第十三讲

✧ 计算理论初步

计算理论初步

- 对角语言与通用语言
- 问题与语言
- 问题的归约
- 有关图灵机的判定问题
- Post 对应问题与问题的不可判定性
- *P* 问题与*NP* 问题
- *NP* -完全问题与 *NP* -难问题

对角语言与通用语言

- ✧ 图灵机与输入串的二进制编码
- ✧ 对角（*diagonalization*）语言
  - 不是递归可枚举的语言
- ✧ 递归语言和递归可枚举语言的补运算
- ✧ 通用（*universal*）语言
  - 是递归可枚举、但不是递归的语言

对角语言与通用语言

✧ 图灵机与输入串的二进制编码

– 图灵机的编码

对于所关心的问题不失一般性，为方便讨论，先对图灵机作一些假定和简化：

(1) 输入字母表为{0,1}；

(2) 假定有限状态为q_1，q_2，…，q_k，并假定初态总是q_1，终态总是q_2（因已假设图灵机到达接受态总是停机，所以假定一个终态即可）.

(3) 假定带符号为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) 假定带头的移动方向为D_1和D_2，分别代表L和R.

对角语言与通用语言

✧ 图灵机与输入串的二进制编码

– 图灵机的编码（续前页）

在这些假定之后，转移规则 $\delta(q_i, X_j) = (q_k, X_l, D_m)$ 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，形如 $C_1 11 C_2 11 \ldots C_{n-1} 11 C_n$，

– 举例　图灵机 $M = (\{q_1, q_2\}, \{0,1\}, \{0,1,B\}, \delta, q_1, B, \{q_2\})$，转移规则 δ（左）及其编码（右）为

$\delta(q_1, 1) = (q_3, 0, R)$，　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,

该图灵机的编码为

01001000101001100010101001001100010010010100 110001000100010010 .

✧ 图灵机与输入串的二进制编码

– 0，1 字符串的编号

将任意 0，1 字符串 w 用 $1w$ 编号. 如 ε 编号为 1，0 编号为 10，1 编号为 11，00 编号为 100 等.

这样，如果一个图灵机的二进制编码为 w_i，而 w_i 为第 i 个 0，1 字符串，就把该图灵机称为第 i 个图灵机.

这里，任何一个输入串 w，可以对应到某个整数编号 j，称之为第 j 个字符串.

– 图灵机与输入串偶对的编码　在通用语言的定义中，将会用到图灵机与输入串偶对 (M,w) 的编码. 设 M 的二进制编码为 C，则 (M,w) 的二进制编码为

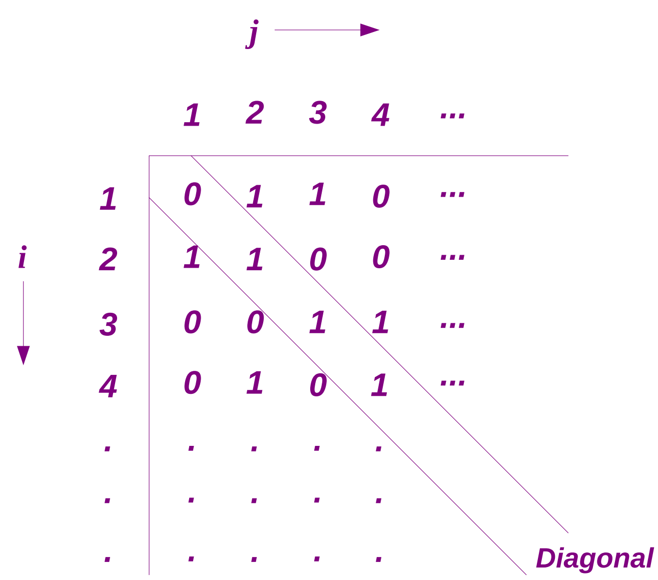
$C\,111\,w$.

对角语言与通用语言

✧ 对角语言

- **定义** 按照上述编码方法，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. 然而，不是每个整数 j 都能对应一个图灵机（即第 j 个 0, 1 字符串不对应任何图灵机的编码），此时不妨认为第j个图灵机为不接受任何字符串的图灵机，即 $L(M_j)=\phi$. 这样，就可以规定对任何 $i\geq 1$，第 i 个图灵机为 M_i. 定义对角语言为

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



- **结论** 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 $M=M_k$，那么对于第 k 个 0, 1 字符串 w_k，试问：是否有 $w_k \in L_d$？这是一个悖论. 因此，不存在这样的 M.

对角语言与通用语言

✧ 递归语言和递归可枚举语言的补运算

– 结论

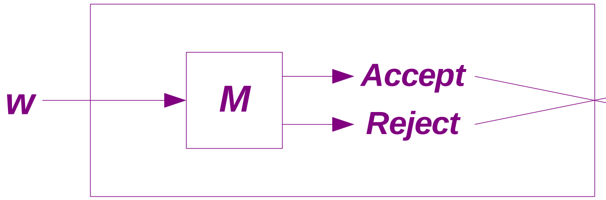
作用于递归语言的补运算是封闭的. 即，如果 L 是递归语言，则 $\overline{L}$ 也是递归语言.

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 $\overline{M}$（参见右下图）：

1. 将 M 的终态作为 $\overline{M}$ 的非终态，且 $\overline{M}$ 在这些状态下没有下一步转移.
2. 增加一个新的终态 r，且 $\overline{M}$ 在状态 r 下没有进一步的转移.
3. 对每一非终态 q，以及每一带符X，若 $\delta(q, X)$ 无定义，则增加转移

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

其中 Y 和 D 可任取.



显然，$\overline{L} = L(\overline{M})$.

对角语言与通用语言

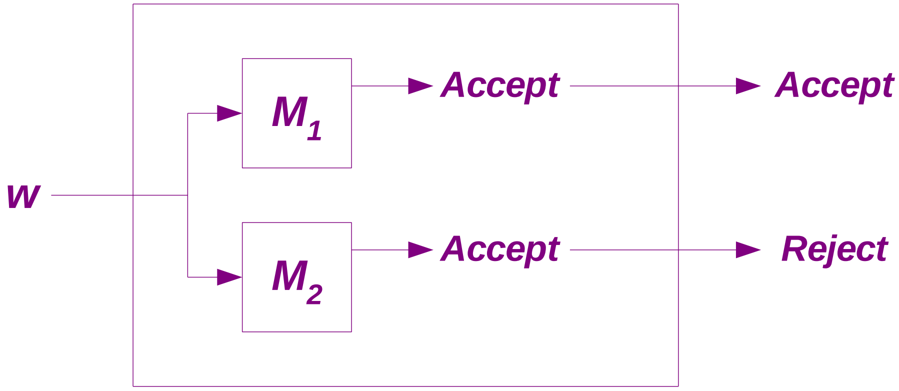
✧ 递归语言和递归可枚举语言的补运算

– **结论** 递归可枚举语言的补运算不是封闭的

即将看到的通用语言 L_u 是递归可枚举语言，但 $\overline{L_u}$ 不是递归可枚举的.

– **结论** 如果语言 L 及 $\overline{L}$ 都是递归可枚举的，则语言 L 及 $\overline{L}$ 都一定是递归的.

证明思路 如右图所示，设 $L = L(M_1)$，$\overline{L} = L(M_2)$，构造图灵机 M 来模拟 M_1 和 M_2 的并行执行. 无论输入串 w 是否属于 L，M 总是能够停机.



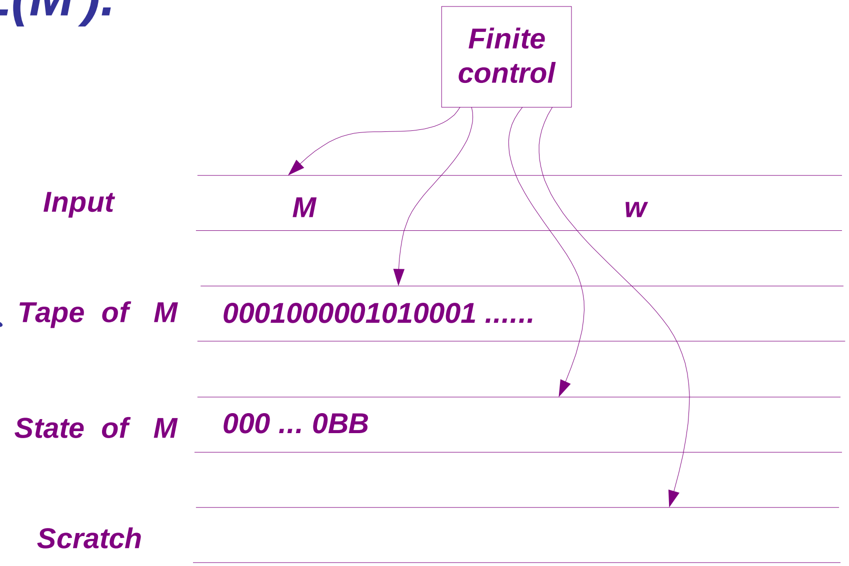
对角语言与通用语言

✧ 通用语言

– **回顾** 设 M 为接受二进制输入串的图灵机， w 为$\{0,1\}^*$ 中的串， M 的二进制编码为 C，则 (M, w) 的二进制编码为 $C\,111\,w$.

– **通用语言** 用于编码 (M,w) 的所有 0，1 字符串的集合，记为 L_u. 其中，(M,w) 满足 $w \in L(M)$.

– **通用图灵机** 可以构造一个图灵机 U，使得$L_u = L(U)$，U 可以是如右图所示的多带图灵机（细节略）. 对于偶对 (M,w)， $w \in L(M)$， 当且仅当 U 接受 (M,w)（编码形式）. 称这样的 U 为通用图灵机.

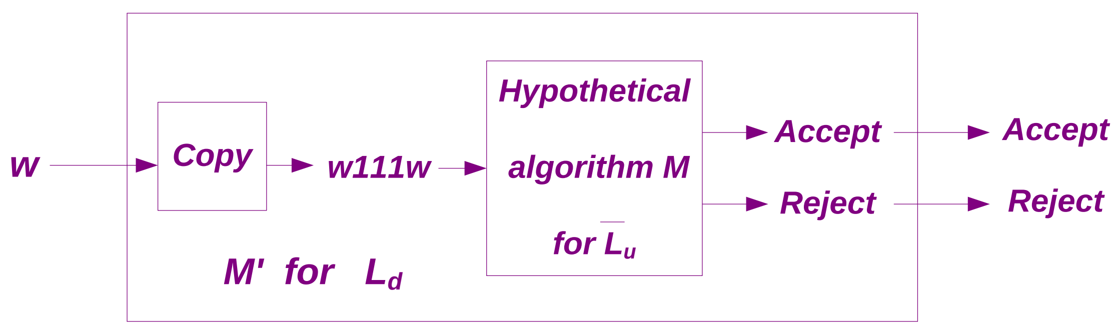


对角语言与通用语言

✧ 通用语言

– **结论** 通用语言 L_u 为递归可枚举的，但不是递归的.

证明思路 已经看到存在通用图灵机 U 满足 $L_u = L(U)$，所以 L_u 是递归可枚举语言. 另一方面，用反证法可以说明 L_u 不是递归的. 否则，$\overline{L_u}$ 也是递归的. 这样，可以得出对角语言 L_d 也是递归语言的结果，但 L_d 甚至不是递归可枚举的. 假定 $\overline{L_u} = L(M)$ ，可以构造图灵机 M'（参见下图），使得 $L_d = L(M')$.



– **推论** 通用语言 L_u 的补不是递归可枚举的.

问题与语言

✧ 回顾　设 $L \subseteq \Sigma^*$ 是字母表Σ上的一个语言，则与L对应的问题（*problem*）定义为：

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

观点　"语言"与"问题"本质上可以互换使用。

理解　二者关系类似于"集合"与"谓词"之间的关系.

✧ 举例—语言对应问题　通用语言 L_u 对应的问题为:

- 任给图灵机 *M* 和输入串 *w*，判定 *w* 是否被 *M* 接受?

✧ 举例—问题对应语言　图灵机停机问题：任给图灵机 *M*，以及输入字符串 *w*，试问对于 *w*，*M* 是否停机（*halts*）？该问题对应语言

$$L_H = \{\, C\,111\,C' \mid \text{对于输入串}C'\text{，图灵机 } C \text{ 将停机} \,\}\ .$$

问题与语言

✧ **问题的判定**（*decision*） 如果一个问题所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*）. 顺便，如果一个问题所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*），否则是**非部分可判定的**。

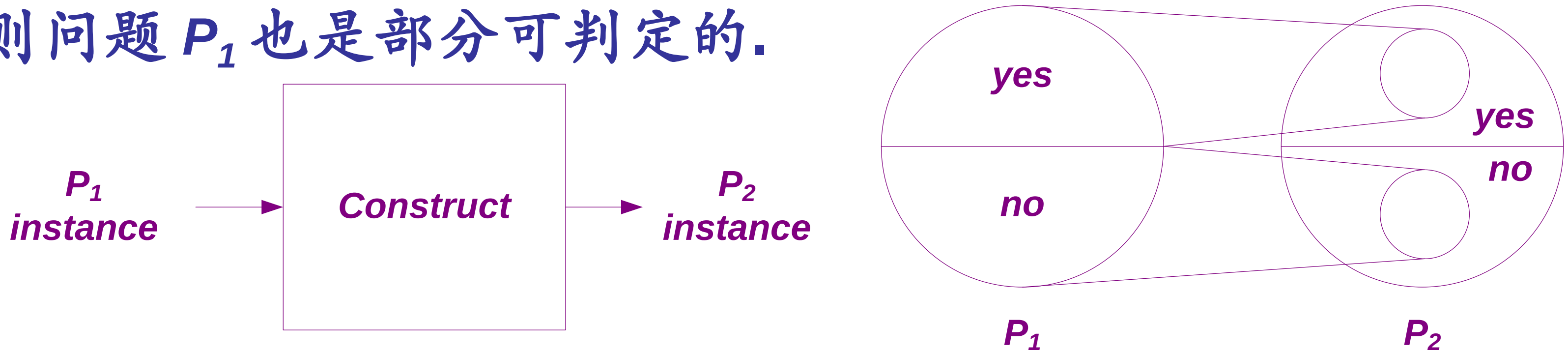
✧ **举例** 因为L_u不是递归的，所以如下问题是不可判定的：

- 任给图灵机 *M* 和输入串*w*，判定*w* 是否被 *M* 接受?

✧ **举例** 随后将证明，图灵机停机问题也是不可判定的，同时所对应的语言 L_H 不是递归的.

✧ 问题的归约

– 一个问题归约到另一个问题　如果可以找到一个算法可以将问题 P_1 的实例（*instances*）转化为问题P_2的实例，并且对于后者作出的回答与前者相同，则称问题 P_1 可以归约到（*reduced to*）问题P_2． 参见下图，如果问题 P_2 是可判定的，则问题P_1也是可判定的；如果问题P_2是部分可判定的，则问题 P_1 也是部分可判定的.



逆否命题 如果 P_1 不是递归的（可判定的），则 P_2 也不是递归的（可判定的）；如果 P_1 不是递归可枚举的（部分可判定的），则问题 P_2不是递归可枚举的（部分可判定的）．

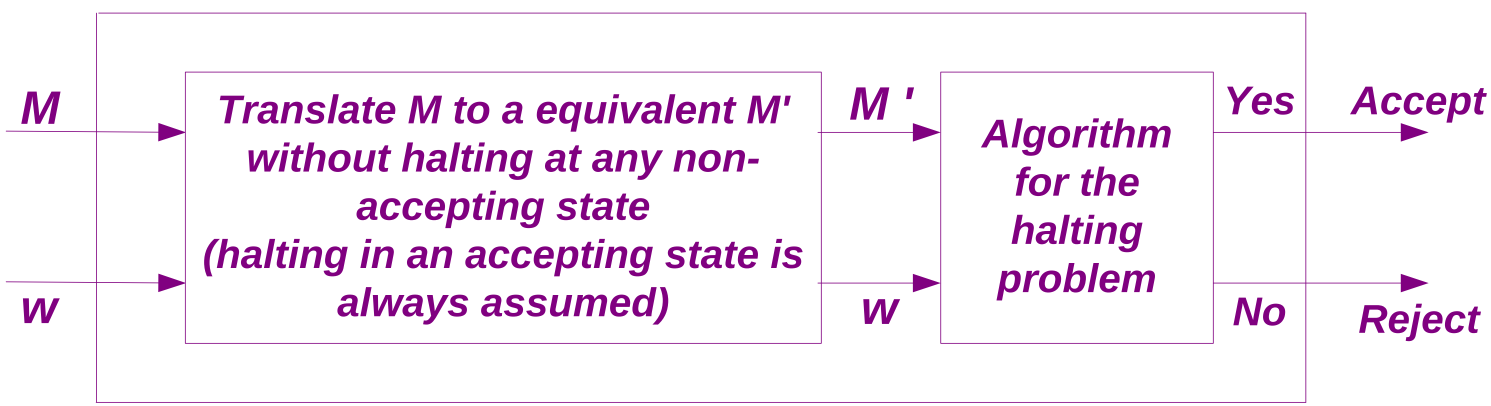
问题的归约

✧ 问题的归约

– 举例

图灵机停机问题：任给图灵机 *M*，以及输入字符串 *w*，试问对于*w*，*M* 是否停机（***halts***）？

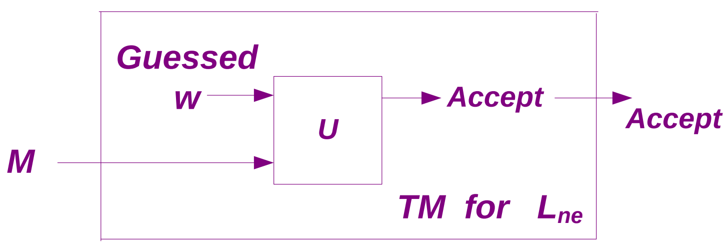
因为通用语言 L_u 可以归约到图灵机停机问题（参见下图），所以图灵机停机问题是不可判定的，其对应的语言 L_H 不是递归语言.

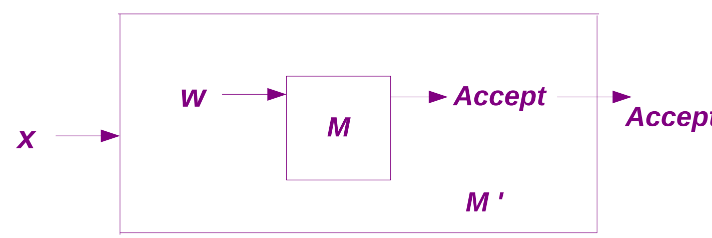


有关图灵机的判定问题

✧ 判定图灵机的语言是否非空

该问题可对应语言 $L_{ne}=\{ M \mid L(M) \neq \phi \}$，可以归约到通用语言 L_u（参见左下图），所以 L_{ne} 是递归可枚举的，即该问题是部分可判定的. 而 L_u 也可以归约到 L_{ne}（参见右下图），所以 L_{ne} 不是递归的，即该问题是不可判定的.





✧ 判定图灵机的语言是否为空

该问题可对应语言 $L_e=\{ M \mid L(M) = \phi \}$，因为 $L_e=\overline{L_{ne}}$，而 L_{ne} 是递归可枚举的但是不可判定的，所以 L_e 不是递归可枚举的.

有关图灵机的判定问题

✧ **Rice 定理** 有关递归可枚举语言的任何非平凡性质都是不可判定的.

设 *L* 为所有递归可枚举语言的集合，关于递归可枚举语言的性质（*property*）可表达为 $P \subseteq L$. 若 *P* 不等于 Φ 或 *L*，则 *P* 为非平凡性质.

前述的 L_{ne} 和 L_e 的不可判定性都是 *Rice* 定理的特例

✧ **举例** 直接应用 *Rice* 定理可以得出下列问题是不可判定的：

1. 任给图灵机可以接受的语言 *L*，问 *L* 是否正规语言？
2. 任给图灵机可以接受的语言 *L*，问 *L* 是否上下文无关语言？

Post 对应问题与问题的不可判定性

✧ **Post 对应问题**　Post 对应问题（Post's Corresponding Problem），简称PCP.　PCP的一个实例包含同一字母表上的两组字符串，$A=w_1,w_2,\ldots,w_k$，$B=x_1,x_2,\ldots,x_k$；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

$$w_{i_1}w_{i_2}\ldots w_{i_m} = x_{i_1}x_{i_2}\ldots x_{i_m}.$$

✧ **举例**

设 $\Sigma=\{0,1\}$，两组字符串A，B由右图定义.　PCP 的该实例有解，其中一个解为整数序列 2,1,1,3，即

$$w_2w_1w_1w_3 = x_2x_1x_1x_3 =101111110.$$

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

Post 对应问题与问题的不可判定性

✧ **结论** *Post* 对应问题是不可判定的.

可以将 L_u 归约到 *PCP* (参考课本)来证明这一结论.

从*PCP* 出发可以证明许多其它的不可判定问题.

✧ **举例** 问题“是否一个给定的*CFG*是歧义的？”是不可判定的.

证明思路 设*PCP*的一个实例包含的两组字符串为$A=w_1, w_2, \ldots, w_k$ 和 $B=x_1, x_2, \ldots, x_k$；构造*CFG* *G*包含如下产生式:

$$S \rightarrow A \mid B;$$

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

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

*PCP*的该实例有解当且仅当 *G* 是歧义的.

即*PCP*可以归约到问题“是否一个给定的*CFG*是歧义的？”

P问题与NP问题

- ✧ 图灵机的时间复杂度（*time complexity*）如果对于任何长为 *n* 的输入串 *w*，图灵机 *M* 可以在最多 *T(n)* 个移动步停机（无论是否有 $w \in L(M)$），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 如果对于任何长为 *n* 的输入串 *w*，非确定图灵机 *M* 的任何一个转移序列可以在最多 *T(n)* 个移动步停机（无论是否有 $w \in L(M)$），则称非确定图灵机 *M* 的时间复杂度为 *T(n)*.
- ✧ 问题（语言）类*P* 如果问题（语言）*L* 满足：存在一个图灵机 *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 *P* 问题，即 *L* 属于*P*.
- ✧ 问题（语言）类*NP* 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 *NP* 问题，即 *L* 属于*NP*.

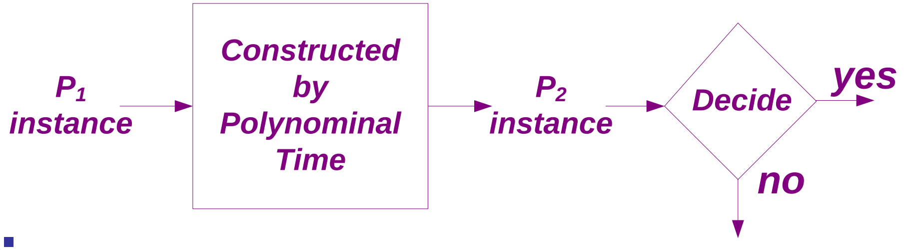
P 问题与 NP 问题

✧ $P \subseteq NP$ 如果一个问题是 P 问题，则它一定是 NP 问题.

✧ P = NP ? 目前仍是一个没有解决的开放问题.

✧ 多项式时间归约 如果问题 P_1 可以在多项式时间内归约到问题 P_2（参见右下图），有

1. 若 P_2 是 P 问题，则 P_1 也是 P 问题.
2. 若 P_2 是 NP 问题，则 P_1 也是 NP 问题.
3. 若 P_1 不是 P 问题，则 P_2 也不是 P 问题.
4. 若 P_1 不是 NP 问题，则 P_2 也不是 NP 问题.



NP-完全问题与 NP-难问题

- ✧ **NP-完全（NP-complete）问题**

  问题 P 如果满足以下条件，则称其为NP-完全问题：

  1. P 是 NP 问题.

  2. 若 P' 是任一 NP 问题，则 P' 可以多项式时间归约到 P.

- ✧ **结论** 设 P_2 是 NP 问题，而 P_1 是 NP-完全问题. 如果问题 P_1 可以多项式时间归约到 P_2，则 P_2 也是 NP-完全问题.

- ✧ **结论** 如果可以证明某个 NP-完全问题是 P 问题，则可以证明 $P = NP$.

- ✧ **NP–难（NP-hard）问题**

  如果可以证明问题 P 满足上述 NP-完全问题的条件 2，但不能证明条件 1，则称 P 是 NP-难问题.

NP-完全问题与 NP-难问题

✧ 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*NP*-完全问题.

从 *Sat* 可以归约到许多其它的 *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.

本课教材介绍了几个其它的 *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.

Wish You a Great Success,

Thank You