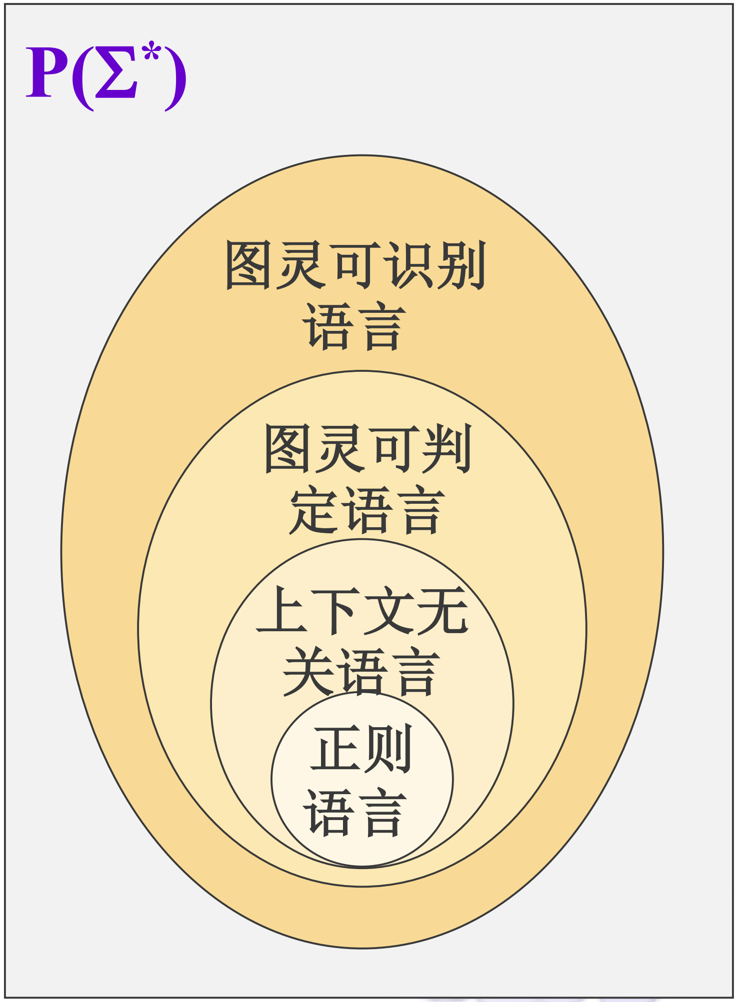


一些自然构造的问题

- 停机问题:
  - $HALT_{TM}$ = { <M,x> | 图灵机M在串x上会停机 } 不可判定
- 成员测试:
  - A_{DFA} = {<B, w>|B是DFA,w是串,B接受w}可判定
  - A_{CFG} = {<B, w>|B是CFG,w是串,B派生w}可判定
  - A_{TM} = {<M, w>|M是一个TM,且接受w}不可判定
- 空性质测试:
  - E_{DFA} = {<A>|A是DFA, L(A)=Φ}可判定
  - E_{CFG} = {<G>|G是CFG, L(G)=Φ}可判定
  - E_{TM} = {<M>|M是TM,L(M)=∅}不可判定
- 等价性质测试:
  - EQ_{DFA} = {<A,B>|A和B都是DFA,且L(A)=L(B)}可判定
  - EQ_{CFG} = {<A,B>|A和B都是CFG,且L(A)=L(B)}不可判定
  - EQ_{TM} = {<A,B>|A和B都是TM,且L(A)=L(B)}不可判定

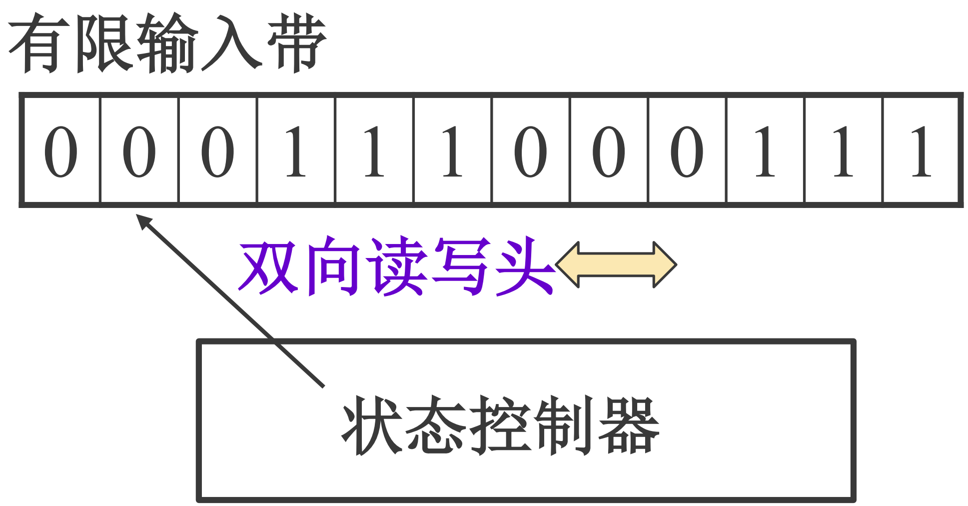
第4章 可判定性

- 1. 算法及其可判定性
- 2. 关于正则语言的可判定性
- 3. 关于图灵机的可判定性
  - A_{TM}图灵可识别 不可判定
  - $HALT_{TM}$ 图灵可识别 不可判定
  - E_{TM}图灵机空性问题不可判定
  - $HALT_{TM}$ 的补 非图灵可识别
  - LBA接受性问题
  - 不可判定问题举例



线性界限自动机(LBA)

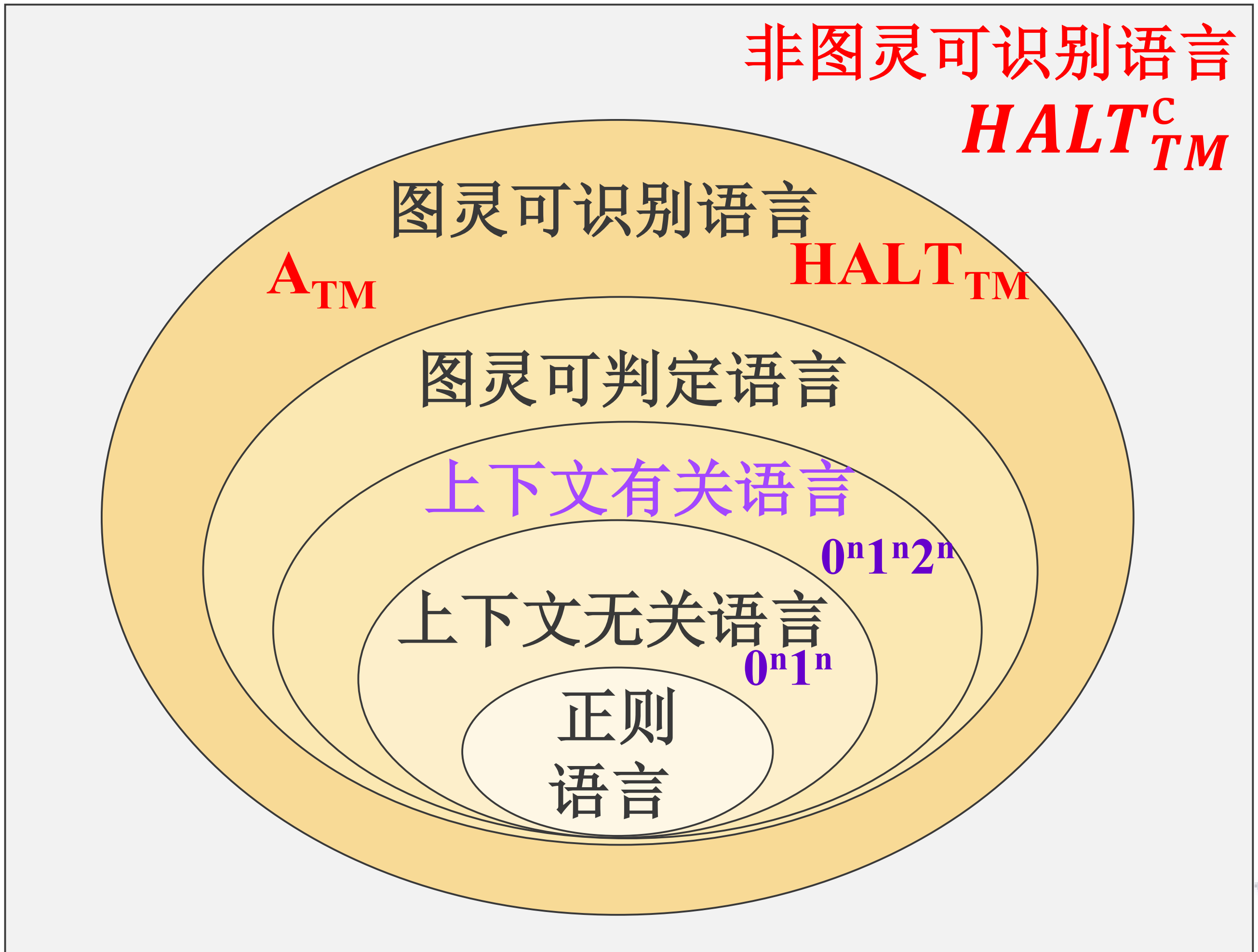
- 线性界限自动机: **L**inear **B**ounded **A**utomata
  - 带头不能移出输入区的图灵机.
  - 等价于“带头不能移出输入区的常数倍.”



LBA

各种语言类的包含关系

$P(\Sigma^*)$



LBA接受性问题

- LBA接受性问题
  - 检测一个给定的线性界限自动机LBA是否接受一个事先给定的串
- 语言
  - $A_{LBA}=\{<M, w>|$ LBA M接受串w$\}$

定理: A_{LBA}是可判定的.

证明：A_{LBA}是可判定的.

- 引理: 设M_1是一个确定型LBA,
  M_1有q个状态和g个带符号。
  对于长度为n的带子, M_1恰有qng^n个不同格局.
- 证明:
  图灵机格局包括当前状态, 带头位置和带内容.
  这三者的不同组合数为qng^n.
  所以M_1恰有qng^n个不同格局.
  #

证明：A_{LBA}是可判定的.

- 证明思路: 在输入w上模拟M
  - 如果M停机接受,则接受
  - 如果M停机拒绝,则拒绝
  - 如果M不停机, 则……?
    - 根据上述引理, M_1恰有qng^n个不同格局
    - 如果M运行超过qng^n步,
    - 则M重复了某个格局,
    - 因此M将陷入死循环。

证明：A_{LBA}是可判定的.

- 证明: 图灵机L判定A_{LBA}.

L = "对输入<M, w>,

M是LBA, w是串:

1) 在w上模拟M运行qng^n步，或者直到它停机.

2) 如果M停机, 则

当M接受时接受,

当M拒绝时拒绝;

如果M还没有停机, 则拒绝. "

#

不可判定问题举例

1）Hilbert第十问题: “多项式是否有整数根”有没有算法?

1970’s 被证明不可判定 (没有判定器, 即没有算法)

M = “对于输入 “p”, p是k元多项式,

1. 取k个整数的向量x (绝对值和从小到大)
2. 若p(x) = 0, 则停机接受.
3. 否则转1. ”

这个图灵机对输入 $p(x,y) = x^2+y^2-3$不停机

2）PCP 问题：Post Correspondence Problem

对比: 一个可判定问题

一元多项式是否有整数根?

M = "对于输入 "p", 1元k次多项式p(x),

1. 计算解的绝对值上界N
2. 对所有|x|≤N
3. 若p(x) = 0, 则停机接受.
4. 停机拒绝. "

对比: 一个可判定问题

设多项式 $c_1x^n + c_2x^{n-1} + \ldots + c_nx + c_{n+1}$ 有根 $x = x_0$,

c_{max}是c_i的最大绝对值. 证明 $|x_0| < (n+1)\, c_{max} / |c_1|$

解: 不妨设$c_1 \neq 0$.

若 $|x_0| \leq 1$, 则 $|x_0| \leq 1 \leq c_{max} / |c_1| < (n+1)\, c_{max} / |c_1|$, 性质成立

若 $|x_0| > 1$, 则由 $c_1x_0^n + c_2x_0^{n-1} + \ldots + c_nx_0 + c_{n+1} = 0$, 得

$$c_1x_0^n = -(c_2x_0^{n-1} + \ldots + c_nx_0 + c_{n+1}),$$

$$|c_1|\,|x_0|^n \leq |c_2||x_0|^{n-1} + \ldots + |c_n||x_0| + |c_{n+1}|,$$

$$|c_1|\,|x_0|^n < (n+1)c_{max}|x_0|^{n-1},$$

$$|x_0| < (n+1)\, c_{max} / |c_1|.$$

例如：$2x^3 + 3x^2 - 7x + 11 = 0$ $\quad |x_0| < (3+1)*11/2=22$

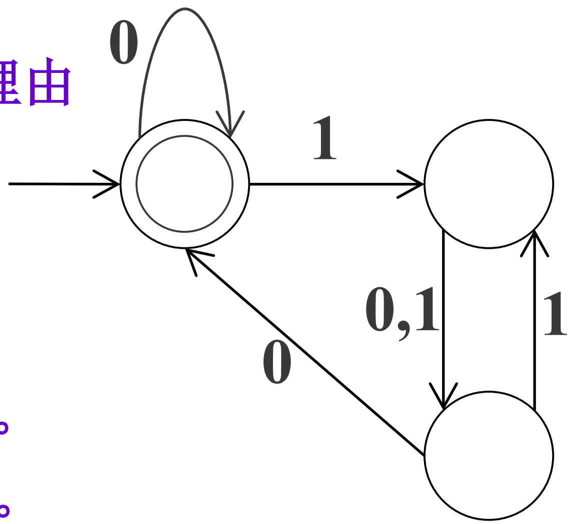
语言类的封闭性

	补	交	并	连接	星号
图灵可识别语言 0型，图灵机	x	√	√	√	√
图灵可判定语言 1型，判定器	√	√	√	√	√
上下文有关语言 1型， LBA	√	√	√	√	√
上下文无关语言 2型， PDA	x	x	√	√	√
正则语言 线性 3型，DFA	√	√	√	√	√

END

计算理论第4章作业

4.1 对于右图所示的DFA M, 回答下列问题，并说明理由



a. $\langle M,0100\rangle \in A_{DFA}$?　b. $\langle M,011\rangle \in A_{DFA}$?

c. $\langle M\rangle \in A_{DFA}$?

e. $\langle M\rangle \in E_{DFA}$?　　　f. $\langle M,M\rangle \in EQ_{DFA}$?

4.2 考虑一个DFA和一个正则表达式是否等价的问题。

将这个问题描述为一个语言并证明它是可判定的。

4.3 设 $ALL_{DFA} = \{\langle A\rangle \mid A$是一个识别$\Sigma^*$的DFA$\}$.

证明ALL_{DFA}可判定.

4.15 设$A = \{\langle R\rangle \mid R$是一个正则表达式，

其所描述的语言中至少有一个串w以111为子串$\}$.

证明A是可判定的。