# 背景知识

岳镝

2025 年 2 月 21 日

# 算法设计与分析

学什么？

TCS

- 算法设计
  (1) 各种经典算法：分治、动态规划、贪心、回溯、线性规划、网络流…
  (4) 现代算法设计技术：近似算法、随机算法…
  α-apx　high-prob
- 算法分析
  (2) 时间复杂度、空间复杂度、均摊分析…
- 一点基础的复杂性理论
  (3) 问题的计算复杂度、难解性…

Upper bound
$O(\cdot)$

Lower bound
$\Omega(\cdot)$

Algorithm

Upper bound

gap

Lower bound

Complexity

# 算法设计与分析

学什么？

- 算法设计
  (1) 各种经典算法：分治、动态规划、贪心、回溯、线性规划、网络流...
  (4) 现代算法设计技术：近似算法、随机算法...
- 算法分析
  (2) 时间复杂度、空间复杂度、均摊分析...
- 一点基础的复杂性理论
  (3) 问题的计算复杂度、难解性...
- ## 什么是算法???

# 算法的严格定义

为什么要定义算法？

# 希尔伯特第 10 问题

## Hilbert's tenth problem (1900)

Given a diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: to devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.

$$P(x) = x^{10} + 9x^8 - 15x^3 + x - 51$$

$$f(x, y, z) = x^9 y^2 z - 61 y z^6 + 38 z - 125$$

$$f(x, y, z) = 0 \leftarrow \exists\, x, y, z \in \mathbb{Z}$$

# 希尔伯特第 10 问题

## Hilbert's tenth problem (1900)

Given a diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: to devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.

- 答案 (Matiyasevich, 1970)：不存在这样的 process。
- 怎么证明？

# 希尔伯特第 10 问题

## Hilbert's tenth problem (1900)

Given a diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: to devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.

- 答案 (Matiyasevich, 1970)：不存在这样的 process。
- 怎么证明？
- 怎么定义
  (1) 算法
  (2) 问题
  (3) 算法 $\mathcal{A}$ 解决问题 $\mathcal{P}$.

# 问题

算法用来解决问题。那么，什么是一个问题？

# 问题

算法用来解决问题。那么，什么是一个问题？

- 问题 1：你今天早上吃包子了吗？

# 问题

算法用来解决问题。那么，什么是一个问题？

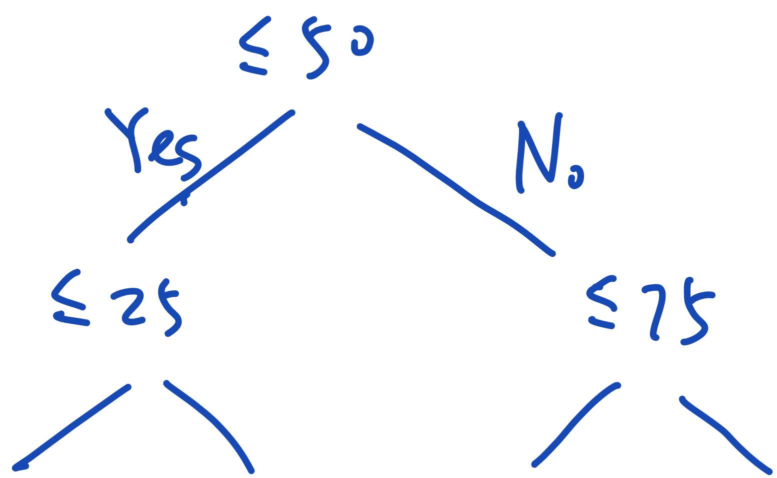
- 问题 1：你今天早上吃包子了吗？
- 问题 2：你今天早上吃了几个包子？

# 问题

算法用来解决问题。那么，什么是一个问题？

- 问题 1：你今天早上吃包子了吗？
- 问题 2：你今天早上吃了几个包子？

像问题 1 这样，只需要回答“是”或者“否”的问题被称为判定性问题。



# 语言和判定性问题

- 只需要回答“是”或者“否”的问题被称为*判定性问题*。
- 用 $\boxed{\{0,1\}^*}$ 表示所有有限长度的二进制串组成的集合
- 子集 $\boxed{L} \subseteq \{0,1\}^*$ 称为一个语言
- $L$ 对应的判定性问题：输入 $x \in \{0,1\}^*$，问 $x \in L$?

000
0101
0010101

# 语言和判定性问题

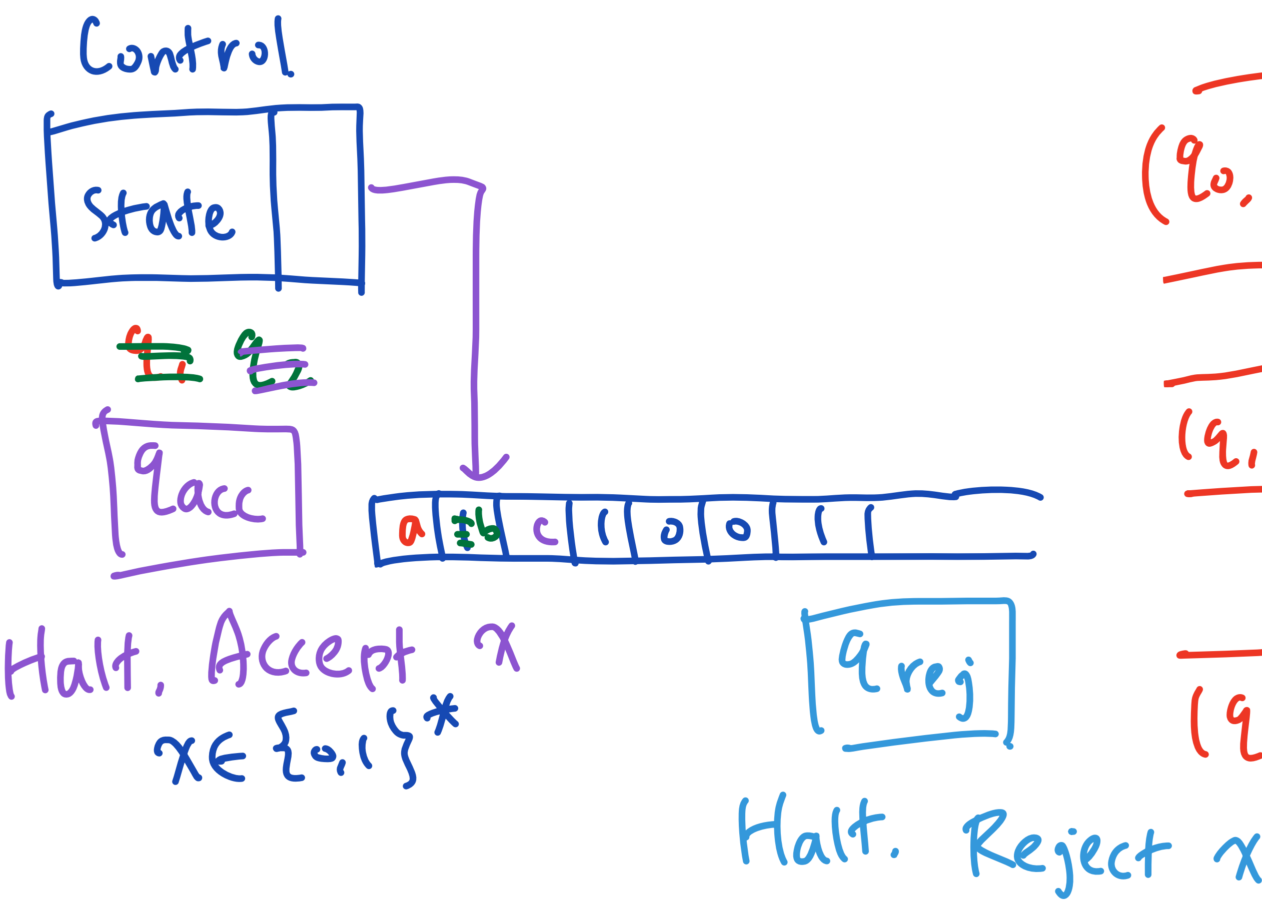
- 只需要回答“是”或者“否”的问题被称为判定性问题。
- 用 $\{0,1\}^*$ 表示所有有限长度的二进制串组成的集合
- 子集 $L \subseteq \{0,1\}^*$ 称为一个语言
- $L$ 对应的判定性问题：输入 $x \in \{0,1\}^*$，问 $x \in L$?
- 希尔伯特第 10 问题
  - 语言：$L_1 = \{p$ 存在$x_1, x_2, \cdots \in \mathbb{Z}$, 使得$p(x_1, x_2, \dots) = 0\}$
  - 问题：$x^3y + y^2z - 19 \in L_1$?
- 你今天早上吃包子了吗？
  - 语言：$L_2 = \{x$: $x$ 今天早上吃了包子$\}$
  - 问题：你 $\in L_2$?

$L_3 = \{y$: 你今天早上吃了 $y\}$

包子 $\in L_3$?

# 图灵机和算法

- 问题：输入 $x \in \{0,1\}^*$，问 $x \in L$?
- 在不同的计算模型下，问题的可解性/复杂性可能是不同的
- 本课程默认采用图灵机模型




| | State | Write | Move |
|---|---|---|---|
| $(q_0, 0)$ | $q_1$ | a | 右 |
| | | | |
| $(q_1, 1)$ | $q_2$ | b | 右 |
| | | | |
| $(q_2, 0)$ | $q_{acc}$ | c | 左 |

# 图灵机和算法

- 问题：输入 $x \in \{0,1\}^*$，问 $x \in L$?
- 在不同的计算模型下，问题的可解性/复杂性可能是不同的
- 本课程默认采用图灵机模型

图灵机是一个 7 元组 $(Q, \Sigma, \Gamma, \delta, q_0, q_{\mathrm{acc}}, q_{\mathrm{rej}})$

(1) $Q$ 是状态集合

(2) $\Sigma$ 是输入字母表，即 $\Sigma = \{0, 1\}$

(3) $\Gamma$ 是纸带字母表，包含空白字符。$\Sigma \subseteq \Gamma$

(4) $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}\}$ 称为转移函数

(5) $q_0 \in Q$ 称为初始状态

(6) $q_{\mathrm{acc}} \in Q$ 称为接受状态

(7) $q_{\mathrm{rej}} \in Q$ 称为拒绝状态

# 图灵机和算法

对于图灵机 $\mathcal{M}$ 和输入 $x \in \{0,1\}^*$

- 若 $\mathcal{M}$ 在 $x$ 上停机，且停机时处于接受状态 $q_{\mathrm{acc}}$，则称 $\mathcal{M}$ 接受 $x$；
- 若 $\mathcal{M}$ 在 $x$ 上停机，且停机时处于拒绝状态 $q_{\mathrm{rej}}$，则称 $\mathcal{M}$ 拒绝 $x$；
- 注意：$\mathcal{M}$ 在 $x$ 上可能不停机

# 图灵机和算法

对于图灵机 $\mathcal{M}$ 和输入 $x \in \{0,1\}^*$

- 若 $\mathcal{M}$ 在 $x$ 上停机，且停机时处于接受状态 $q_{\text{acc}}$，则称 $\mathcal{M}$ 接受 $x$；
- 若 $\mathcal{M}$ 在 $x$ 上停机，且停机时处于拒绝状态 $q_{\text{rej}}$，则称 $\mathcal{M}$ 拒绝 $x$；
- 注意：$\mathcal{M}$ 在 $x$ 上可能不停机
- 称语言 $L(\mathcal{M}) := \{x \in \{0,1\}^* : \mathcal{M} \text{ 接受} x\}$ 为图灵机 $\mathcal{M}$ 识别的语言。
- 反之，称语言 $L$ 为 *图灵可识别*，若存在图灵机 $\mathcal{M}$ 识别 $L$

# 图灵机和算法

对于图灵机 $\mathcal{M}$ 和输入 $x \in \{0,1\}^*$

- 若 $\mathcal{M}$ 在 $x$ 上停机，且停机时处于接受状态 $q_{\mathrm{acc}}$，则称 $\mathcal{M}$ 接受 $x$；
- 若 $\mathcal{M}$ 在 $x$ 上停机，且停机时处于拒绝状态 $q_{\mathrm{rej}}$，则称 $\mathcal{M}$ 拒绝 $x$；
- 注意：$\mathcal{M}$ 在 $x$ 上可能不停机
- 称语言 $L(\mathcal{M}) := \{x \in \{0,1\}^* : \mathcal{M} \text{ 接受} x\}$ 为图灵机 $\mathcal{M}$ 识别的语言。
- 反之，称语言 $L$ 为 *图灵可识别*，若存在图灵机 $\mathcal{M}$ 识别 $L$

练习：希尔伯特第 10 问题是图灵可识别的吗？

# 图灵机和算法

图灵可识别

- 称语言 $L(\mathcal{M}) := \{x \in \{0,1\}^* : \mathcal{M} \text{ 接受} x\}$ 为图灵机 $\mathcal{M}$ 识别的语言。
- 反之，称语言 $L$ 为 *图灵可识别*，若存在图灵机 $\mathcal{M}$ 识别 $L$

# 图灵机和算法

图灵可识别

- 称语言 $L(\mathcal{M}) := \{x \in \{0,1\}^* : \mathcal{M}\ 接受x\}$ 为图灵机 $\mathcal{M}$ 识别的语言。
- 反之，称语言 $L$ 为 *图灵可识别*，若存在图灵机 $\mathcal{M}$ 识别 $L$

图灵可判定

- 图灵可判定 $=$ 图灵可识别 $+$ 停机
- 称语言 $L$ 为 *图灵可判定*，若存在图灵机 $\mathcal{M}$ 识别 $L$，且 $\mathcal{M}$ 在任意输入上停机

$\forall x \in L.\quad M(x) = acc$

$\forall x \notin L.\quad M(x) = rej$

# 图灵机和算法

图灵可识别

- 称语言 $L(\mathcal{M}) := \{x \in \{0,1\}^* : \mathcal{M}\text{ 接受}x\}$ 为图灵机 $\mathcal{M}$ 识别的语言。
- 反之，称语言 $L$ 为 *图灵可识别*，若存在图灵机 $\mathcal{M}$ 识别 $L$

图灵可判定

- 图灵可判定 $=$ 图灵可识别 $+$ 停机
- 称语言 $L$ 为 *图灵可判定*，若存在图灵机 $\mathcal{M}$ 识别 $L$，且 $\mathcal{M}$ 在任意输入上停机

**定理 (Matiyasevich, 1970)**

希尔伯特第 10 问题不是图灵可判定的

# 图灵机和算法

## Church-Turing Thesis

Turing decidable

Every effectively calculable function can be $\boxed{\text{computed}}$ by a Turing machine.

Intuive

λ- calculus

Turing Machine

Church

Turing

# 图灵机和算法

### Church-Turing Thesis

Every effectively calculable function can be computed by a Turing machine.

- 算法 $\Longleftrightarrow$ 图灵机
- 算法 $\boxed{\mathcal{A}}$ 计算问题 $\mathcal{P}$ $\Longleftrightarrow$ 图灵机 $\boxed{\mathcal{M}}$ 判定语言 $\boxed{L}$.

# 图灵机和算法

### Church-Turing Thesis

Every effectively calculable function can be computed by a Turing machine.

- 算法 $\Longleftrightarrow$ 图灵机
- 算法 $\mathcal{A}$ 计算问题 $\mathcal{P}$ $\Longleftrightarrow$ 图灵机 $\mathcal{M}$ 判定语言 $L$.

### Hilbert's tenth problem (1900)

Given a diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: to devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.

Algorithm

Turing decidable