逻辑学

杨睿之

复旦大学哲学学院

2023 年秋季

前情提要

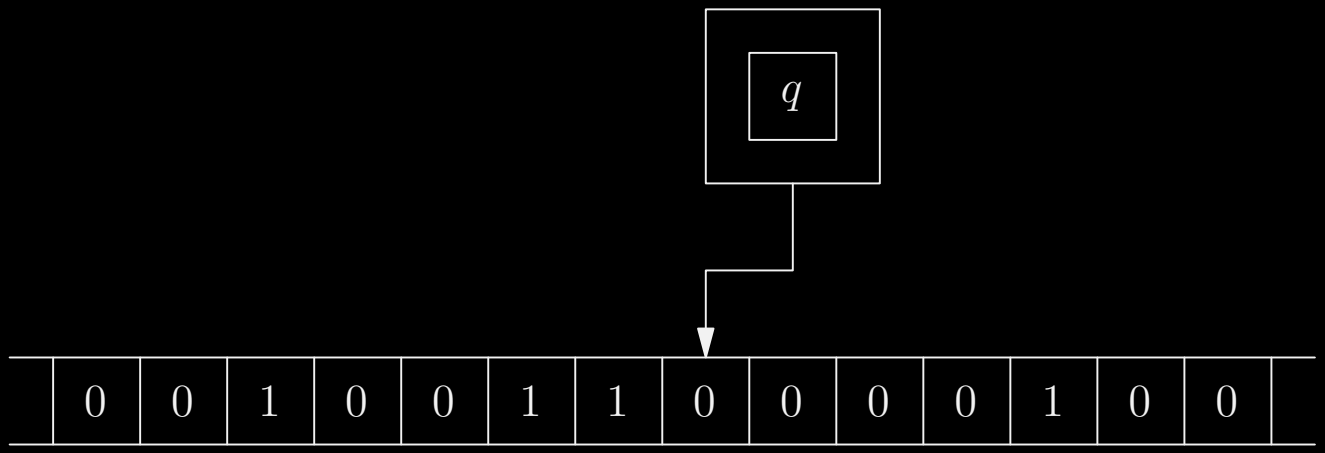
- 谓词逻辑的可靠性与完备性：

$$\Sigma \vDash \varphi \Leftrightarrow \Sigma \vdash \varphi$$

- 图灵对机械可计算直观的刻画——图灵机可计算
- 图灵关于图灵机可计算正确刻画可计算直观的论证

图灵机可计算

图灵机：



图灵机可计算

一个图灵机有：一个读写头，一个双向无穷纸带和

- 有穷 字母表 （alphabet）$\mathcal{A} = \{a_1, \ldots, a_n\}$
- 有穷 内部状态 （state）集 $\mathcal{Q} = \{q_s, q_h, q_1, \ldots, q_m\}$

  我们一般设置两个特殊的状态，q_s 是开始（start）状态，q_h 是停机（halting）状态
- 有穷 指令 （instruction）集 $\mathcal{I}$

图灵机可计算

图灵机的指令形如：

- p_1aRp_2：表示图灵机如果在内部状态 p_1 读写头读到字母 a 时，读写头向右移动一格并变更状态为 p_2
- p_1aLp_2：类似，读写头向左移动一格
- $p_1a_1a_2p_2$：表示图灵机在内部状态 p_1 读写头读到字母 a_1 时，读写头把该格字母改写为 a_2 并进入 p_2 状态

图灵机可计算

- 我们可以假设图灵机的字母表只有 $\{0, 1\}$（或只有 1 和“空”），这不会导致“图灵机可计算”概念变弱
- 我们可以假设纸带在任何时候只有有穷个格子不是空的。这样，图灵机在任何一个时刻的 状况 （configuration）可以用一个有穷字符串来表示。例如：

$$10011q011$$

图灵机可计算

我们可以把一台图灵机设想为一个自然数上的 k 元 部分函数 （partial function）Φ，它把一些自然数的 k 元数 $(x_1, \ldots, x_k)$ 组映射为自然数 $\Phi(x_1, \ldots, x_k)$，但不一定在任何输入下都有定义。

- 我们说 输入 $(x_1, \ldots, x_k)$ 时，表示图灵机处于“初始状况”：$q_s[(x_1+1)\text{个}1]0\ldots0[(x_k+1)\text{个}1]$
- 当图灵机内部状态为 q_h 时，我们称它 停机 。停机时读写头位置开始向右连续 1 的个数为它的 输出

图灵机可计算

此外，还我们规定指令集满足

- 不存在以 q_h 开头的指令（说停机就真停机了）
- 对人任何状态和字母的组合 qa 至多存在一个以它们开头的指令。即这是一台 确定的图灵机 （deterministic Turing machine）

图灵机可计算

例

我们可以用图灵机来计算加法：考虑指令集

- $q_s 10 q_1, q_1 0 R q_2$ 删去首个 1 并向右移一格
- $q_2 1 R q_2, q_2 01 q_3$ 向右移到第一个 0，改成 1
- $q_3 1 L q_3, q_3 0 R q_4$ 向左回到第一个 1
- $q_4 10 q_5, q50 R q6, q610 q7, q70 R qh$ 删去开头两个 1

图灵机可计算

- 今天主流的程序语言都是 图灵完全的 。例如，一个部分函数是图灵可计算的，当且仅当存在一个 python 程序来算它
- 正如一个计算机程序就是一个文本文件，可以被编码为一个自然数。我们也可以把图灵机乃至图灵机可计算的函数编码为自然数：$\Phi_0, \Phi_1, \ldots$
- 存在 通用图灵机 （universal Turing machine）U：

  $U(e, n) = \Phi_e(n)$

图灵机可计算

定理

停机问题（halting problem）不是可计算的（或可判定的），不存在图灵机可计算的函数 Ψ，使得对任意自然数 e, n，

$$\Psi(e,n) = \begin{cases} 0 & \text{如果 } \Phi_e(n)\uparrow \text{（不停机）} \\ 1 & \text{如果 } \Phi_e(n)\downarrow \text{（停机）} \end{cases}$$

证明.

反设有 Ψ。任取可计算的 2 元全函数 f。证明 $f \neq \Psi$。考虑程序 Φ_d：输入 e，若 $f(e,e) = 0$，输出 0；否则不停机。考虑 $\Phi_d(d)$。

谓词逻辑不是可判定的

请回忆：

- $\varphi_1, \ldots, \varphi_n \vDash \psi$，当且仅当 $\vDash \varphi_1 \to \cdots \to \varphi_n \to \psi$，当且仅当 $\{\varphi_1, \ldots, \varphi_n, \neg\psi\}$ 不可满足
- 一个涉及有穷个公式的推理是否有效、一个公式是否是有效式、一个有穷公式集是否是可满足的，这三类问题可以互相转化

谓词逻辑不是可判定的

定理

一个有穷的谓词逻辑公式集 $\{\varphi_1, \ldots, \varphi_n\}$ 是否是可满足的，不是能行可判定的（可计算的或图灵可计算的）。

回忆：如果只考虑命题逻辑或只含有一元谓词符号的谓词逻辑，这类问题是可计算的

谓词逻辑不是可判定的

证明思路：我们证明，如果这类问题是可计算的，那么停机问题也是可计算的。

具体来说，我们设计一个能行的变换 f，输入一个图灵机 M 及其输入 n，输出一个有穷的谓词逻辑语句集 $f(M,n)$ 使得，

M 在输入 n 下不停机当且仅当 $f(M,n)$ 可满足

谓词逻辑不是可判定的

证明思路：我们证明，如果这类问题是可计算的，那么停机问题也是可计算的。

具体来说，我们设计一个能行的变换 f，输入一个图灵机 M 及其输入 n，输出一个有穷的谓词逻辑语句集 $f(M,n)$ 使得，

M 在输入 n 下不停机当且仅当 $f(M,n)$ 可满足

谓词逻辑不是可判定的

引入函数符号

- 用关系表示函数。如果一个二元关系 R 满足 $\forall x \exists y (Rxy \wedge \forall y_1 \forall y_2 (Rxy_1 \to Rxy_2 \to y_1 = y_2))$ 对每个 x 存在唯一的 y 与 x 有 R 关系)，我们称 R 表示了一个 函数 。一搬又把这个公式简写为 $\forall x \exists! y Rxy$
- 如果声明了 $\forall \exists! y Rxy$，我们可以引入一个函数符号 f 指代 R 表示的函数。此时，我们可以用 $Pf(x)$ 作为 $\exists y (Rxy \wedge Py)$ 的缩写

谓词逻辑不是可判定的

- 我们希望用谓词逻辑的语言描述图灵机完整的运行过程
- 我们把图灵机的纸带每个时刻的状态从上到下排下来，这形成了一个“下半平面”
- 我们心目中的 模型 的论域是这个平面中的格子

谓词逻辑不是可判定的

描述图灵机运行过程的语言

- 我们需要一元谓词符号 0 和 1 表示一个格子里的符号是 0（空）还是 1
- 我们需要一个一元谓词符号 F 表示这些格子属于起始的第一行
- 我们用一元谓词符号 $Q_s, Q_h, Q_1, \ldots, Q_n$ 表示读写头处于这一格，并且图灵机当时的状态相应地为 $q_s, q_h, q_1, \ldots, q_n$ 中的一个。

谓词逻辑不是可判定的

描述图灵机运行过程的语言

- 我们需要函数符号 d, l, r（即二元谓词符号），$d(x)$ 表示 x 正下方的一格，$l(x)$ 表示 x 左边那格，$r(x)$ 表示 x 右边那格
- 令 E 和 W 是两个二元谓词符号我们用 Exy 表示 y 在 x（同一行）的右侧（东侧），Wxy 表示 y 在 x（同一行）的左侧（西侧）

谓词逻辑不是可判定的

下面这些语句对所有图灵机都满足（非完整）

- $\forall x \forall y(Exy \rightarrow (x \neq y \wedge \neg Wxy))$
- $\forall x(\exists y(x\neg y \wedge Exy) \wedge \exists z(x\neg z \wedge Wxz))$——每一行是双向无穷的

谓词逻辑不是可判定的

下面这些语句对所有图灵机都满足（非完整）

- $\forall x \forall y(m(x) = m(y) \to x = y)$（$m$ 可以是 d, r, l 之一）
  ——这几个函数都是 ——的：对每个格子，只有一个格子向下/左/右移动一格能到它
- $\forall x(\neg F x \to \exists y d(y) = x)$ ——不是 F 行的都有前一行
- $\forall x(d(r(x)) = r(d(x)))$ —— “殊途同归”

谓词逻辑不是可判定的

下面这些语句对所有给定状态集的图灵机都满足（非完整）

- $\exists!x(Fx \wedge Q_s x \wedge 1x)$ ——有且仅有一格，它是在第一行的，读写头所在的那格，此时图灵机状态为 q_s，该格中符号为 1。
- $\forall x(Qx \rightarrow \neg Q'x)$（$Q$ 是 $Q_s, Q_h, Q_1 \ldots, Q_n$ 中任何一个，Q' 是其中任何一个不同于 Q 的）——图灵机至多只能处于一个状态中

谓词逻辑不是可判定的

下面这些语句对所有给定状态集的图灵机都满足（非完整）

- $\forall x(Q^{\vee}x \vee \exists y(Exy \wedge Q^{\vee}y) \vee \exists y(Wxy \wedge Q^{\vee}y))$（我们用 $Q^{\vee}x$ 作为 $(Q_s x \vee Q_h x \vee Q_1 x \vee \cdots \vee Q_n x)$ 的缩写）——每一行读写头至少在某一格上
- $\forall x \forall y(Qx \to (Exy \vee Wxy) \to \neg Q^{\vee}y)$（$Q$ 可以说 $Q_s, \ldots, Q_n$ 中任何一个）——每一行读写头至多在一格上

谓词逻辑不是可判定的

下面这些语句对所有给定状态集的图灵机都满足（非完整）

- $\forall x((0x \wedge \neg 1x) \vee (1x \wedge \neg 0x))$——每个格子中有且仅有一个符号 0 或 1
- $\forall x(\neg Q^{\vee} x \rightarrow (0x \leftrightarrow 0d(x)))$——如果某个时刻读写头不在这格上，那么下个时刻这个格子中的符号不会变

谓词逻辑不是可判定的

描述给定图灵机的指令集

- 假设有指令 $q_1 0 1 q_2$，我们可以写一句

  $$\forall x\Big(Q_1 x \to 0x \to \big(1d(x) \wedge Q_2 d(x)\big)\Big)$$

- 假设有指令 $q_1 1 L q_2$，我们可以写一句

  $$\forall x\Big(Q_1 x \to 1x \to Q_2 d(l(x))\Big)$$

谓词逻辑不是可判定的

描述一个输入

- 例如，输入 n

  $\exists x(Fx \wedge Q_s x \wedge 1x \wedge \cdots \wedge 1r^n(x) \wedge \forall y((Wxy \vee Er^n(x)y) \rightarrow 0y))$

  其中，$r^n(x)$ 是 $r(\ldots r(x)\ldots)$（迭代 n 次）的缩写

不停机

- $\forall x \neg Q_h x$

谓词逻辑不是可判定的

- 给定图灵机 M（包括给定状态集和指令集）和输入 n，把上面提到的语句合取起来就是我们要的 $f(M,n)$
- 注意，在 $f(M,n)$ 我们用到了等词和一系列二元谓词符号

祝大家期末顺利