不可判定问题

不可判定问题

- 图灵机编码
- 对角线语言与通用语言
- 图灵机语言的性质
- 判定问题与语言
- 计算复杂性问题

不可判定问题

- 图灵机编码
- 对角线语言与通用语言
- 图灵机语言的性质
- 判定问题与语言
- 计算复杂性问题

语言层次

图灵机的数码表示

- 对图灵机作一些假定：

  (1) 输入字母表为{0,1}

  (2) 设有限状态为q_1， q_2， … ， q_k，并假定初态总是q_1 ，终态总是 q_2（假定一个终态即可）

  (3) 设带符号为X_1， X_2， … ， X_m，并假定 X_1 总代表 0，X_2 总代表 1， X_3 总代表 B

  (4) 假定带头移动方向为D_1 和 D_2，分别代表 L 和 R

图灵机编码

- 图灵机的二进制编码：状态转移函数

$$\delta(q_i, X_j) = (q_k, X_l, D_m)$$

可以编码为：　$0^i10^j10^k10^l10^m$

所有转移规则的编码排列在一起可以作为该图灵机的编码。例如:

$$11C_111C_211\ldots C_{n-1}11C_n$$

图灵机编码

- 若一图灵机的二进制编码为 w_i，而 w_i 为对应整数 i 的 0，1 字符串，则称该图灵机为第 i 个图灵机。

例：

$$w_{128}=10000000$$

对应第128个图灵机。

图灵机编码

例: $M = (\{q_1, q_2\}, \{0,1\}, \{0,1,B\}, \delta, q_1, B, \{q_2\})$

状态转移 δ:	对应编码:
$\delta(q_1, 1) = (q_3, 0, R)$	0100100010100
$\delta(q_3, 0) = (q_1, 1, R)$	0001010100100
$\delta(q_3, 1) = (q_2, 0, R)$	00010010010100
$\delta(q_3, B) = (q_3, 1, L)$	0001000100010010

图灵机的编码为:

11010010001010011000101010010011000100100101001 1000100010001001 0

2^63+2^62+2^60+2^57+2^53+2^51+2^48+2^47+2^43+2^41+2^39+2^36
+2^33+2^32+2^28+2^25+2^22+2^20+2^17+2^16+2^12+2^8+2^4+2
= 1.514378978530552e+19

图灵机编码

- 0，1 输入字符串的编码
  - 将任意 0，1 字符串 w 用 $1w$ 编码.
    如：ε 编码为 1，0 编码为 10，1 编码为 11，00 编码为 100 等。
  - 任何一个输入串 w，可以对应到某个整数 j，称之为第 j 个字符串。
    如：w=0000000，对应第128个输入串。

不可判定问题

- 图灵机编码
- 对角线语言与通用语言
- 图灵机语言的性质
- 判定问题与语言
- 计算复杂性问题

字符串编码

- ## 图灵机编码

  每个图灵机对应一个整数 i，即该图灵机的二进制编码 w_i 是第 i 个 0，1 字符串。

- ## 输入串二进制编码

  任何一个输入串 w，可以对应到某个整数 j，为第 j 个输入字符串。

对角线语言

说明：不是每个整数 j 都能对应一个图灵机。若整数 j 没有对应的图灵机，认为第 j 个图灵机不接受任何字符串，即 $L(M_j)=\varnothing$

- 对角线语言 L_d

  定义：对任何 $i\geq 1$，设M_i为第 i 个图灵机。称语言

$$L_d=\{ w_i \mid w_i \notin L(M_i) \}$$

为对角线（对角化）语言。

对角线语言

对角线语言 $L_d = \{ w_i \mid w_i \notin L(M_i) \}$.

$j \longrightarrow$

$i \downarrow$		1	2	3	4	…
	1	0	1	1	0	…
	2	1	1	0	0	…
	3	0	0	1	1	…
	4	0	1	0	1	…
	.	.	.	.	.	
	.	.	.	.	.	
	.	.	.	.	.	

Diagonal

图灵机 M_i 是否接受输入 w_j?

1 表示“接受”；

0 表示“拒接”

对角线语言

定理: L_d不是递归可枚举语言。

证明：若存在某个图灵机 M，满足 $L(M)=L_d$.

设 M 是第 k 个图灵机，即 $M = M_k$，

对于第 k 个 0，1 输入字符串w_k，是否有

$$w_k \in L_d ?$$

这是一个悖论。因此，不存在这样的 M.

因此L_d不是递归可枚举语言。

通用图灵机

- 通用语言
  - 设 M 为接受二进制输入串的图灵机，M的二进制编码为 C，w 为$(0+1)^*$ 中的串C'
  - 图灵机与输入串偶对 (M, w) 的二进制编码记为:

$$C\,111\,C'$$

- 定义：称语言

$$L_u=\{C\,111\,C' \mid (M, w) \in\{0,1\}^*, w \in L(M)\}$$

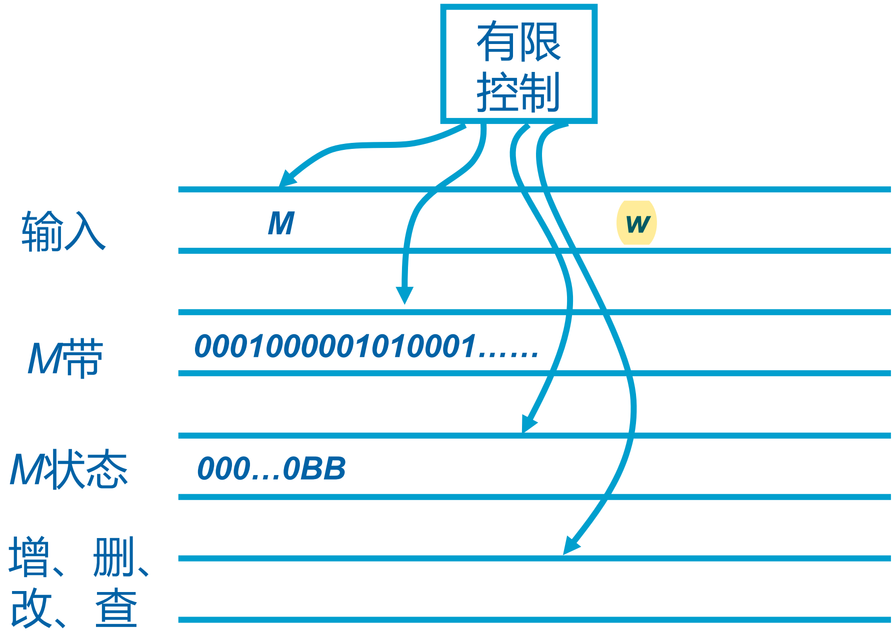
为通用语言。

通用图灵机

构造一多带图灵机 U（右图所示）。

U 接受偶对 (M,w)，当且仅当 $w \in L(M)$，称 U 为通用图灵机。

显然，$L_u=L(U)$.



Theorem: 通用语言 L_u 为递归可枚举语言.

不可判定问题

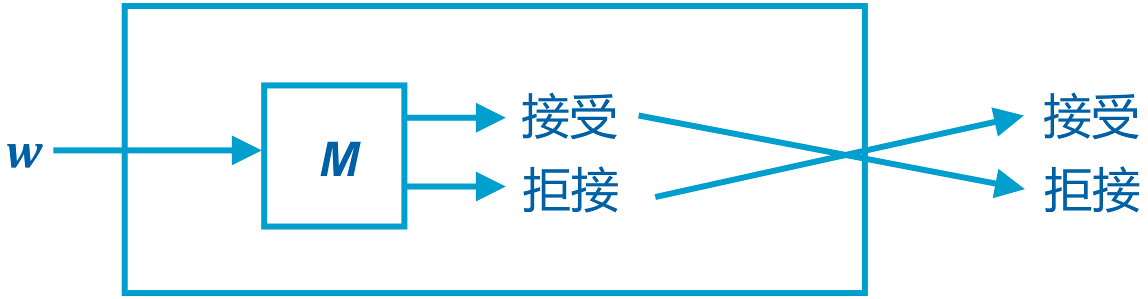
- 图灵机编码
- 对角线语言与通用语言
- 图灵机语言的性质
- 判定问题与语言
- 计算复杂性问题

递归语言的性质

定理: 若 L 是递归语言，则 L的补$\bar{L}$也是递归语言。

证明思路:

设图灵机 M 总可以停机，且满足 $L = L(M)$。对 M 进行如下修改，以构造图灵机 $\bar{M}$（参见下图）：



递归语言的性质

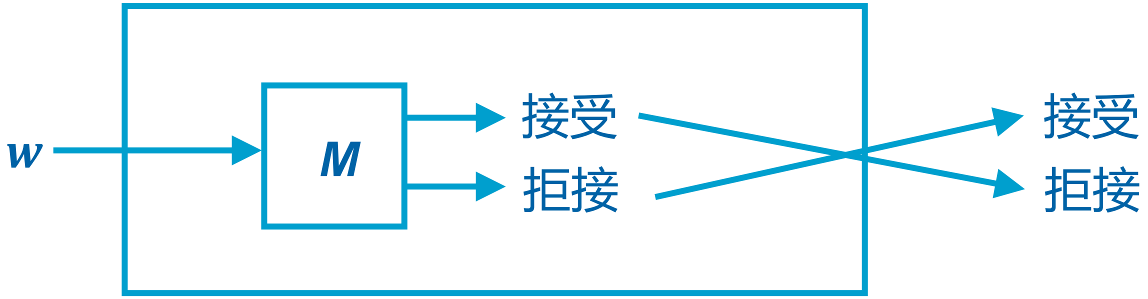
1. 将 M 的终态作为 $\bar{M}$ 的非终态。
2. 增加新的终态 r，且 $\bar{M}$ 在状态 r 没有转移。
3. 对 M 中每一非终态 q，以及带符 X，增加转移

$$\delta(q, X) = (r, Y, D)$$

其中 Y 和 D 可任取。

显然，$\bar{L} = L(\bar{M})$,

且 L 是递归语言。



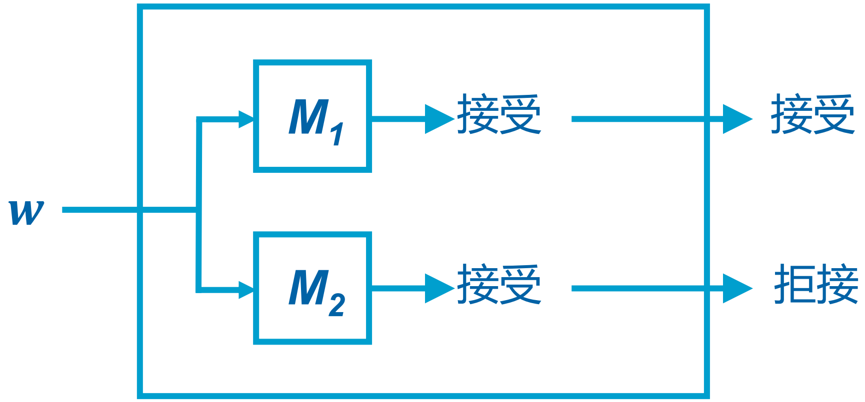
递归可枚举语言的性质

定理: 若语言L和补$\overline{L}$都是递归可枚举语言，则语言L和补$\overline{L}$均为递归语言。

证明思路:

设$L = L(M_1)$，$\overline{L} = L(M_2)$，构造图灵机M模拟M_1和M_2的并行执行。

（下图所示）



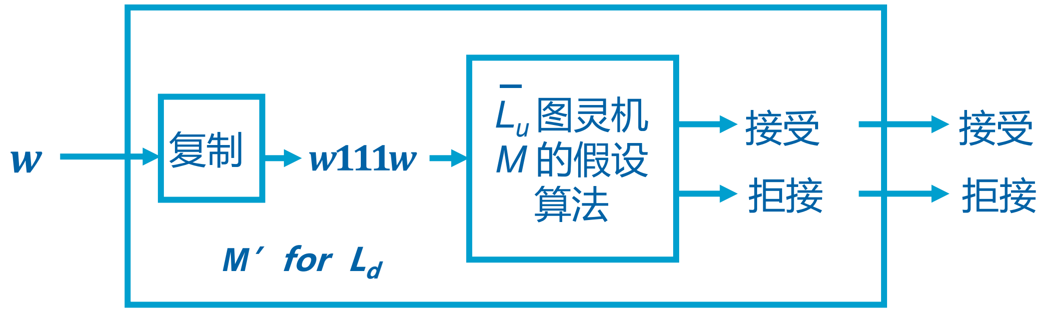
无论输入串w是否属于L，M总是能够停机。

递归可枚举语言的性质

定理：通用语言L_u是递归可枚举语言，但不是递归语言。

证明思路：

若L_u是递归，则$\bar{L}_u$也是递归。假定 $\bar{L}_u = L(M)$，由M 构造图灵机 M' (见右图)，使得 $L_d = L(M')$，则语言 L_d 也是递归可枚举语言，矛盾。

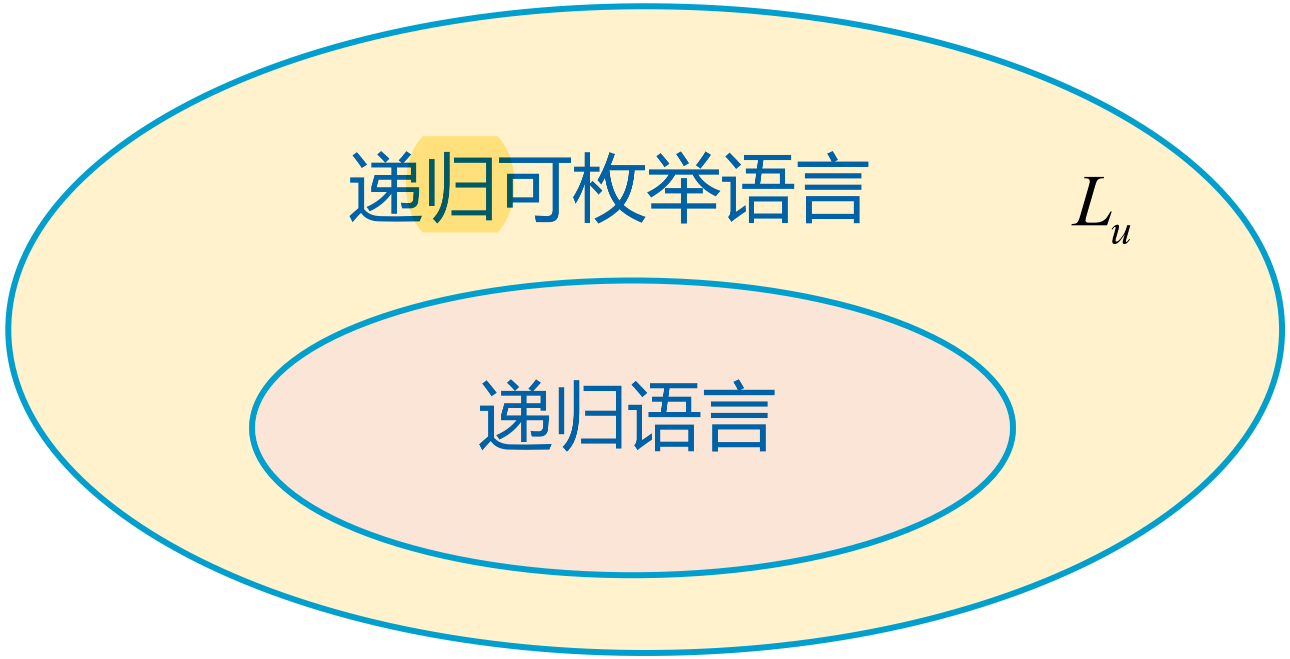


结论：通用语言 L_u 的补不是递归可枚举的。

推论：递归可枚举语言对补运算不是封闭的。

递归可枚举语言的性质

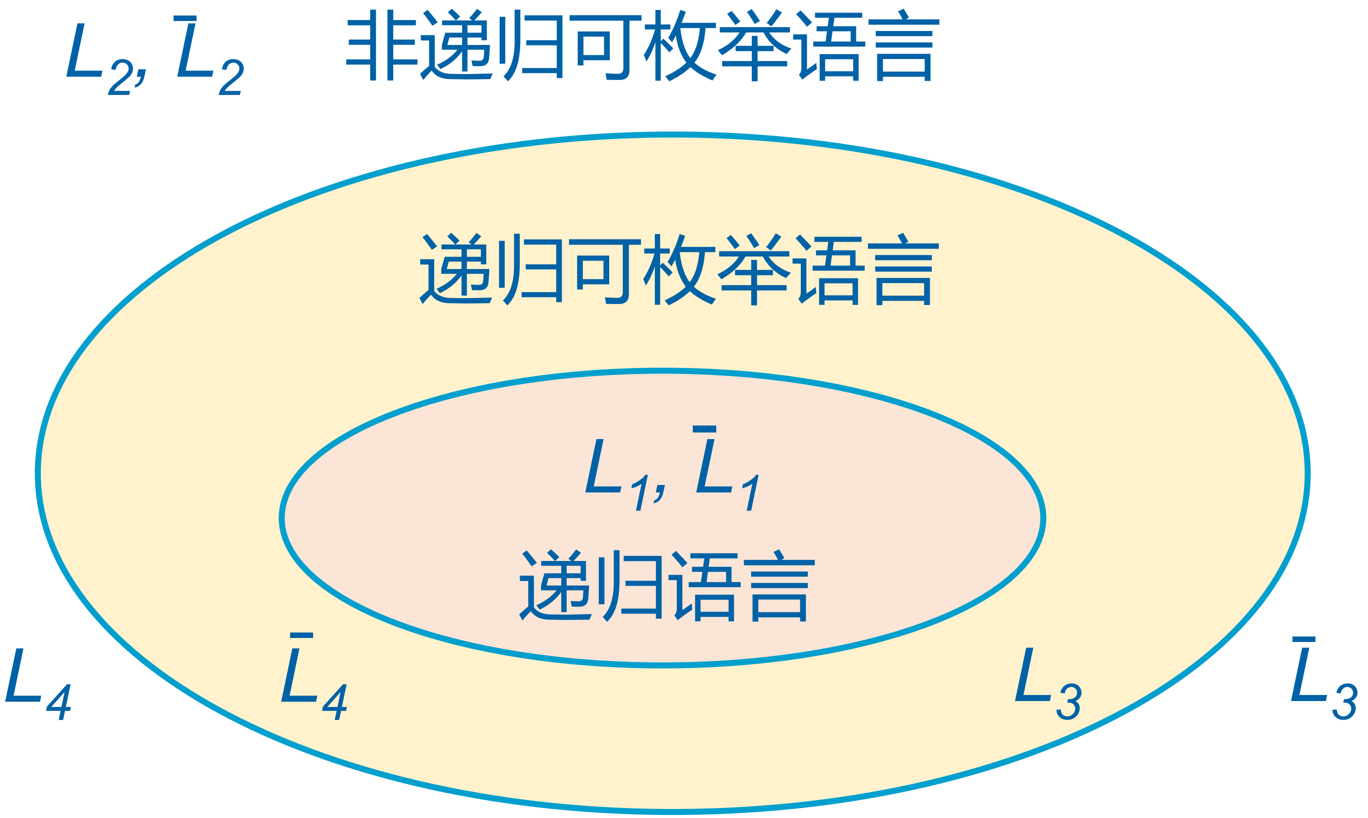
L_d 非递归可枚举语言 $\overline{L_u}$



语言分类

- L 和 $\overline{L}$ 都是递归, 即二者均在内环。
- L 和 $\overline{L}$ 都不是递归可枚举语言, 即二者均在外环。
- L 是递归可枚举语言但非递归语言，且 $\overline{L}$ 不是递归可枚举语言。即一个在中环，另一个在外环。
- $\overline{L}$ 是递归可枚举语言但非递归语言，且 L 不是递归可枚举语言。结论与上述相反。

非递归可枚举语言



$L_2, \overline{L}_2$ 非递归可枚举语言
递归可枚举语言
$L_1, \overline{L}_1$
递归语言
L_4
$\overline{L}_4$
L_3
$\overline{L}_3$

不可判定问题

- 图灵机编码
- 对角线语言与通用语言
- 图灵机语言的性质
- 判定问题与语言
- 计算复杂性问题

问题与语言

- 语言对应问题

设 $L \subseteq \Sigma^*$ 是字母表 Σ 上的一个语言，则与语言 L 对应的判定问题定义为：

任给一个串$w \in \Sigma^*$，判定$w \in L$ 是否成立?

例：

通用语言 L_u对应的问题为： 任给图灵机 M 和输入w，判定w 是否被 M 接受?

问题与语言

- 问题对应语言

例：图灵机停机问题：任给图灵机M和输入字符串w，试问对于w，M是否停机？该问题对应语言：

$$L_H = \{ C111C' \mid \text{对输入串}C'\text{，图灵机}C\text{将停机}\}$$

“语言”与“判定问题”可以互换。

问题与语言

- 问题的判定 （decision）
  - 若一个问题对应的语言是递归的，则称该问题是可判定的(decidable)，否则是不可判定的(undecidable)；
  - 若一个问题对应的语言是递归可枚举的，则称该问题是部分可判定的(partially decidable)。

问题与语言

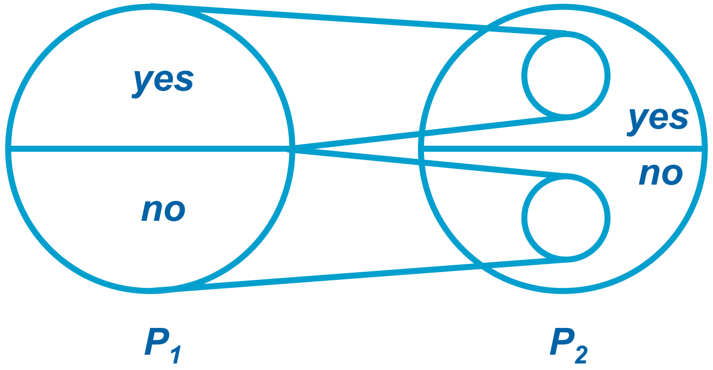
例：

因L_u不是递归的，则如下问题是不可判定的：任给图灵机 M 和输入串w，判定w 是否被 M 接受？

图灵机停机问题也是不可判定的(后面证明)，所对应的语言 L_H不是递归的。

问题的简约

如果有一算法将问题 P_1 的实例 (instances) 转化为问题 P_2 具有相同答案的实例，并且后者能给出前者的答案：



则称问题 P_1 可以简约到 (reduced to) 问题$P_{2.}$

问题的简约

定理：设存在从问题P_1到问题P_2的简约：

a) 若问题P_2是可判定的，则问题P_1也是可判定的；

b) 若问题P_2是部分可判定的，则问题P_1也是部分可判定的。

推论：设存在从问题P_1到问题P_2的简约：

a) 若P_1是不可判定的，则P_2也是不可判定的;

b) 若P_1非递归可枚举，则P_2也是非递归可枚举。

问题的简约

例：

图灵机停机问题：任给图灵机 M 和输入字符串 w，试问对于w，M 是否停机?

- 问题P_1：通用语言L_u
- 问题P_2：停机对应语言L_H
- P_1简约为P_2

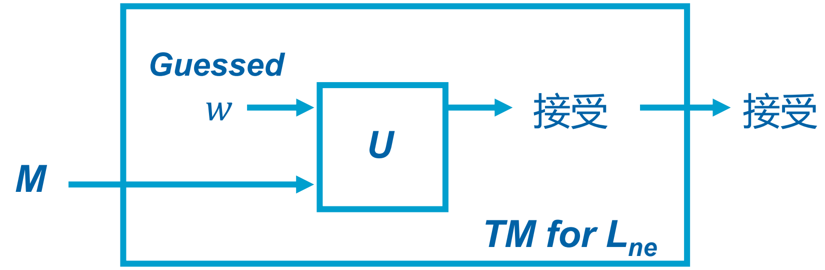
图灵机停机问题是不可判定的。

图灵机的判定问题

- 语言非空图灵机

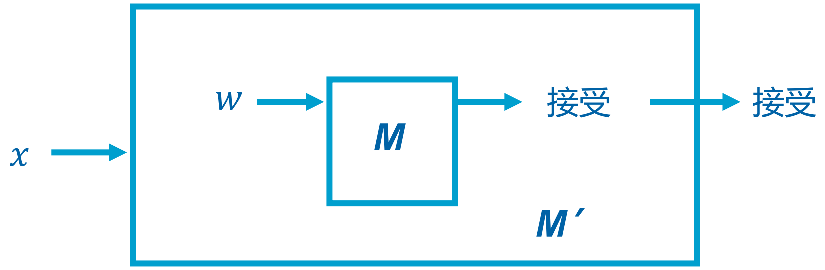
该问题可对应语言：

$$L_{ne}=\{\, M \mid L(M) \neq \varnothing \,\}$$



– L_{ne}可以简约到通用语言 L_u，则该问题是部分可判定的；

– 而 L_u 也可以简约到 L_{ne}，则该问题是不可判定的。



定理: L_{ne} 是递归可枚举语言，但不是递归语言。

图灵机的判定问题

- 语言为空图灵机

该问题可对应语言：

$$L_e = \{ M \mid L(M) = \varnothing \}$$

因为

$$L_e = \overline{L}_{ne}$$

而 L_{ne} 是递归可枚举且不可判定的，所以 L_e 不是递归可枚举的。

定理: L_e 不是递归可枚举语言。

图灵机的判定问题

- 语言的非平凡性质

  设 L 为所有递归可枚举语言的集合。满足性质 P 的递归可枚举语言的集合仍记为 P，即 $P \subseteq L$. 若 P 不等于$\varnothing$或 L，则称 P 为非平凡性质。

定理 (Rice 定理)：

递归可枚举语言的每一个非平凡性质都是不可判定的。

L_{ne}和L_e的不可判定性都是Rice 定理的特例。

有关图灵机的判定问题

例：

直接应用Rice 定理可以得出下列问题是不可判定的：

1. 任给一个语言L，判定 L 是否正则语言？
2. 任给一个语言L，判定 L 是否上下文无关语言？

Post 对应问题

- Post 对应问题（PCP）

PCP的实例包含同一字母表上数量相同的两组字符串：

$$A = w_1, w_2, \ldots, w_k$$

$$B = x_1, x_2, \ldots, x_k$$

(w_i, x_i) $1 \le i \le k$ 称为对应对。

称PCP的该实例有解，当且仅当存在整数序列:

$$i_1, i_2, \ldots, i_m$$

使得 $w_{i_1} w_{i_2} \cdots w_{i_m} = x_{i_1} x_{i_2} \cdots x_{i_m}$

Post 对应问题

Post 对应问题的定义说明：

- 表A和表B中的串的内容可以不同；
- 表A和表B中的项数k必须相同；
- 在表A中取一些项$i_1, i_2, \ldots, i_m$，在表B中取同样个数的对应的项$i_1, i_2, \ldots, i_m$；
- $i_1, i_2, \ldots, i_m$可以重复选取。

Post 对应问题

例：设 Σ={0,1}，两组字符串A，B由下图定义

	A	B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

图(1)

	A	B
i	w_i	x_i
1	10	101
2	011	11
3	101	011

图(2)

a) 图(1) PCP 的实例有解，其中一个解为: 2,1,1,3

即 $w_2w_1w_1w_3 = x_2x_1x_1x_3 =101111110$

b) 图(2) PCP 的实例无解。

Post 对应问题

定理： Post 对应问题是不可判定的。

可以将 L_u 简约到 PCP (参考书)。 从PCP 出发可证明许多其它不可判定问题 。

定理： 任意一个CFG G 是否二义的问题是不可判定的。

Post 对应问题

证明思路：

设PCP的一个实例包含的两组字符串为

$$A=w_1,\ w_2,\ \ldots,\ w_k \quad \text{和} \quad B=x_1,x_2,\ldots,x_k$$

构造CFG G包含如下产生式：

$$S \to A \mid B;$$

$$A \to w_1Aa_1 \mid w_2Aa_2 \mid \ldots \mid w_kAa_k \mid w_1a_1 \mid w_2a_2 \mid \ldots \mid w_ka_k\ ;$$

$$B \to x_1Ba_1 \mid x_2Ba_2 \mid \ldots \mid x_kBa_k \mid x_1a_1 \mid x_2a_2 \mid \ldots \mid x_ka_k\ ;$$

PCP的该实例有解当且仅当 G 是歧义的，即“是否一个给定的CFG是歧义的？”可以归化到PCP问题

不可判定问题

- 图灵机编码
- 对角线语言与通用语言
- 图灵机语言的性质
- 判定问题与语言
- 计算复杂性问题

时间复杂度

- 图灵机的时间复杂度

  如果对任何长为n的输入串w，图灵机 M 可以在最多 $T(n)$ 移动步停机，则称图灵机 M 的时间复杂度为$T(n)$。

- 非确定图灵机的时间复杂度

  若对任何长为n 的输入串 w，非确定图灵机 M 的任何一个转移序列，可以最多 $T(n)$ 个移动步停机，则称非确定图灵机 M 的复杂度为$T(n)$。

$\mathcal{P}$ 问题与 $\mathcal{NP}$ 问题

- $\mathcal{P}$ 问题

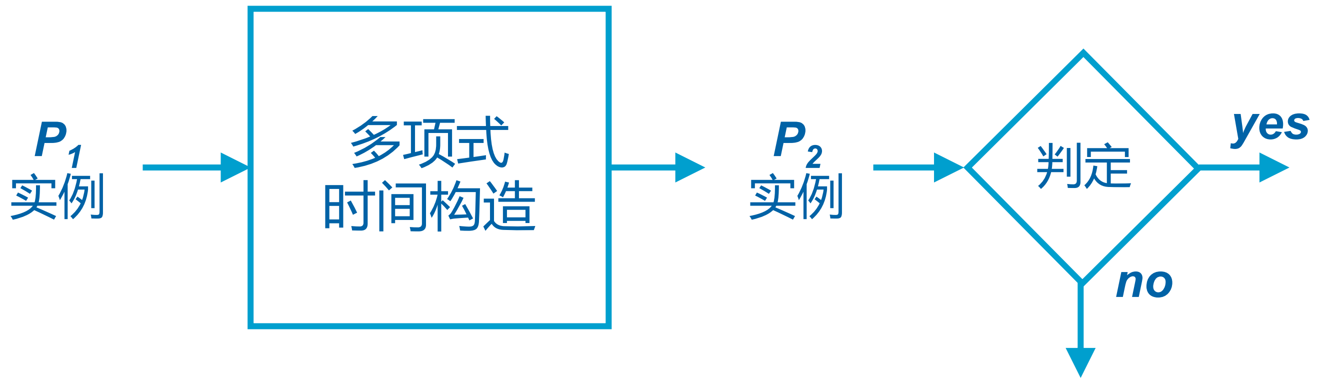
  如果问题（语言）L 满足：存在一确定图灵机 M，使得 $L=L(M)$，且 M 的时间复杂度 $T(n)$ 为多项式，则称该问题是 $\mathcal{P}$ 问题，即 L 属于$\mathcal{P}$.

- $\mathcal{NP}$ 问题

  如果问题（语言）L 满足：存在一个非确定图灵机 M，使得 $L=L(M)$，且 M 的时间复杂度 $T(n)$ 为多项式，则称该问题是 $\mathcal{NP}$ 问题，即 L 属于$\mathcal{NP}$.

$\mathcal{P}$ 问题与 $\mathcal{NP}$ 问题

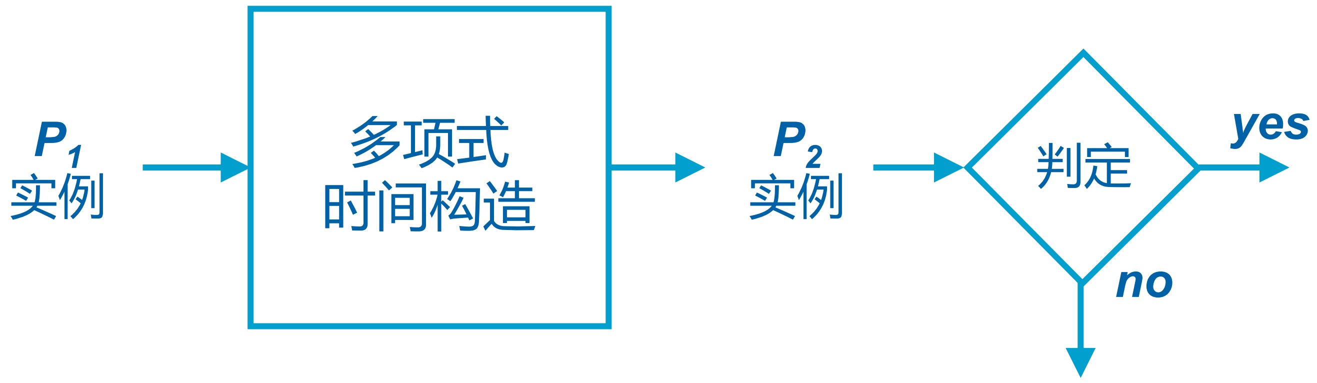
- $\mathcal{P} \subseteq \mathcal{NP}$，如果问题是 $\mathcal{P}$ 问题，则它一定是 $\mathcal{NP}$ 问题。
- $\mathcal{P} = \mathcal{NP}$? 目前仍是一个没有解决的问题。
- 多项式时间简约：



$\mathcal{P}$ 问题与 $\mathcal{NP}$ 问题

若问题 P_1 可以在多项式时间内简约到问题 P_2，则有：

1. 若 P_2 是 $\mathcal{P}$ 问题，则 P_1 也是 $\mathcal{P}$ 问题。
2. 若 P_2 是 $\mathcal{NP}$ 问题，则 P_1 也是 $\mathcal{NP}$ 问题。
3. 若 P_1 不是 $\mathcal{P}$ 问题，则 P_2 也不是 $\mathcal{P}$ 问题。
4. 若 P_1 不是 $\mathcal{NP}$ 问题，则 P_2 也不是 $\mathcal{NP}$ 问题。



$\mathcal{NP}$– 完全问题

$\mathcal{NP}$-完全 ($\mathcal{NP}$–complete)问题

问题 P 如果满足以下条件，则称其为$\mathcal{NP}$-完全问题:

1. P 是 $\mathcal{NP}$ 问题
2. 若 P' 是任一 $\mathcal{NP}$ 问题，则 P' 可以多项式时间简约到 P

定理

设 P_2 是 $\mathcal{NP}$ 问题，而 P_1 是 $\mathcal{NP}$-完全问题. 如果问题 P_1 可以多项式时间简约到 P_2，则 P_2 也是 $\mathcal{NP}$-完全问题

$\mathcal{NP}$– 完全问题

定理

如果可以证明某个 $\mathcal{NP}$–完全问题是 $\mathcal{P}$ 问题，则可以证明

$$\mathcal{P} = \mathcal{NP}$$

$\mathcal{NP}$–难 ($\mathcal{NP}$–hard)问题

如果可以证明问题 $\mathcal{P}$ 满足上述 $\mathcal{NP}$–完全问题的条件 2, 但不能证明条件 1, 则称 $\mathcal{P}$ 是 $\mathcal{NP}$–难问题。

$\mathcal{NP}$ – 完全问题

可满足性 (satisfiability) 问题SAT

布尔表达式的可满足性：
如 $x \wedge \neg(y \vee z)$ 是可满足的，而 $x \wedge \neg x$ 不是可满足的。

SAT 问题是指：
任给一个布尔表达式，它是不是可满足的？

结论 (Cook 定理)

SAT 是$\mathcal{NP}$-完全问题。

$\mathcal{NP}$ – 完全问题

- 从 Sat 可以简约到许多其它的 $\mathcal{NP}$-完全问题，它在计算复杂性理论中的作用可以和可计算性理论中的通用语言（问题）和 Post 对应问题相比拟。
- 本教材介绍了几个其它的 $\mathcal{NP}$-完全问题：
  - CSAT, 3SAT
  - 独立集问题，顶点覆盖问题，有向哈密顿回路问题，无向哈密顿回路问题，旅行商问题等。

课后作业

- 必做题：
  - P-372 Ex.9.1.1 (b)
  - P-373 Ex.9.1.2
  - P-381 Ex.9.2.1, Ex.9.2.3 (b)
  - P-390 Ex.9.3.1
  - P-390 Ex.9.3.3, Ex.9.3.4 (b)

- 思考题:
  - P-372 Ex.9.1.3 (b)
  - P-382 Ex.9.2.6

Thank you