arXiv:2302.06444v1 [math.LO] 13 Feb 2023

超限可计算性的对称性

Lorenzo Galeotti[1], Ethan S. Lewis[2], Benedikt L¨owe[3,4,5]

[1] 阿姆斯特丹大学学院，Postbus 94160，1090 GD 阿姆斯特丹，荷兰；l.galeotti@uva.nl[2] 1179 W 1200 S，斯普林维尔，UT 84663，美利坚合众国；etonlels@gmail.com[3] 逻辑、语言与计算研究所，阿姆斯特丹大学，Postbus 94242，1090GE 阿姆斯特丹，荷兰；b.loewe@uva.nl[4] 汉堡大学数学系，Bundesstrasse 55，20146 汉堡，德国[5] 丘吉尔学院、露西·卡文迪什学院、圣埃德蒙学院及纯数学与数理统计系，剑桥大学，Storey's Way，剑桥，CB3 0DS，英格兰

摘要。有限图灵计算在输入、输出、程序、时间和存储空间之间具有一种基本对称性。标准的超限计算模型打破了这种对称性；我们探讨恢复这种对称性的方法，并研究由此产生的计算模型。在由序数集构造的宇宙中，该模型展现出与有限图灵计算相同的对称性，但这一论断独立于冯·诺伊曼–哥德尔–伯奈斯集合论。

1 引言

计算理论的一个基本特征是，可计算性的构成要素，即输入/输出、程序、时间和存储空间，可以被视为同一类对象：自然数（必要时通过编码实现）。图灵机接收一个有限符号串作为输入，以一个有限符号串作为其程序，并产生一个有限符号串作为输出。此外，其纸带和时间流都由自然数作为索引。因此，由于有限符号串可以编码为一个自然数，所有这些对象都属于同一类型。

我们将这一特征称为对称性。对称性的各个方面贯穿于计算理论的一般框架：输入与程序之间的对称性是软件原理（通用机的存在性）以及 s-m-n 定理的原因；程序与时间之间的对称性构成了之字形方法的基础。该方法通过将程序空间与时间的笛卡尔积认同为 $\mathbb{N}\times\mathbb{N}$ 并使用康托尔之字形函数，使我们能够将无穷多个计算并行化为一个计算。

最早的超限计算模型是 Hamkins-Kidder 机器，或称无限时间图灵机（ITTM），其定义见 [9]。这些机器的存储空间序型为 ω，但允许计算具有任意

超限序数长度，从而打破了时间与空间之间的对称性。正如 [20,11,5,17,23,24] 中所述，这种不对称性使其复杂性理论与普通复杂性理论大为不同。

在 [13,14], Koepke 对 Hamkins-Kidder 机器进行了对称化，并定义了如今称为 Koepke 机器或序数图灵机的概念：Koepke 机器拥有一条以序数为索引的类大小的磁带，并在序数时间内运行，从而重新确立了时间与存储空间之间的对称性。[1] 然而，Koepke 机器并不具备我们在有限图灵计算中发现的完全对称性：尽管时间和存储空间由任意序数表示，程序仍然是有限对象。

本文将为计算模型和可计算性提供一个通用框架，使我们能够用抽象的术语来表述对称性的探索；这将在§ 3中完成。在此框架下，我们将在§4中定义相关的可计算性模型，即普通图灵可计算性、Hamkins-Kidder 可计算性、Koepke 可计算性，以及我们提出的新概念——对称可计算性。我们将在§5中研究对称可计算性的基本性质，最后在§ 6中证明，在冯·诺依曼–哥德尔–伯奈斯类理论（NBGC）中无法证明对称可计算性的完全对称性：对称性成立，当且仅当宇宙是由一个序数集合构造出来的。

本文包含第二作者在第一作者和第三作者指导下完成的硕士论文 [16]中的结果。这些结果见于 [2,8,3] 和 [1, 练习 3.9。7]。

2 类论

本文采用冯·诺依曼–格德尔–伯奈斯类论。[2] 语言是通常的集合论语言 $\mathrm{L}_{\in}$，其中只有一个二元关系符号 $\in$。我们定义一元谓词 $\mathrm{set}(x) := \exists y(x \in y)$。利用该谓词，我们可以定义两个集合量词 $\exists^{\mathrm{set}} x\varphi := \exists x(\mathrm{set}(x) \wedge \varphi)$ 和 $\forall^{\mathrm{set}} x\varphi := \forall x(\mathrm{set}(x) \to \varphi)$。如果一个公式的所有量词都是集合量词，就称它为集合论的。在此语境下，我们用 AC 表示集合的选择公理，即“每个集合 x 都有一个选择函数”这一陈述，并将其与全局选择公理相区别，后者是“存在一个全局选择类函数”这一陈述。我们用 NBG 表示不含全局选择公理的冯·诺依曼–格德尔–伯奈斯类论 [12, 第 70 页：公理 A–D]，用 NBGC 表示在 NBG 中加入全局选择公理后得到的理论 [12, 第 70 页：公理 A–E]。Easton 的一个著名结果表明，如果 NBG 是相容的，那么 NBG +AC 不能证明全局选择公理（参见例如 [6, 定理 3.1]）。

我们可以通过递归地将所有量词替换为相应的集合量词，把语言 L 中的公式转化为集合论公式。

[1] Carl 在 [1, 第 9] 章中论证，Koepke 机器是有限运算和复杂性理论的自然无穷类比，这一点已在 [4]中得到详细探讨。[2] 更多详情请参见，例如， [18, 第 4]章。

这使我们能够表述 NBGC 的著名保守性定理（例如参见 [6, 推论 4.1 和定理 4.2]）：[3]

定理 1 若 φ 是 $L_\in$中的一个语句，则 ZFC $\vdash \varphi$ 当且仅当 NBGC $\vdash \varphi^{\mathrm{set}}$当且仅当 NBG +AC $\vdash \varphi^{\mathrm{set}}$。

我们将序数集的可构造性公理定义为如下命题："存在一个序数集合 x，使得 V=L[x]"。这是一个集合论语句，并且蕴含全局选择公理；因此，定理 1 蕴含以下结果。

定理 2 若 NBGC 一致，则 NBGC 既不能证明也不能否证序数集的可构造性公理。

3 图灵计算与可计算性的一般框架

我们将在一个一般的背景下讨论对称性，使相关的可计算性模型成为一般框架的特例。我们的一般框架将在所有序数构成的类序数类上运行，并将所有集合构成的类 V 视为这些机器的潜在程序。本文中的所有计算模型都是图灵机的变体：它们有一条按序数编址、类长的纸带，[4] 以及一个根据程序在纸带上移动的读写头。本文余下部分中，我们固定使用一个至少包含两个元素 0 和 1 的有限字母表 Σ。

图灵硬件与计算。在最高抽象层次上，我们讨论图灵硬件：纸带和读写头，以及对它们工作方式的描述。我们假设纸带始终以序数为索引，并被划分为离散单元，其中可以写入 Σ 中的一个符号；我们还假设时间由离散时刻构成，并以序数为索引，且在每个时刻，读写头都位于某个单元格上；最后，我们假设离散状态也以序数为索引。

对于类 X 和 Y，我们将 $f\colon X \dashrightarrow Y$ 写作"f 是一个类函数，其定义域 dom(f) $\subseteq X$，值域 ran(f) $\subseteq Y$"，并且当且仅当 $x \in$dom(f) 且 $f(x)\uparrow$ 时，$f(x)\downarrow$；否则为 $f(x)\uparrow$。我们用从 Ord 到 Σ 的任意偏类函数表示纸带内容；我们将这些对象组成的类记作 $\Sigma^{(\mathrm{Ord})}$。我们将考虑该类的若干相关子类：$\Sigma^{<\mathrm{Ord}} := \{x \in \Sigma^{(\mathrm{Ord})}$；dom($x$) $\in$ Ord$\}$、$\Sigma^{\mathrm{FS}} := \{x \in \Sigma^{(\mathrm{Ord})}$；dom($x$) 为有限集$\}$、$\Sigma^{\omega} := \{x \in \Sigma^{(\mathrm{Ord})}$；dom($x$) $= \omega\}$，

[3] 有关该定理历史的更多信息，请参阅 [6, 第 242] 页和 [7, 第 381] 页。[4] 对于大多数可计算性模型而言，纸带的数量无关紧要；然而，对于 Hamkins-Kidder 机器，单带机器与三带机器有所不同（参见 [10]）。由于我们不详细讨论 Hamkins-Kidder 机器，这一点与本文无关。

$\Sigma^* = \Sigma^{<\omega} := \{x \in \Sigma^{(\mathrm{Ord})} ; \mathrm{dom}(x) \in \omega\}$，以及 $\Sigma^{\mathrm{O}} := \{x \in \Sigma^{(\mathrm{Ord})} ; |\mathrm{dom}(x)| = 1$ 且 $\mathrm{ran}(x) = \{0\}\}$。

类 Σ^{O} 和 Σ^{FS} 分别是序数类 Ord 和 $\mathrm{Ord}^{<\omega}$ 的表示。类 Ord 和 Σ^{O} 之间存在一个规范双射；类 Σ^{FS}和 $\mathrm{Ord}^{<\omega}$ 可通过 G¨odel 配对函数对应起来。[5] 此外，G¨odel 配对函数给出了 Ord 与 $\mathrm{Ord}^{<\omega}$ 之间一个可定义的双射，以及一个双射 $(w, v) \mapsto w * v : \Sigma^{<\mathrm{Ord}} \times \Sigma^{<\mathrm{Ord}} \to \Sigma^{<\mathrm{Ord}}$。

机器的一个快照由纸带内容、一个状态和读写头的位置组成，也就是说，它是一个来自 $\mathrm{Snap} := \Sigma^{(\mathrm{Ord})} \times \mathrm{Ord} \times \mathrm{Ord}$ 的元组。

读写头的行为由转移规则支配。转移规则是一种类函数，它描述了在给定读写头过去的行为和一个程序 p的情况下，读写头将执行什么操作。目前，我们仍允许所有集合都作为程序（我们认为这种规定属于软件的一部分），因此，转移函数是一个类函数 $T : \mathrm{Snap}^{<\mathrm{Ord}} \times \mathrm{V} \to \mathrm{Snap}$。固定一个转移函数 T 后，给定一个程序 p 和一个快照 $s = (x, \alpha, \beta)$，我们通过递归定义一个序数长度的快照序列：$C_{p,\, s}(0) := s$，以及 $C_{p,\, s}(\gamma) := T(C_{p,\, s} \restriction \gamma, p)$，其中 $\gamma > 0$。我们将此称为程序 p 以初始快照 s开始的计算。

本文只考虑两种不同的转移函数：有限转移函数 T_{f}，普通图灵机和 Hamkins-Kidder 机器使用该函数；以及超限转移函数 T_{t}，Koepke 机器使用该函数（定义见 §4）。

图灵软件。计算模型由硬件（即转移函数 T）和可用于计算的程序类 P 组成。指定程序类即可确定哪些计算是按照 P中的程序进行的计算。

本文只考虑两类程序：有限程序类 P_{f} 和超限程序类 P_{t}（定义见 § 4）。

可计算性。计算模型确定了一个计算类，但尚未说明这些计算的作用。为说明这一点，我们考虑普通的图灵计算概念：对于每个程序和快照，我们都会得到一个无限快照序列，但这些无限序列有许多种解释方式。遵循图灵最初的开创性定义 [21, § 2], ，我们指定起始状态和停机状态，给出停机计算的定义，然后将计算解释为产生一个偏函数（定义见 §4）。

抽象地说，我们称一个解释由一个偏类函数I 组成，它将一个转移函数 T 和每个程序 $p \in P$ 映射到一个偏函数 $I(T, p) : \Sigma^{(\mathrm{Ord})} \dashrightarrow \Sigma^{(\mathrm{Ord})}$，以及一个称为解释的定义域的类 $D \subseteq \Sigma^{(\mathrm{Ord})}$。可计算性模型是一个计算模型（即一个转移函数 T和一个程序类 P）与一个解释的组合。如果存在某个 $p \in P$，使得 $f = I(T, p) \restriction D$，我们就称 $f : D \dashrightarrow D$ 在该可计算性模型下是可计算的。

[5] G¨odel 配对函数是 Ord 与 Ord^2 之间绝对可定义的类双射；参见 [12, 第 30–31] 页。

请注意，对于给定的计算模型和固定的解释函数， D有一定的选择余地。例如，对于通常的图灵计算和常见的教科书式解释，我们通常令 $D = \Sigma^*$ ，因此，可计算性是偏函数 $f\colon \Sigma^* \dashrightarrow \Sigma^*$的一种性质。不过，我们也可以考虑 $D = \Sigma^\omega$，即让图灵机处理长度为 ω的任意纸带内容，从而得到不同的可计算性模型。[6] 另一方面，如果固定计算模型和解释函数的类型， D 就不能完全自由选择：类 D 需要对运算 $I(T,p)$ 关于 $p \in P$封闭。例如，如果我们的计算模型是采用常见解释的 Koepke 机器，我们不能选择 $D = \Sigma^*$ ，甚至不能选择 $D = \Sigma^\omega$，因为存在某些程序，Koepke 机器执行这些程序时会产生一个不再属于 D 的输出。

在本文中，我们将考虑两类解释函数：有限解释 I_{f} 和超限解释 I_{t} （定义见第 4 节）。

4 计算模型的具体模型

程序。我们固定三个移动标记 $\mathrm{MT} := \{\blacktriangleleft, \blacktriangledown, \blacktriangleright\}$，它们分别表示读写头移动的指令（“向左移动”、“不执行任何操作”和“向右移动”），并用 m 作为移动标记的变量。在这些状态（按序数编号）中，我们特别指出三个状态：编号为 0 的初始状态、编号为 1 的停机状态，以及编号为 2 的极限状态。我们将 $\Sigma_\circ := \Sigma \cup \{\circ\}$ 写作一个特殊符号，其中 $\circ$ 表示空单元格。我们的所有程序都是偏函数 $p\colon \mathrm{Ord} \times \Sigma_\circ \dashrightarrow \mathrm{Ord} \times \Sigma_\circ \times \mathrm{MT}$。如果一个程序的定义域是有限的，我们称其为有限程序；如果其定义域是一个集合，则称其为超限程序。有限程序和超限程序的类分别记为 P_{f} 和 P_{t}。

通过类 Ord、$\mathrm{Ord} \times \Sigma_\circ$ 和 $\mathrm{Ord} \times \Sigma_\circ \times \mathrm{MT}$ 的规范识别，我们可以将程序编码为 $\Sigma^{(\mathrm{Ord})}$ 的元素。在我们的编码下，我们将 P_{f} 视为类 Σ^{FS} ，并将 P_{t} 视为类 $\Sigma^{<\mathrm{Ord}}$。

转移函数。给定一个程序 p，现在我们将定义转移函数 T_{f} （“有限转移函数”）和 T_{t} （“超限转移函数”）。它们在后继长度的序列上完全相同，并且与图灵为其机器定义的普通转移函数一致；而在极限长度的序列上则有所不同。

如果 $\vec{s} = (s_\xi;\ \xi < \gamma + 1)$ 是一个后继长度的快照序列，那么转移函数只依赖于 $s_\gamma = (x, \alpha, \beta)$，即列表中的最后一个快照。因此， $x \in \Sigma^{(\mathrm{Ord})}$ 是时间 γ时的纸带内容， α 是时间 γ时的状态，而 β 是时间 γ时读写头的位置。如果 $p(\alpha, x(\beta))$ 未定义，则令

[6] 这是一个有趣的可计算性模型，它揭示了单带机器与三带机器之间的差异（参见脚注 4）：由于停机计算中只有有限多个单元格会发生改变，对于单带机器，恒等函数是可计算的，而常值函数不可计算；相比之下，对于三带机器，值为 $w \in \Sigma^*$ 的常值函数是可计算的，但其他常值函数以及恒等函数都不可计算。

$T(\vec{s}) := s_\gamma$；否则，令 $p(\alpha, x(\beta)) = (\delta, \sigma, \mathrm{m})$。那么 $T(\vec{s}) = (y, \alpha^+, \beta^+)$，其中$\alpha^+ := \delta$，

$$y(\eta) := \begin{cases} x(\eta) & \text{if } \eta \neq \beta, \\ \sigma & \text{if } \eta = \beta, \end{cases} \quad \text{and } \beta^+ := \begin{cases} \beta - 1 & \text{if } \mathbf{m} = \blacktriangleleft \text{ and } \beta \text{ is a successor}, \\ 0 & \text{if } \mathbf{m} = \blacktriangleleft \text{ and } \beta \text{ is a limit}, \\ \beta + 1 & \text{if } \mathbf{m} = \blacktriangleright, \\ \beta & \text{if } \mathbf{m} = \blacktriangledown. \end{cases}$$

如果 $\vec{s} = (s_\xi ; \xi < \lambda)$ 是一个长度为 λ 的极限长度快照序列，其中 $s_\xi = (x_\xi, \alpha_\xi, \beta_\xi)$，那么这两个转移函数在纸带内容的定义上是一致的，但在读写头位置和状态的处理上存在分歧。记作$T_{\mathrm{f}}(\vec{s}) = (y, \alpha_{\mathrm{f}}, \beta_{\mathrm{f}})$ 以及 $T_{\mathrm{t}}(\vec{s}) = (y, \alpha_{\mathrm{t}}, \beta_{\mathrm{t}})$。对于纸带内容，我们假设 Σ 上具有一个全序，并定义 $y(\eta) := \operatorname{liminf}\{x_\xi(\eta); \xi < \lambda\}$。

有限转移函数 T_{f} 将读写头移动到单元格 0，并移动到极限状态（索引为 2），即 $\alpha_{\mathrm{f}} := 2$ 和 $\beta_{\mathrm{f}} := 0$。请注意，在使用有限转移函数的任何计算中，读写头永远不会到达由无限序数索引的单元格。

超限转移函数 T_{t} 将读写头和单元格都移至序列中出现的序数的下极限，即 $\alpha_{\mathrm{t}} := \operatorname{liminf}\{\alpha_\xi ; \xi <\lambda\}$ 和 $\beta_{\mathrm{t}} := \operatorname{liminf}\{\beta_\xi ; \xi < \lambda \wedge \alpha_\xi = \alpha_{\mathrm{t}}\}$。

解释。我们为任意纸带内容 $x \in \Sigma^{(\mathrm{Ord})}$统一定义两个解释函数。这两种解释都接收纸带内容 x 和程序 p，定义初始快照 $s := (x, 0, 0)$，并生成程序 p 在初始快照 s 下的计算$C_{p,\ s}$。

有限解释 I_{f} 认为，如果存在一个自然数 n，使得 $C_{p,s}(n)$ 的状态为 1（即停机状态），那么计算就会停机；超限解释 I_{t} 认为，如果存在一个序数 α，使得 $C_{p,s}(\alpha)$ 的状态为 1，那么计算就会停机。如果这样的数存在，其中最小的一个称为该计算的停机时间。这隐含地定义了这些可计算性模型所考虑的时间：一般来说，我们称某个可计算性模型的相关时间为其停机时间的上确界。采用有限解释的模型，其相关时间至多为 ω；采用无限解释的模型，其相关时间至多为序数类 Ord。我们令 $\Omega \subseteq D$ 为一个与该模型相关时间相对应的子类，例如，如果相关时间为序数类 Ord，则令 $\Omega = \Sigma^{\mathrm{O}}$。

如果一次计算停机，我们就称其停机时间的纸带内容为该计算的输出。最后，对于 $I = I_{\mathrm{f}}$ 或 $I = I_{\mathrm{t}}$ 以及相应的停机概念，我们令 $I(T, p)(x) := y$，当且仅当 $C_{p,\ s}$ 停机且 y 是其输出；否则令$I(T, p)(x)\uparrow$。

可计算性模型。利用有限规格 T_{f}、P_{f}和 I_{f}，以及超限规格 T_{t}、P_{t}和 I_{t}，我们现在可以将已知的可计算性模型（以及一个新的模型）作为特例重新得到。

首先注意，如果转移函数是有限的，那么超出由自然数编号的单元格范围的任何纸带内容都与计算无关，因为读写头永远不会移动到这些单元格。因此，相关的输入域

必须是 Σ^* 或 Σ^ω。此外，如果解释函数是有限的，那么所有继续进行到 ω 或更远位置的计算都将在解释中被忽略，因此我们可以不失一般性地假设转移函数也是有限的。这就得到了表1中列出的可计算性模型。

转移程序	解释	输入/输出			
(a) 有限	有限	有限	Σ^*	普通图灵机
(b) 有限	有限	超限	Σ^ω	Hamkins-Kidder 机器
(c) 超限	有限	超限	$\Sigma^{<\mathrm{Ord}}$	Koepke 机器
(d) 超限	超限	超限	$\Sigma^{<\mathrm{Ord}}$	对称机器；参见 § 5

表1。所考察的可计算性模型

我们简要讨论所述模型中输入／输出的选择：

表1 (a)。由于转移函数是有限的，只有 Σ^* 和 Σ^ω 才适合作为输入／输出的选择。然而，由于解释也是有限的，停机计算永远无法读取整条无限纸带，因此 Σ^* 是输入／输出的自然选择。选择 Σ^ω 会得到脚注 6 中讨论的可计算性模型。该模型的相关时间为 ω。

表1 (b)。同样，在这种情况下，有限的转移函数意味着我们只能选择 Σ^* 或 Σ^ω 作为输入；然而， Σ^* 在解释的运算下并不封闭（Hamkins-Kidder 机器可以填满整条纸带，然后停机），因此 Σ^ω 是唯一剩下的自然选择。 [9,22] 中研究了 Hamkins-Kidder 机器的相关时间。

表1 (c)。与针对第 (a) 行给出的论证类似，停机的 Koepke 机器只会查看纸带上的一组单元格。因此，输入／输出的自然选择是 $\Sigma^{<\mathrm{Ord}}$。与 (a) 类似，也可以考虑 $D = \Sigma^{(\mathrm{Ord})}$ ，此时适用脚注 6 中的讨论。Koepke 可计算性的相关时间是由所有序数组成的序数类 Ord。若 $p \in \Sigma^{<\mathrm{Ord}}$，我们称偏函数 $f\colon \Sigma^{<\mathrm{Ord}} \dashrightarrow \Sigma^{<\mathrm{Ord}}$ 在参数为 p 时是 Koepke 可计算的，当且仅当偏函数 $p * w \mapsto f(w)$ 是 Koepke 可计算的。我们将在命题 4 中证明，这一概念等价于第 (d) 行引入的新概念。

表1（d）。在（d）行中引入的可计算性概念称为对称可计算性，它弥补了超限可计算性缺乏对称性的问题。在这个模型中，时间、空间和程序都是超限的。对称可计算性所涉及的时间是包含所有序数的序数类 Ord。

5 对称机器

在表1中，我们将对称可计算性模型定义为由超限转移函数、超限程序和超限解释给出的模型，并使用 $\Sigma^{<\mathrm{Ord}}$ 作为输入和输出。我们将相应的计算模型称为

对称机器。与 Hamkins-Kidder 机器（其时间远多于空间）和 Koepke 机器（与其可用的时间和空间相比，其程序极为微小）不同，对称机器具有集合大小的时间、空间和程序。它们所代表的可计算性模型，系统地将通常图灵计算中的“有限”一词替换为“集合大小的”。

命题 3 若 $f : \Sigma^{<\mathrm{Ord}} \dashrightarrow \Sigma^{<\mathrm{Ord}}$ 是一个集合，则 f 是对称可计算的。

证明概要。显然，若 $x \in \Sigma^{<\mathrm{Ord}}$，则存在一个超限程序 p，它在空输入时会产生 x（只需明确指定 x 的值）。

由于 f 是一个集合，取某个 ξ 和 $\Sigma^{(\xi)} := \{w \in \Sigma^{(\mathrm{Ord})} \,\mathrm{dom}(w) \subseteq \xi\}$，使得 $\mathrm{dom}(f) \subseteq \Sigma^{(\xi)}$ 且 $\mathrm{ran}(f) \subseteq \Sigma^{(\xi)}$。根据选择公理，取一个双射 g，它将某个序数 λ 与 $\Sigma^{(\xi)}$ 对应起来。我们通过令 $x_f(\alpha,\beta,0) := g(\alpha)(\beta)$ 且 $x_f(\alpha,\beta,1) := f(g(\alpha))(\beta)$ 来定义 $x_f : \lambda \cdot \xi \cdot 2 \dashrightarrow \Sigma$。

现在寻找一个将 x_f 写入纸带的程序。给定输入 $w \in \Sigma^{(\xi)}$后，我们现在可以如下确定 $f(w)$：搜索 x_f 的 0-分量，直到找到 w；当你在索引 α处找到 w时，输出 $\{(\beta, x_f(\alpha,\beta,1)) : \beta < \xi\}$。Q.E.D.

命题 4 偏函数 $f : \Sigma^{<\mathrm{Ord}} \dashrightarrow \Sigma^{<\mathrm{Ord}}$是对称可计算的，当且仅当存在一个 $p \in \Sigma^{<\mathrm{Ord}}$，使得 f 在参数 p下是 Koepke 可计算的。

证明概要。根据命题 3，任意参数 p 都是对称可计算的，因此，如果 f 在参数 p下是 Koepke 可计算的，则它可以如下对称计算：给定输入 w后，先计算 p，然后计算 $w * p$，再计算 $f(p * w)$。

反过来，注意到就硬件和解释而言，Koepke 机器实际上就是对称机器。因此，通用 Koepke 机器也是通用对称机器，即存在一个 Koepke 机器u，使得对于所有超限程序 p 和所有 $w \in \Sigma^{<\mathrm{Ord}}$，我们有

$$I_{\mathrm{t}}(T_{\mathrm{t}}, u)(p * w) = I_{\mathrm{t}}(T_{\mathrm{t}}, p)(w).$$

因此，如果偏函数 f 由程序 p对称计算，那么它就是带参数 p的 Koepke 可计算函数。Q.E.D.

A与往常一样，我们可以定义停机问题 by

$$K := \{v * w \in \Sigma^{<\mathrm{Ord}} ; I_{\mathrm{t}}(T_{\mathrm{t}}, v)(w)\downarrow\}$$

（其中 v 被解释为超限程序）。通常的证明表明，K不能被对称计算。

现在我们将更仔细地考察对称可计算性的对称性性质，并探讨它是否与普通图灵可计算性所展现的对称性完全对应。对于普通图灵机，时间

和空间都以自然数为索引；然而，程序和输入/输出乍看之下并非自然数，而是由有限集中的元素构成的有限序列，也就是说，通过某种编码，它们是 Σ^*的元素。在这种情况下，对称性体现在存在一个可计算的编码函数，它将Σ^* 和 ω对应起来。通过这种编码，我们可以将可计算性模型的四种不同参数视为同一类型的对象。

在对称计算的情形下，“有限”一词系统性地被“集合大小”取代，因此时间和空间以序数为索引，程序以及输入/输出（在编码意义上）都是 $\Sigma^{<\mathrm{Ord}}$ 的元素。遗憾的是，一般而言，我们无法将 $\Sigma^{<\mathrm{Ord}}$ 与序数类 Ord 等同起来：§3 中提到的序数类 Ord、$\mathrm{Ord}^{<\omega}$、Σ^{O} 和 Σ^{FS} 之间的编码可以由 Koepke 机器执行（例如参见 [13, 第 4] 节），但类 $\Sigma^{<\mathrm{Ord}}$ 是一种截然不同的对象：除其他性质外，它包含整个康托空间（函数 $f\colon\omega \to \Sigma$），因此，若存在一种将 $\Sigma^{<\mathrm{Ord}}$ 的元素编码为序数的可计算编码，就会导出实数的可计算良序。因此，若无额外的集合论假设，就无法证明存在这样的类函数。

6 对称性条件

我们在抽象框架内给出半可判定性和可计算可枚举性的定义。对于普通图灵可计算性模型，这些定义与通常的定义一致。

定义 5 设某个可计算性模型由转移函数 T、程序类 P 和定义域类为 D 的解释 I 给出。设 $A \subseteq D$ 为非空类，并设 ψ_A 为满足以下条件的函数：若 $w \in A$，则 $\psi_A(w) = 0$；否则 $\psi_A(w)\uparrow$（伪特征函数）。那么，若 ψ_A 可计算，就称 A 为半可判定。固定某个表示该模型相关时间的 $\Omega \subseteq D$，若存在程序 $p \in P$，使得 $A = \{I(T,p)(w)\,;\ w \in \Omega\}$，则称 A 为可计算枚举。

定理 6（民间定理）对于普通图灵可计算性模型和任意非空集合 $A \subseteq \Sigma^*$，以下结论成立：

(i) 集合 A 为半可判定，当且仅当它是某个部分可计算函数 $f\colon \Sigma^* \dashrightarrow \Sigma^*$ 的值域；(ii) 集合 A 为可计算枚举，当且仅当它是半可判定的。

等价关系 (i) 是一个经典教材论证 [19, 定理 V]；对于等价关系 (ii)，正向推导是 (i) 的直接推论，而反向推导则利用了 Σ^* 与相关时间 ω 之间的可计算双射。因此，将这一证明改编为对称可计算性的情形，将保留等价关系 (i) 以及 (ii) 的正向推导。

定理 7 对于对称可计算性模型和任意非空类$A \subseteq \Sigma^{<\mathrm{Ord}}$，以下结论成立：

(i) 类 A 是半可判定的，当且仅当它是可计算类函数 $f\colon \Sigma^{<\mathrm{Ord}} \dashrightarrow \Sigma^{<\mathrm{Ord}}$ 的值域；(ii) 如果类 A是可计算枚举的，那么它是半可判定的。

与定理 6 相比，定理 7 缺少 (ii) 的逆命题；事实证明，这一差异对于我们从 § 5 寻求所需的对称性至关重要。我们用 SC 表示陈述“类 $\Sigma^{<\mathrm{Ord}}$ 是对称可计算枚举的”，将其称为对称性条件，并注意到，根据 §2 的含义，它是一个集合论语句。

命题 8 对称性条件 SC 等价于以下陈述：“每个对称半可判定类都是对称可计算枚举的”。

证明略。显然， $\Sigma^{<\mathrm{Ord}}$ 是半可判定的，因此“⇐”显而易见。对于“⇒”，设 $g : \Sigma^{\mathrm{O}} \to \Sigma^{<\mathrm{Ord}}$ 为一个可计算枚举， A 为任意半可判定类。根据定理 7，我们有一个可计算满射 $f : \Sigma^{<\mathrm{Ord}} \to A$。于是， $f \bigcirc g$枚举 A。Q.E.D.

对称性条件表明，类 $\Sigma^{<\mathrm{Ord}}$ 与序数类 Ord 可以通过 SC 提供的可计算枚举相互对应。因此，假设 SC 成立，时间、空间、程序以及输入/输出都可以视为同一类型的对象，而对称可计算性模型具有普通图灵可计算性所体现的对称性。

我们现在将看到，SC 独立于 NBG，并刻画 SC 在何种情况下成立。证明这一刻画的关键要素是以下结果，它是对 [13, 定理 62]的直接相对化。

定理 9（Koepke）设 x 为序数集合。那么，任意 $w \in \Sigma^{<\mathrm{Ord}}$ 属于 $\mathrm{L}[x]$ ，当且仅当存在有限程序 p 和 $v \in \Sigma^{\mathrm{O}}$ ，使得 p 的 Koepke 计算以 v 为输入、以 x 为参数时停机，并产生输出 w。

证明概要。反向结论由以下事实推出：从参数 x 出发的 Koepke 计算是绝对定义的。因此，如果一台 Koepke 机器在输入 $x * v$时产生输出 w，那么 w属于每个同时包含 x 和 v的模型。由于 $v \in \Sigma^{\mathrm{O}} \subseteq \mathrm{L}[x]$，我们有 $w \in \mathrm{L}[x]$。对于正向结论，假设 $w \in \mathrm{L}[x]$ ，并设 α 为一个指数闭序数，使得$w \in \mathrm{L}_\alpha[x]$。那么，根据 [15, 定理 7 (a)]， w 是 α-Koepke 可计算的，因此可由给出 α的参数进行 Koepke 计算，也就是说，可由 $\Sigma^{\mathrm{O}} \subseteq \Sigma^{\mathrm{FS}}$中的一个参数进行 Koepke 计算。Q.E.D.

引理 10 若 x 是序数集合且 $\Sigma^{<\mathrm{Ord}} \subseteq \mathrm{L}[x]$，则 $\mathrm{V}=\mathrm{L}[x]$。

证明思路。假设 $\Sigma^{<\mathrm{Ord}} \subseteq \mathrm{L}[x]$。反设 $\mathrm{V} \neq \mathrm{L}[x]$。令 A 为一个不属于 $\mathrm{L}[x]$ 的 $\in$-极小集合，即 $A \notin \mathrm{L}[x]$，但 $A \subseteq \mathrm{L}[x]$。存在一个可由 x 定义的双射 $G : \mathrm{Ord} \to \mathrm{L}[x]$ （例如参见 [12, 第 193] 页）。

定义 $w(\alpha)\downarrow = \mathbf{0}$ 当且仅当 $G(\alpha) \in A$；于是 $w \in \Sigma^{<\mathrm{Ord}} \subseteq \mathrm{L}[x]$。但这样一来，$A = \{G(\alpha) \in \mathrm{L}[x]; w(\alpha) = \mathbf{0}\}$，因此 $A \in \mathrm{L}[x]$。Q.E.D.

定理 11 对称性条件 SC 成立，当且仅当宇宙可由一个序数集合构造。

证明概要。对于“(ii)⇒(i)”，使用（可计算的）G¨odel 配对函数，得到一个可计算双射 $C : \Sigma^{\mathrm{O}} \to \Sigma^{\mathrm{O}} \times \Sigma^{\mathrm{FS}}$，并将有限程序与 Σ^{FS}对应起来。若 $u \in \Sigma^{\mathrm{O}}$，令 $C(u) = (v, p)$，并令 $F(u)$ 为在输入 v 上运行有限程序 p 的结果。由定理 9，F 枚举 $\Sigma^{<\mathrm{Ord}}$。

对于“(i)⇒(ii)”，假设 SC 成立。令 p 为可计算枚举 $\Sigma^{<\mathrm{Ord}}$ 的程序（它可以编码为一个序数集合）。由引理 10，只需证明 $\Sigma^{<\mathrm{Ord}} \subseteq \mathrm{L}[p]$。但这可由以下事实推出：$p$ 定义了一个从序数类到 $\Sigma^{<\mathrm{Ord}}$的类满射。Q.E.D.

由定理 2 和定理 11 可知，SC 独立于 NBGC。我们注意到，在 V=L 这一特殊情形下，Koepke 可计算性与对称可计算性等价（参见 [1, 练习 3.9.7 (d)]）；然而，如果取任意一个不可构造的序数集合 z，那么根据定理 11，$\mathrm{L}[z]$ 是 SC 的一个模型；集合 z 是对称可计算的（由命题 3 可知），但根据定理 9（令 $x = \varnothing$），它不是 Koepke 可计算的，因此这两种可计算性模型并不相同。这也回答了 [16, 关于如何区分更强版本 SC$_\kappa$ （“$\Sigma^{<\mathrm{Ord}}$ 可由大小为 $<\kappa$ 的程序可计算枚举”）的12] 问题 5。例如，若 x 是一个不可构造实数，那么 SC$_{\aleph_1}$ 在 L[x]中成立，但 SC$_{\aleph_0}$不成立。

参考文献

1. M. Carl。 *Ordinal Computability. An Introduction to Infinitary Machines*, *De Gruyter Series in Logic and Its Applications*第9卷。De Gruyter，2019年。2. M. Carl。空间有界 OTM 与 REG$^\infty$。 *Computability*，11：41–56，2022年。3. M. Carl、L. Galeotti 和 R. Paßmann。无穷直觉主义集合论的可实现性。 *Ann. Pure Appl. Log.* 即将出版，arxiv:2009.12172。4. M. Carl、B. L¨owe 和 B. Rin。Koepke 机器与无穷命题语言的可满足性。载于 J. Kari、F. Manea 和 I. Petre 编， *UnveilingDynamics and Complexity*， *13th Conference on Computability in Europe*， *CiE 2017*， *Turku Finland June 12-16 2017 Proceedings* 10307 *Lecture Notes in* ，，，，，第 卷， *Computer Science* 187 197 2017 页码 –。Springer，。5. V. Deolalikar、J. D. Hamkins 和 R. Schindler。无限时间图灵机的 P $\neq$ NP $\cap$ co-NP。 *J. Log. Comput.*，15(5)：577–592，2005年。6. U. Felgner。集合与类上的选择函数。载于 G. H. M¨uller 编， *Sets and Classes: On The Work by Paul Bernays* 84 *Studies in Logic and the* ，第 卷，页码 –。 *Foundations of Mathematics* 217 255 1976 Elsevier，。7. J. Ferreir´ *Labyrinth of Thought: A History of Set Theory and Its Role in Modern* os。 。 *Mathematics* ¨ 2007 Birkhauser Basel，巴塞尔，。

8. L. Galeotti。超现实 Blum-Shub-Smale 机器。载于 F. Manea、B. Martin、D. Paulusma 和 G. Primiero 编，*Computing with Foresight and Industry, 15th Conference on Computability in Europe CiE 2019 Durham UK , , , , , July 15–*19，2019年，*Proceedings*，第11558卷 *Lecture Notes in Computer Science*，页 13–24。Springer，2019年。9. J. D. Hamkins 和 A. Lewis。无限时间图灵机。*J. Symb. Log.*，65 2 567 604，2000（）：–。10. J. D. Hamkins 和 D. E. Seabold。仅使用一条纸带的无限时间图灵机。*Mathematical Logic Quarterly*，47(2)：271–287，2001年。11. J. D. Hamkins 和 P. D. Welch。几乎所有 f的 $\mathrm{P}^f \neq \mathrm{NP}^f$。*Math. Log. Q.*，49 5 536 540，2003（）：–。12. T. S. Jech。*Set Theory*。Springer 数学专著。Springer-Verlag，第三千年版，2003年。13. P. Koepke。序数上的图灵计算。*Bull. Symb. Log.*，11 397，2005 3 377（）：—。14. P. Koepke。序数可计算性。载于 K. Ambos-Spies、B. L¨owe 和 W. Merkle 编，*Mathematical Theory and Computational Practice 5th Conference on Com- , , , , , putability in Europe CiE 2009 Heidelberg Germany July 19-24 2009. Proceed- , 第 卷, ings 5635 Lecture Notes in Computer Science 280 289* 页码 –。Springer，2009年。15. P. Koepke 和 B. Seyfferth。序数机器与可容许递归论。*Annals of Pure and Applied Logic*，160(3)：310–318，2009年。16. E. S. Lewis。无限程序计算。硕士论文，阿姆斯特丹大学，2018年。ILLC 出版物 MoL-2018-14。17. B. L¨owe。无穷计算的空间界。载于 A. Beckmann、U. Berger、B. L¨owe 和 J. V. Tucker 编，*Logical Approaches to Computational Barriers, Second Conference on Computability in Europe, CiE 2006, Swansea, UK, July* 2006，*Proceedings*，第3988卷 *Lecture Notes in Computer Science*，页319–329，2006年。18. E. Mendelson。*Introduction to Mathematical Logic*。数学教材。CRC Press，第6版，2015年。19. H. Rogers。*Theory of Recursive Functions and Effective Computability*。MIT Press，1987年。20. R. Schindler。无限时间图灵机的 P ≠ NP。*Monatsh. Math.*，139：335–340，2003年。21. A. M. Turing。论可计算数及其在 *Entschei- dungsproblem Proceedings of the London Mathematical Society* 42 230 265, 1937 中的应用。, ：–。22. P. D. Welch。离散超限时间图灵机模型的特征：停机时间、稳定化时间和范式定理。*Theor. Comput. Sci.*，410 426 442，2009：–。23. J. Winter。无限时间图灵机中的空间复杂性。硕士论文，阿姆斯特丹大学，2007年。ILLC 出版物 MoL-2007-14。24. J. Winter。对于无限时间图灵机，P = PSPACE 是否成立? 载于 M. Archibald、V. Brattka、V. Goranko 和 B. L¨owe 编，*Infinity in Logic and Computation, International Conference ILC 2007 Cape Town South Africa , , , , , November 3–5, 2007 Revised Selected Papers 5489 Lecture Notes in Artificial Intelli- , , 第 gence 126 137, 2009* 卷，页码 –。