可判定性

吴 铤

2017年11月13日

可判定性

1 可判定语言

2 不可判定性

可判定性

1 可判定语言

2 不可判定性

■ DFA的空性质测试: 判断某个DFA是否不接受任何字符 串。 $\Diamond E_{DFA} = \{(A) | A \not\in DFA, \exists L(A) = \emptyset\}$

定理 $1(E_{DFA}$ 是可判定语言)

证明.

对于输入 $\langle A \rangle$

■ 标记 4 的起始状态;

■ 型发以下涉等,且主被标比集合不叫

2511/22191 (011) 30,06511

画 加泉没有接受状态被标记。则接受: 否

■ DFA的空性质测试: 判断某个DFA是否不接受任何字符

串。 $\diamondsuit E_{DFA} = \{\langle A \rangle | A \not\in DFA, \ \underline{\perp}L(A) = \emptyset\}$

定理 $1(E_{DFA}$ 是可判定语言)

证明

- 标记A的起始状态;
- 重复以下步骤,直至被标记集合不增:
- 加里沿有接受状太被标记 刚接受,否则拓
- 🔢 如果沒有接受状念做杯记,则接受; 省则拒绝

■ DFA的空性质测试: 判断某个DFA是否不接受任何字符 串。令 $E_{DFA} = \{\langle A \rangle | A$ 是DFA, 且 $L(A) = \emptyset\}$

定理 $1(E_{DFA}$ 是可判定语言)

证明.

- 标记A的起始状态;
- ☑ 重复以下步骤,直全被标记集合不增:
 - 态出发的,则将其标记。
- 如果没有接受状态被标记,则接受;否则拒绝

■ DFA的空性质测试: 判断某个DFA是否不接受任何字符 串。令 $E_{DFA} = \{\langle A \rangle | A$ 是DFA, 且 $L(A) = \emptyset \}$

定理 $1(E_{DFA}$ 是可判定语言)

证明.

对于输入 $\langle A \rangle$

 \blacksquare 标记A的起始状态;

🛮 重复以卜步骤,直全被标记集合个增:

态出发的,则将其标记品

■ 如果没有接受状态被标记,则接受; 否则拒绝

■ DFA的空性质测试: 判断某个DFA是否不接受任何字符 串。令 $E_{DFA} = \{\langle A \rangle | A \neq DFA, \perp L(A) = \emptyset \}$

定理 $1(E_{DFA}$ 是可判定语言)

证明.

- 1 标记A的起始状态;
- 2 重复以下步骤,直至被标记集合不增:
- 对于未标记某个状态,如果有一个到达它的转移是从已标记状态出发的,则将其标记;
- 3 如果没有接受状态被标记,则接受;否则拒绝。



■ DFA的空性质测试: 判断某个DFA是否不接受任何字符 串。令 $E_{DFA} = \{\langle A \rangle | A$ 是DFA, 且 $L(A) = \emptyset \}$

定理1 (EDFA是可判定语言)

证明.

- 1 标记A的起始状态;
- 2 重复以下步骤, 直全被标记集合个增:
 - 对于未标记某个状态,如果有一个到达它的转移是从已标记状态。如果有一个到达它的转移是从已标记状态。
- 添山火田,州市大州。
- 3 如果没有接受状态被标记,则接受;否则拒绝。



■ DFA的空性质测试: 判断某个DFA是否不接受任何字符 串。令 $E_{DFA} = \{\langle A \rangle | A \neq DFA, \perp L(A) = \emptyset \}$

定理1 (EDFA是可判定语言)

证明.

对于输入〈A〉

- 标记A的起始状态;
- 2 重复以下步骤,直至被标记集合不增:
 - 1 对于未标记某个状态,如果有一个到达它的转移是从已标记状态出发的,则将其标记;
- 3 如果没有接受状态被标记,则接受;否则拒绝。



■ DFA的空性质测试: 判断某个DFA是否不接受任何字符 串。令 $E_{DFA} = \{\langle A \rangle | A$ 是DFA, 且 $L(A) = \emptyset \}$

定理 $1(E_{DFA}$ 是可判定语言)

证明.

- 1 标记A的起始状态;
- 2 重复以下步骤,直至被标记集合不增:
 - 对于未标记某个状态,如果有一个到达它的转移是从已标记状态出发的,则将其标记;
- 3 如果没有接受状态被标记,则接受,否则拒绝。



定理1 (EDFA是可判定语言)

证明.

- 1 标记A的起始状态;
- 2 重复以下步骤,直至被标记集合不增:
 - 对于未标记某个状态,如果有一个到达它的转移是从已标记状态出发的,则将其标记;
- 3 如果没有接受状态被标记,则接受;否则拒绝。



定理2 $(EQ_{DFA}=\{\langle A,B\rangle|A,B$ 是DFA, $L(A)=L(B)\}$ 是可判定语言)

证明.

考虑L(A), L(B)的对称差:

$$L(C) \stackrel{def}{=} L(A) \oplus L(B)$$
$$= (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$$

则就是要判断 $\langle C
angle\stackrel{?}{\in} E_{DFA}$ 。

定理2 $(EQ_{DFA} = \{\langle A,B \rangle | A,B$ 是DFA, $L(A) = L(B)\}$ 是可判定语言)

证明.

考虑L(A), L(B)的对称差:

$$L(C) \stackrel{def}{=} L(A) \oplus L(B)$$
$$= (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$$

则就是要判断 $\langle C
angle \stackrel{?}{\in} E_{DFA}$

定理2 $(EQ_{DFA} = \{\langle A, B \rangle | A, B \not\in DFA, L(A) = L(B)\}$ 是可判定语言)

证明.

考虑L(A), L(B)的对称差:

$$L(C) \stackrel{def}{=} L(A) \oplus L(B)$$
$$= (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$$

则就是要判断 $\langle C
angle \stackrel{?}{\in} E_{DFA}$ 。



证明.

考虑L(A), L(B)的对称差:

$$L(C) \stackrel{def}{=} L(A) \oplus L(B)$$
$$= (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$$

则就是要判断 $\langle C \rangle \stackrel{?}{\in} E_{DFA}$ 。

- DFA接受问题: 设B,w分别为给定的DFA和串,判断B是 否接受w。
 - 构造图灵机M如下
 - \blacksquare 检查输入的 $\langle B, w \rangle$;
 - \blacksquare 在图灵机M上模拟B在输入为w上的运行;
 - \blacksquare 如果最终B进入接受状态,则接受,否则拒绝;
- NFA接受问题: 设B,w分别为给定的NFA和串,判断B是 否接受w。
- 正则表达式的派生问题: 设R,w分别为给定的正则表达式和串,判断R是否派生w。

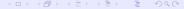


- DFA接受问题: 设B, w分别为给定的DFA和串,判断B是 否接受w。
 - 构造图灵机 M 如下
 - 检查输入的 $\langle B, w \rangle$;
 - \blacksquare 在图灵机M上模拟B在输入为w上的运行;
 - 如果最終B进入接受状态,则接受,否则拒绝;
- NFA接受问题:设B, w分别为给定的NFA和串,判断B是 不接受w。
- 正则表达式的派生问题: 设R,w分别为给定的正则表达式和串,判断R是否派生w。

- DFA接受问题: 设B,w分别为给定的DFA和串,判断B是 否接受w。
 - 构造图灵机M如下
 - 1 检查输入的 $\langle B, w \rangle$;
 - ② 在图灵机M上模拟B在输入为w上的运行;
 - 3 如果最终B进入接受状态,则接受,否则拒绝;
- NFA接受问题:设B, w分别为给定的NFA和串,判断B是 否接受w。
- 正则表达式的派生问题:设R,w分别为给定的正则表达式和串,判断R是否派生w。



- DFA接受问题: 设B, w分别为给定的DFA和串,判断B是 否接受w。
 - 构造图灵机M如下
 - **1** 检查输入的 $\langle B, w \rangle$;
 - 2 在图灵机M上模拟B在输入为w上的运行;
 - 3 如果最终B进入接受状态,则接受,否则拒绝;
- NFA接受问题:设B, w分别为给定的NFA和串,判断B是 不接受w。
- 正则表达式的派生问题:设R,w分别为给定的正则表达式和串,判断R是否派生w。



- DFA接受问题: 设B, w分别为给定的DFA和串,判断B是 否接受w。
 - 构造图灵机 M 如下
 - **1** 检查输入的 $\langle B, w \rangle$;
 - 2 在图灵机M上模拟B在输入为w上的运行;
 - $\mathbf{3}$ 如果最终B进入接受状态,则接受,否则拒绝;
- NFA接受问题:设B, w分别为给定的NFA和串,判断B是 不接受w。
- 正则表达式的派生问题: 设R,w分别为给定的正则表达式和串,判断R是否派生w。



- DFA接受问题: 设B, w分别为给定的DFA和串,判断B是 否接受w。
 - 构造图灵机 M 如下
 - **1** 检查输入的 $\langle B, w \rangle$;
 - 2 在图灵机M上模拟B在输入为w上的运行;
 - 3 如果最终B进入接受状态,则接受,否则拒绝;
- NFA接受问题:设B, w分别为给定的NFA和串,判断B是 不接受w。
- 正则表达式的派生问题: 设R,w分别为给定的正则表达式和串,判断R是否派生w。

- DFA接受问题: 设B, w分别为给定的DFA和串,判断B是 否接受w。
 - 构造图灵机 M如下
 - **1** 检查输入的 $\langle B, w \rangle$;
 - 2 在图灵机M上模拟B在输入为w上的运行;
 - 3 如果最终B进入接受状态,则接受,否则拒绝;
- NFA接受问题: 设B, w分别为给定的NFA和串,判断B是 否接受w。
- 正则表达式的派生问题: 设R,w分别为给定的正则表达式和串,判断R是否派生w。



- DFA接受问题: 设B, w分别为给定的DFA和串,判断B是 否接受w。
 - 构造图灵机M如下
 - **1** 检查输入的 $\langle B, w \rangle$;
 - 2 在图灵机M上模拟B在输入为w上的运行;
 - 3 如果最终B进入接受状态,则接受,否则拒绝;
- NFA接受问题: 设B,w分别为给定的NFA和串,判断B是 否接受w。
- 正则表达式的派生问题: 设R,w分别为给定的正则表达式和串,判断R是否派生w。

$$A_{CFG} = \{\langle G, w \rangle | G \not\in CFG, G$$
派生串 $w \}$
 $E_{CFG} = \{\langle G \rangle | G \not\in CFG, L(G) = \emptyset \}$
 $EQ_{CFG} = \{\langle G, H \rangle | G, H$ 都是 $CFG, L(G) = L(H) \}$

证明:按如下方式构造TM S:

- 将G转换为与之等价的乔姆斯基范式;
- 2 列出所有2n-1步的派生,其中n是w的长度;
- $\mathbf B$ 如果其中有一个产生w,则接受,否则拒绝。

证明: 按如下方式构造TM S:

对于输入 $\langle G, w \rangle$,

- \blacksquare 将G转换为与之等价的乔姆斯基范式;
- ② 列出所有2n-1步的派生,其中n是w的长度;
- **3** 如果其中有一个产生w,则接受,否则拒绝。

证明:接如下方式构造TM S:

对于输入 $\langle G, w \rangle$,

- \blacksquare 将G转换为与之等价的乔姆斯基范式;
- ② 列出所有2n-1步的派生,其中n是w的长度;
- \mathbf{B} 如果其中有一个产生w,则接受,否则拒绝。

证明:按如下方式构造TM S: 对于输入 $\langle G, w \rangle$,

- 将G转换为与之等价的乔姆斯基范式;
- ② 列出所有2n-1步的派生,其中n是w的长度;
- **3** 如果其中有一个产生w,则接受,否则拒绝。

定理3 (A_{CFG}是可判定语言)

证明:按如下方式构造TM S: 对于输入 $\langle G, w \rangle$,

- 将G转换为与之等价的乔姆斯基范式;
- ② 列出所有2n-1步的派生,其中n是w的长度;
- \mathbf{B} 如果其中有一个产生w,则接受,否则拒绝。

证明:接如下方式构造TM S:对于输入 $\langle G, w \rangle$,

- 将G转换为与之等价的乔姆斯基范式;
- ② 列出所有2n-1步的派生,其中n是w的长度;
- 3 如果其中有一个产生w,则接受,否则拒绝。

定理 $4(E_{CFG}$ 是可判定语言)

证明.

对于输入的CFG(G):

- 将G中所有的终结符全都作上标记;
- 重复以下步骤, 直至找不到可以标记的变元:
 - in $\mathcal{H}G\mathcal{H}[M]M \rightarrow U_1U_2\cdots U_n$, $MU_1, i=1, \cdots, k$ is in $\mathcal{H}^{-1}M$ in $\mathcal{$
- 如果起始状态没有被标记,则接受,否则拒绝,

可判定语言

定理4 (E_{CFG}是可判定语言)

证明.

对于输入的 $CFG\langle G \rangle$:

- 将*G*中所有的终结符全都作上标记;
- 2 重复以下步骤, 直至找不到可以标记的变元:
 - 若G有规则 $A \rightarrow U_1U_2 \cdots U_k$,且 U_i , $i = 1, \cdots, k$ 中的每个符号都已标记,则标记变元A:
- 3 如果起始状态没有被标记,则接受,否则拒绝。

定理5 (EQCFG不是可判定的)

定理4 (E_{CFG}是可判定语言)

证明.

对于输入的 $CFG\langle G \rangle$:

- 将*G*中所有的终结符全都作上标记;
- 2 重复以下步骤,直至找不到可以标记的变元:
 - 看G有规则 $A \to U_1U_2 \cdots U_k$,且 U_i , $i = 1, \cdots, k$ 中的每个符号都已标记,则标记变元A.
- 3 如果起始状态没有被标记,则接受,否则拒绝。

定理5 (EQCFG不是可判定的)

定理4 (E_{CFG}是可判定语言)

证明.

对于输入的 $CFG\langle G \rangle$:

- 将*G*中所有的终结符全都作上标记;
- 2 重复以下步骤, 直至找不到可以标记的变元:
 - 若G有规则 $A \rightarrow U_1U_2 \cdots U_k$,且 U_i , $i = 1, \cdots, k$ 中的每个符号都已标记,则标记变元A;
- 3 如果起始状态没有被标记,则接受,否则拒绝。

定理5 (EQCFG不是可判定的)

与上下文无关语言相关的可判定型问题

定理4(E_{CFG}是可判定语言)

证明.

对于输入的 $CFG\langle G \rangle$:

- 将*G*中所有的终结符全都作上标记;
- 2 重复以下步骤, 直至找不到可以标记的变元:
 - **I** 若G有规则 $A \rightarrow U_1U_2 \cdots U_k$,且 U_i , $i = 1, \cdots, k$ 中的每个符号都已标记,则标记变元A;
- 3 如果起始状态没有被标记,则接受,否则拒绝。

定理 $5(EQ_{CFG}$ 不是可判定的)

与上下文无关语言相关的可判定型问题

定理4(ECFG是可判定语言)

证明.

对于输入的 $CFG\langle G \rangle$:

- 1 将*G*中所有的终结符全都作上标记;
- 2 重复以下步骤, 直至找不到可以标记的变元:
 - **■** 若G有规则 $A \rightarrow U_1U_2 \cdots U_k$,且 U_i , $i = 1, \cdots, k$ 中的每个符号都已标记,则标记变元A;
- 3 如果起始状态没有被标记,则接受,否则拒绝。

定理5 (EQCFG不是可判定的)

与上下文无关语言相关的可判定型问题

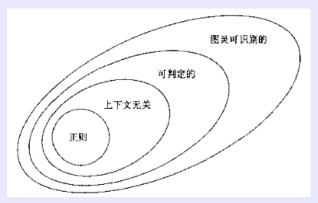
证明.

对于输入的 $CFG\langle G \rangle$:

- 1 将*G*中所有的终结符全都作上标记;
- 2 重复以下步骤, 直至找不到可以标记的变元:
 - **I** 若G有规则 $A \rightarrow U_1U_2 \cdots U_k$,且 U_i , $i = 1, \cdots, k$ 中的每个符号都已标记,则标记变元A;
- 3 如果起始状态没有被标记,则接受,否则拒绝。

定理5 (EQCFG不是可判定的)





停机问题: $A_{TM} = \{\langle M, w \rangle | M$ 是一个TM, 且接受w}

定理 $6(A_{TM}$ 是可识别不可判定的)

可识别的证明.

设置通用图灵机U,对于输入 $\langle M, w \rangle$ 执行:

- \blacksquare 对于输入w,模拟M;
- 若M进入接受状态,则接受;
 - 如果进入拒绝状态,则拒绝。

停机问题: $A_{TM} = \{\langle M, w \rangle | M$ 是一个TM, 且接受w}

定理 $6(A_{TM}$ 是可识别不可判定的)

可识别的证明.

设置通用图灵机U,对于输入 $\langle M, w \rangle$ 执行:

- $\mathbf{1}$ 对于输入w,模拟M
- 若*M*进入接受状态,则接受;; 如果进入拒绝状态,则拒绝。

停机问题: $A_{TM} = \{\langle M, w \rangle | M$ 是一个TM, 且接受w}

定理 $6(A_{TM}$ 是可识别不可判定的)

可识别的证明.

设置通用图灵机U,对于输入 $\langle M, w \rangle$ 执行:

- 对于输入w,模拟M;
- 至 若M进入接受状态,则接受;如果进入拒绝状态,则拒绝。



- 对于任意字母表∑,其上所有串构成的集合∑*必定可数;
- 由所有图灵机构成的集合是可数的;
- 字母表∑上所有语言构成的集合£是不可数的。

- 对于任意字母表∑,其上所有串构成的集合∑*必定可数;
- 由所有图灵机构成的集合是可数的;
- 字母表∑上所有语言构成的集合£是不可数的。

- 对于任意字母表∑,其上所有串构成的集合∑*必定可数;
- 由所有图灵机构成的集合是可数的;
- ■字母表∑上所有语言构成的集合£是不可数的。

- 对于任意字母表∑,其上所有串构成的集合∑*必定可数;
- 由所有图灵机构成的集合是可数的;
- 字母表∑上所有语言构成的集合£是不可数的。

- 对于任意字母表∑,其上所有串构成的集合∑*必定可数;
- 由所有图灵机构成的集合是可数的;
- 字母表∑上所有语言构成的集合£是不可数的。

假设 A_{TM} 是可判定的,H为其判定器。即对于输入 $\langle M, w \rangle$

- 若M接受w,则H接受 $\langle M, w \rangle$;
- 若M不接受w,则H拒绝 $\langle M, w \rangle$;

$$H(\langle M, w \rangle) = \begin{cases}$$
接受 若 M 接受 w
拒绝 若 M 不接受 w



假设 A_{TM} 是可判定的,H为其判定器。即对于输入 $\langle M,w\rangle$

- 若M接受w,则H接受 $\langle M, w \rangle$;
- 若M不接受w,则H拒绝 $\langle M, w \rangle$;

$$H(\langle M, w \rangle) = \begin{cases} \text{接} \\ \text{接} \\ \text{持} \end{cases} \quad \text{若} M \text{接} \\ \text{接} \\ \text{持} \end{cases}$$

假设 A_{TM} 是可判定的,H为其判定器。即对于输入 $\langle M,w \rangle$

- 若M接受w,则H接受 $\langle M, w \rangle$;
- 若M不接受w,则H拒绝 $\langle M, w \rangle$;

$$H(\langle M, w \rangle) = \begin{cases}$$
接受 若 M 接受 w
拒绝 若 M 不接受 w

假设 A_{TM} 是可判定的,H为其判定器。即对于输入 $\langle M,w \rangle$

- 若M接受w,则H接受 $\langle M, w \rangle$;
- 若M不接受w,则H拒绝 $\langle M, w \rangle$;

則

$$H(\langle M, w \rangle) = \begin{cases} \text{接} \mathfrak{G} & \text{若} M \text{接} \mathfrak{G} w \\ \text{拒绝} & \text{若} M \text{不} \text{接} \mathfrak{G} w \end{cases}$$

假设 A_{TM} 是可判定的,H为其判定器。即对于输入 $\langle M,w \rangle$

- 若M接受w,则H接受 $\langle M, w \rangle$;
- 若M不接受w,则H拒绝 $\langle M, w \rangle$;

$$H(\langle M, w \rangle) =$$
 接受 若 M 接受 w 拒绝 若 M 不接受 w



■ 若M接受 $\langle M \rangle$,则拒绝;

■ 若M不接受 $\langle M \rangle$,则接受;

即

$$D(\langle M \rangle) = \begin{cases} \dot{\mathbf{E}} & \ddot{\mathbf{E}} H(\langle M, \langle M \rangle \rangle) = \dot{\mathbf{E}} \mathbf{E} \\ \dot{\mathbf{E}} & \ddot{\mathbf{E}} H(\langle M, \langle M \rangle \rangle) = \dot{\mathbf{E}} \mathbf{E} \end{cases}$$

- 若M不接受 $\langle M \rangle$,则接受;

即

$$D(\langle M \rangle) = \begin{cases} \text{接受} & \text{若}H(\langle M, \langle M \rangle \rangle) = \text{拒绝} \\ \text{拒绝} & \text{若}H(\langle M, \langle M \rangle \rangle) = \text{接受} \end{cases}$$

- 者M接受 $\langle M \rangle$,则拒绝;
- 若M不接受 $\langle M \rangle$,则接受;

即

$$D(\langle M \rangle) = \begin{cases} \text{接受} & \text{若}H(\langle M, \langle M \rangle \rangle) = \text{拒绝} \\ \text{拒绝} & \text{若}H(\langle M, \langle M \rangle \rangle) = \text{接受} \end{cases}$$

$$D(\langle D \rangle) = egin{cases}$$
接受 若 D 不接受 $\langle D \rangle \$ 拒绝 若 D 接受 $\langle D \rangle \ \end{cases}$

- 者M接受 $\langle M \rangle$,则拒绝;
- 若M不接受 $\langle M \rangle$,则接受;

即

$$D(\langle M \rangle) = \begin{cases} \dot{\mathbf{E}} & \ddot{\mathbf{E}} H(\langle M, \langle M \rangle \rangle) = \mathbf{E} \\ \dot{\mathbf{E}} & \ddot{\mathbf{E}} H(\langle M, \langle M \rangle \rangle) = \dot{\mathbf{E}} \\ \end{cases}$$

- 若M接受 $\langle M \rangle$,则拒绝;
- 若M不接受 $\langle M \rangle$,则接受;

即

- 若M接受 $\langle M \rangle$,则拒绝;
- 若M不接受 $\langle M \rangle$,则接受;

即

- 者M接受 $\langle M \rangle$,则拒绝;
- 若M不接受 $\langle M \rangle$,则接受;

即

定义1(补图灵可识别)

若某个语言是一个图灵可识别语言的补集,则称其 是补图灵可识别的。

定理8

某个语言可判定的充要条件是:其即是图灵可识别的, C是补图灵可识别的。

定义1(补图灵可识别)

若某个语言是一个图灵可识别语言的补集,则称其 是补图灵可识别的。

定理8

某个语言可判定的充要条件是: 其即是图灵可识别的, 又是补图灵可识别的。

定义1(补图灵可识别)

若某个语言是一个图灵可识别语言的补集,则称其 是补图灵可识别的。

定理8

某个语言可判定的充要条件是: 其即是图灵可识别的, 又是补图灵可识别的。

图灵机M的构造:对于输入w

- I 对输入w并行地运行 M_1, M_2
- 2 如果 M_1 接受,则接受; 如果 M_2 接受,则拒绝

定理 $9(\overline{A_{TM}}$ 不是图灵可识别的)

器,则

图灵机M的构造:对于输入w

- I 对输入w并行地运行 M_1, M_2 ;
- **2** 如果 M_1 接受,则接受; 如果 M_2 接受,则拒绝

定理 $9(\overline{A_{TM}}$ 不是图灵可识别的)



图灵机M的构造:对于输入w

- I 对输入w并行地运行 M_1, M_2 ;
- ② 如果 M_1 接受,则接受; 如果 M_2 接受,则拒绝

定理 $9(\overline{A_{TM}}$ 不是图灵可识别的

图灵机M的构造: 对于输入w

- I 对输入w并行地运行 M_1, M_2 ;
- 2 如果 M_1 接受,则接受; 如果 M_2 接受,则拒绝

定理 $9(A_{TM}$ 不是图灵可识别的)

图灵机M的构造:对于输入w

- I 对输入w并行地运行 M_1, M_2 ;
- 2 如果 M_1 接受,则接受; 如果 M_2 接受,则拒绝

定理 $9(A_{TM}$ 不是图灵可识别的

器,则

图灵机M的构造:对于输入w

- I 对输入w并行地运行 M_1, M_2 ;
- 2 如果 M_1 接受,则接受; 如果 M_2 接受,则拒绝

定理 $9(\overline{A_{TM}}$ 不是图灵可识别的)

$$HALT_{TM} = \{\langle M, w \rangle | M$$
是一个 TM , 且对输入 w 停机}
$$E_{TM} = \{\langle M \rangle | M$$
是一个 TM , $L(M) = \emptyset \}$
$$REGULAR_{TM} = \{\langle M \rangle | M$$
是一个 TM , $L(M)$ 是正则语言 }
$$EQ_{TM} \{\langle M_1, M_2 \rangle | M_1, M_2$$
是 TM , $L(M_1) = L(M_2) \}$

定理 $10 (HALT_{TM}$ 是不可判定的)

证明: 假设TM R判定 $HALT_{TM}$,则构造TM S如下: 对于输入 $\langle M, w \rangle$,S执行

- \blacksquare 对于输入 $\langle M, w \rangle$ 执行TM R;
- ② 如果*R*拒绝,则拒绝;否则
 - \blacksquare 在w上模拟M,直至其停机
 - \square 如果M接受,则接受,否则拒绝。

则 $_{
m IM}$ $_{S}$ 是判定 $_{A_{TM}}$ 的图灵机,矛盾。

定理 $10 (HALT_{TM}$ 是不可判定的)

证明: 假设TM R判定 $HALT_{TM}$,则构造TM S如下:

对于输入 $\langle M, w \rangle$,S执行

- 对于输入 $\langle M, w \rangle$ 执行TM R;
- ② 如果*R*拒绝,则拒绝;否则
- \blacksquare 在w上模拟M,直至其停机
 - \square 如果M接受,则接受,否则拒绝。
- 则 $_{
 m IM}$ $_{S}$ 是判定 $_{A_{TM}}$ 的图灵机,矛盾。

定理 $10 (HALT_{TM}$ 是不可判定的)

证明: 假设TM R判定 $HALT_{TM}$,则构造TM S如下: 对于输入 $\langle M, w \rangle$,S执行

- 对于输入 $\langle M, w \rangle$ 执行TM R;
- \mathbf{D} 如果R拒绝,则拒绝;否则
- \blacksquare 在w上模拟M,直至其停机
 - 如果M接受,则接受,否则拒绝。
- 则 $\mathrm{TM}\ S$ 是判定 A_{TM} 的图灵机,矛盾。

- I 对于输入 $\langle M, w \rangle$ 执行TM R;
- ② 如果*R*拒绝,则拒绝;否则
 - \mathbf{E}_{w} **工** \mathbf{E}_{w} **工** \mathbf{E}_{w} **工** \mathbf{E}_{w} **L** \mathbf{E}_{w} **E** \mathbf{E}_{w} **E**
- 则 $\mathrm{TM}\ S$ 是判定 A_{TM} 的图灵机,矛盾。

证明: 假设TM R判定 $HALT_{TM}$,则构造TM S如下: 对于输入 $\langle M, w \rangle$,S执行

- I 对于输入 $\langle M, w \rangle$ 执行TM R;
- 2 如果R拒绝,则拒绝;否则

■ 在w上模拟M, 直至其停机② 如果M接受,则接受,否则拒绝。

则 $_{
m IM}$ $_{S}$ 是判定 $_{A_{TM}}$ 的图灵机,矛盾。

证明: 假设TM R判定 $HALT_{TM}$,则构造TM S如下: 对于输入 $\langle M, w \rangle$,S执行

- I 对于输入 $\langle M, w \rangle$ 执行TM R;
- 2 如果R拒绝,则拒绝;否则
 - 1 在w上模拟M,直至其停机
 - 2 如果M接受,则接受,否则拒绝。

则 $_{
m IM}$ $_{S}$ 是判定 $_{A_{TM}}$ 的图灵机,矛盾。

证明: 假设TM R判定 $HALT_{TM}$,则构造TM S如下: 对于输入 $\langle M, w \rangle$,S执行

- I 对于输入 $\langle M, w \rangle$ 执行TM R;
- 2 如果R拒绝,则拒绝;否则
 - 在w上模拟M,直至其停机
 - 2 如果M接受,则接受,否则拒绝。

则 $_{
m IM}$ $_{S}$ 是判定 $_{A_{TM}}$ 的图灵机,矛盾。

证明: 假设TM R判定 $HALT_{TM}$,则构造TM S如下: 对于输入 $\langle M, w \rangle$,S执行

- I 对于输入 $\langle M, w \rangle$ 执行TM R;
- 2 如果R拒绝,则拒绝;否则
 - 在w上模拟M,直至其停机
 - 2 如果M接受,则接受,否则拒绝。

则TM S是判定 A_{TM} 的图灵机,矛盾。

- 1 构建TM M₁如下:
 - ,如果输入 $x \neq w$ 则拒绝,否则在x上运行M,当M接受时,接受;
- 2 在 $\langle M_1 \rangle$ 上运行R:
 - 如果R接受,则拒绝;如果R拒绝,则接受。

证明:假设TM R判定 E_{TM} ,则构造TM S如下:

对于输入 $\langle M, w \rangle$, 执行

- I 构建 $TM M_1$ 如下
 - 如果输入 $x \neq w$ 则拒绝,否则在x上运行M,当M接受时,接受:
- 2 在 $\langle M_1 \rangle$ 上运行R
 - 如果R接受,则拒绝;如果R拒绝,则接受。

- 构建TM *M*₁如下:
 - 如果输入 $x \neq w$ 则拒绝,否则在x上运行M,当M接受时,接受;
- 2 在 $\langle M_1 \rangle$ 上运行R:
 - 如果R接受,则拒绝;如果R拒绝,则接受。

- 构建TM M₁如下:
 - 如果输入 $x \neq w$ 则拒绝,否则在x上运行M,当M接受时,接受;
- 2 在 $\langle M_1 \rangle$ 上运行R:
 - 如果R接受,则拒绝;如果R拒绝,则接受。

- 构建TM M₁如下:
 - 如果输入 $x \neq w$ 则拒绝,否则在x上运行M,当M接受时,接受;
- 2 在 $\langle M_1 \rangle$ 上运行R
 - 如果R接受,则拒绝;如果R拒绝,则接受。

- 构建TM M₁如下:
 - 如果输入 $x \neq w$ 则拒绝,否则在x上运行M,当M接受时,接受;
- 2 在 $\langle M_1 \rangle$ 上运行R:
 - 如果*R*接受,则拒绝;如果*R*拒绝,则接受。

- 构建TM M₁如下:
 - 如果输入 $x \neq w$ 则拒绝,否则在x上运行M,当M接受时,接受;
- 2 在 $\langle M_1 \rangle$ 上运行R:
 - 如果*R*接受,则拒绝;如果*R*拒绝,则接受。