R-复杂度微积分的性质

Rares Folea[1] 和 Emil Slusanschi[2]

arXiv:2601.18437v1 [cs.CC] 26 Jan 2026

本文介绍了 *r*-复杂度演算的一系列一般性质。*r*-复杂度是一种复杂度度量，用于评估现实世界算法的性能和渐近行为。本文阐述了自反性、传递性和对称性等特征，讨论了不同类别 *r*-复杂度之间的若干转换规则，并确立了基本算术原理。此外，本文还考察了该体系中的加法性质，并将其特征与传统巴赫曼-朗道记号中常用的特征进行了比较。通过运用这些性质，本研究旨在推动 *r*-复杂度新应用的探索与开发，并加快微积分在这一精细化复杂度模型中的普及。

关键词： r-复杂度，计算复杂性，代码复杂度，复杂度模型的属性，渐近分析，算法性能

1. 引言

算法是计算机科学的核心，因为它们是主要的构建模块；渐近记号则被广泛接受为通过计算所分析算法的复杂度来估算其性能的主要方法 [5, 6, 9]。在检查和比较两个或更多代码片段时，计算渐近复杂度很重要，原因有很多：它有助于预测解决方案的可扩展性，可用于比较不同算法，也能帮助工程师理解每种解决方案固有的局限性与权衡。此外，它还能帮助识别代码中的瓶颈，开发人员随后便可对其进行优化。算法复杂度通常采用 Bachmann-Landau [2, 3]渐近记号表示，包括大Θ、大 O 和大Ω度量。

[1]布加勒斯特理工大学自动控制与计算机学院计算机科学与工程系，罗马尼亚；工程与激光及加速器应用博士研究生院（S.D.I.A.L.A.），电子邮箱：rares.folea@stud.acs.upb.ro[2]布加勒斯特理工大学自动控制与计算机学院计算机科学与工程系，罗马尼亚，电子邮箱：emil.slusanschi@cs.pub.ro

渐近代码复杂度是多个领域关注的主题，包括逻辑学、编码理论和计算系统。一般而言，渐近复杂度主要旨在描绘算法对资源（例如 CPU 时间总量或内存）的使用量如何随着输入规模的增加而增长，从而估算算法在更大输入下的表现。借助这一指标，渐近记号可以比较多个算法的效率；例如，对于足够大的输入规模，一个 $\Theta(n \cdot log\ n)$ 算法比一个 $\Theta(n^2)$ 算法更高效。随着时间推移，人们提出了许多渐近记号 [4, 5, 6, 7, 8, 9, 10, 11]，但其基本概念本质上始终不变：渐近记号的定义基于将不同的复杂度函数与给定的参考函数进行比较。此外，人们也一直在尝试将确定给定程序 [12, 13]的复杂度类这一过程自动化。

r-复杂度模型 [1] 是巴赫曼-朗道记号的一种替代渐近记号，能够为相似程序提供更有用的复杂度反馈，并揭示细微见解，即使这些算法属于同一传统复杂度类。该模型旨在弥补传统模型的若干不足，例如，为复杂度函数相似的两个相似算法之间的差异提供解决方案 $v,\ w:\ \mathbb{N} \longrightarrow \mathbb{R}$，对于它们，存在 g，使得 $v(n) \in \Theta(g(n))$ 且 $w(n) \in \Theta(g(n))$.

在**第 2 节**中，我们介绍研究动机，并展示一个情境：在传统分析中，两个解决同一问题的算法会被归入同一个复杂度类，尽管它们在实际中的性能差异巨大。接着，在**第 3 节**中，我们给出各种 r-复杂度类的一些形式化定义及相关推论，包括大 r-Theta、大 r-O 和大 r-Omega，并介绍判定某个元素是否属于给定集合的替代标准。**第 4 节** 探讨 r-复杂度中的自反性、传递性、对称性和投影等性质。**第 5 节** 考察 r-复杂度类在加法运算下的行为，最后，在**第 6 节**中，我们总结研究的主要成果。

2. 动机

r-复杂度模型通过分析主导常数来展现其价值，揭示某些算法行为上的显著差异，而这些差异会对总执行时间产生重大影响。为了帮助软件工程师更直接地协作和分享知识，使他们能够使用 r-复杂度渐近记法讨论和比较算法，本文介绍了 r-复杂度类中的一组常用微积分性质，可用于更快地处理复杂度。

在传统的巴赫曼-朗道记号中，大Θ定义 [4, 5, 6] 指出，当且仅当函数 $f(n)$属于 $\Theta(g(n))$复杂度集合时

存在两个分别记为 c_1 和 c_2的正数常量，以及一个正整数 n_0，使得对于任何大于或等于 n_0的 n 值， $f(n)$均被c_1 （下界）和 c_2 （上界）常量乘以 $g(n)$所限定:

$$\Theta(g(n)) = \{f(n) \mid \exists c_1, c_2 \in \mathbb{R}_+^*, \exists n_0 \in \mathbb{N}^* \; s.t. \; c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n), \; \forall n \geq n_0\}$$

为了说明缺乏区分可能导致开发者误入歧途的情形，请考虑两个解决相同问题的算法 $Alg1$ 和 $Alg2$，它们由不同的复杂度函数定义： f_1， f_2 : $\mathbb{N} \longrightarrow \mathbb{R}$，且这两个函数恰好相差一个常数： $f_1 = x \cdot f_2$，其中 $x \in \mathbb{R}_+$， $x > 1$。

这些复杂度函数意味着这两个算法属于同一个大Θ复杂度集合，因为存在 g，使得$f_2 \in \Theta(g(n))$，例如:

$$\exists c_1, c_2 \in \mathbb{R}_+^*, \exists n_0 \in \mathbb{N}^* \; s.t. \;\; c_1 \cdot g(n) \leq f_2(n) \leq c_2 \cdot g(n) \;\; , \; \forall n \geq n_0$$

holds for $c_1^{'}, c_2^{'} \in \mathbb{R}_+^*, n_0^{'} \in \mathbb{N}^*$, where:

$$\begin{cases} c_1^{'} = x \cdot c_1 \\ c_2^{'} = x \cdot c_2 \\ n_0^{'} = n_0 \end{cases}$$

because $\forall n \geq n_0^{'}$:

$$c_1^{'} \cdot g(n) = x \cdot c_1 \cdot g(n) \leq x \cdot f_2(n) = f_1(n) = x \cdot f_2(n) \leq x \cdot c_2 \cdot g(n) = c_2^{'} \cdot g(n)$$

Th意味着:

$$c_1^{'} \cdot g(n) \leq f_1(n) \leq c_2^{'} \cdot g(n)$$

因此，根据大Θ的定义， $f_2 \in \Theta(g(n)) \Rightarrow f_1 \in \Theta(g(n))$。

同样，对于 $c_1^{'} = \dfrac{1}{x} \cdot c_1$、 $c_2^{'} = \dfrac{1}{x} \cdot c_2$ 和$n_0 = n_0$，有 $f_1 \in \Theta(g(n))$ $\Rightarrow f_2 \in \Theta(g(n))$。

请注意，即使 $f_1 > f_2$，这两个复杂度函数也属于同一个复杂度类。这一观察表明，两个算法的复杂度函数即使相差一个常数（哪怕高达 $2024^{2024,}$ ），仍属于同一个复杂度类，尽管实际运行时间可能相差五千个数量级以上。作为比较，只有当这两个复杂度函数相差 30 个数量级时 $f_1 = 10^{30} \cdot f_2$，才意味着对于给定输入 n，若 $Alg2$ 在 1 阿秒 （10^{-9} 纳秒的一部分）内结束运行，那么 $Alg1$ 预计约需 3 千年才能结束运行。尽管时间差异巨大，经典复杂度模型无法区分这类复杂度函数仅相差一些常数的算法子集，

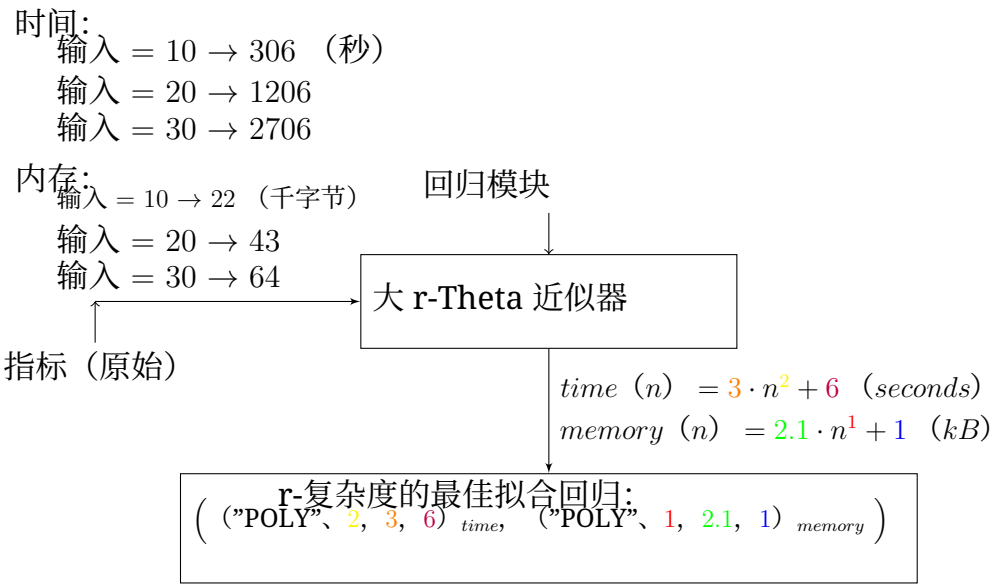

FIGURE 1. 展示了使用两项原始指标——时间和空间——在 [14],中进行回归拟合过程的一个示例。研究的目标是利用简化函数近似实际的大 r-Theta 复杂度函数，该函数通过多个性能指标对程序行为建模，之后可用于构建基于复杂度的代码嵌入。嵌入中分析时间复杂度的一部分 POLY 2 3 6 表示一个二次多项式，其中 3 是多项式平方项的系数，6 是截距。

一般而言，了解这些复杂度特征后，就更容易评估算法在各种场景下的有效性和适用性。

文献 [1], 介绍了 r-复杂度的一些应用，并对一种普遍看法提出了质疑：尽管 Strassen 算法的渐近复杂度较低，但对于任何输入规模，它总是优于传统矩阵乘法。论文结果表明，对于少于 2500 万个元素的输入，缓存友好的循环排序算法尽管复杂度有所增加，仍能更快完成。该论文还根据相关的 r- 复杂度函数，估算出使用暴力算法生成所有可能局面的总时间约为 10^{123} 秒。这是论证当前受限于现有计算能力，无法通过暴力算法完美解决国际象棋的核心依据。对于有限输入，这一论点无法通过传统复杂度模型得以确立。

r-复杂度已证明有用的另一个领域是构建代表程序逻辑的代码嵌入 [14]，这些嵌入之后可用于

将算法分类为不同类别，例如贪心算法、暴力算法或分治法。嵌入演算基于这样一种思路：通过将回归模型拟合到算法执行期间由性能分析器获取的指标，并将这些指标视为输入规模的函数，从而估算相关 r-复杂度函数的系数。例如，图1所示的一个实际案例通过实验数据表明，算法的执行时间与输入规模的平方成正比，而算法使用的内存则与输入数据的规模成正比。利用 r-复杂度模型来估算大 r-Θ 类，我们可以了解具体的比例关系，例如执行时间随输入规模的平方增长，且增长倍数为三；或者内存的增长速度是输入规模的 2。1 倍。

3. r-复杂度集合的定义

r-复杂度集合的定义最初在 [1] 中提出。本文通过将推论与定义中阐述的知识联系起来，扩展这些陈述的表达能力，从而凸显渐近演算中的重要推论。本节所述推论的有效性直接源于这些类的定义。

R-复杂度微积分既可以通过序列极限，也可以通过函数极限来进行。任取两个复杂度函数 $f, g : \mathbb{N}_+ \longrightarrow \mathbb{R}$。以下符号和术语将用于描述由函数 $f : \mathbb{N} \longrightarrow \mathbb{R}$ 表征的算法复杂度的渐近行为。我们将所有复杂度微积分的集合定义为 $\mathcal{F} = \{f : \mathbb{N} \longrightarrow \mathbb{R}\}$。

定义 3.1. 大 *r-Theta* 被定义为一个集合，它表示在渐近分析语境下，与函数 g 具有相同增长率的一组数学函数。更正式地说，大 *r-Theta r-* 复杂度类定义为 [1] ：

$$\Theta_r(g(n)) = \{f \in \mathcal{F} \mid \forall c_1, c_2 \in \mathbb{R}_+^* \ s.t. c_1 < r < c_2, \exists n_0 \in \mathbb{N}^* \\ s.t. \ \ c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \ , \ \forall n \geq n_0\}$$

推论 3.1。 函数 f 属于由函数 g 定义的大 *r-Theta* 复杂度类的判据可直接由下述内容得出：

$$f \in \Theta_r(g(n)) \Leftrightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} = r$$

我们将采用 [1], 中所述的大 r-O、大 r-Omega、小r-O 和小 r-Omega 的相同定义，但需要说明的是，后两种模型仅出于完整性考虑而定义，因为与传统记法 [16, 17] 相比，它们并未提供任何新意。类似于大 r-Theta，根据大 r-O、大r-Omega、小 r-O 和小r-Omega 类的定义，以下推论可直接得到：

推论 3.2。 由 g 定义的大 *r-O* 类中接纳 f 的标准：

$$f \in \mathcal{O}_r(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = l,\ l \in [0, r]$$

推论 3.3. 由g定义的 f 属于大 *r-Omega* 类的判据：

$$f \in \Omega_r(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = l,\ l \in [r, \infty)$$

f 属于小r-O 和小 r-Ω 复杂度类的判据与传统记法相比没有变化。

推论 3.4. 由 g定义的 f 属于小 *r-O* 类的判据：

$$f \in o_r(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

推论 3.5. 由 g定义的 f 属于小 *r-Ω* 类的判据：

$$f \in \omega_r(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$

4. r-复杂度中的常见性质

本章将介绍 r-复杂度中关于自反性、传递性、对称性和投影 的新推论，并将其与传统 Bachmann–Landau 符号中的等价性质进行比较；这些性质在文献 [9, 10, 11, 15] 中已得到充分分析并确立。通过识别 r-复杂度中的自反性、对称性和转置性质，我们认为，在这一模型中可以更高效地解决某些问题。为确立本节中的结果，我们将依赖第 3 节中引入的定义，以及 [1] 中描述的基础模型。

定理 4.1。 大 *r-Theta* 符号下 *r*-复杂度的自反性

$$f \in \Theta_r\left(\frac{1}{r} \cdot f(n)\right)\ \forall r \neq 0$$

证明。 根据 $\Theta_r(f(n))$ 的定义

$$\Theta_r(f(n)) = \{f' \in \mathcal{F} \mid \forall c_1, c_2 \in \mathbb{R}_+^*\ s.t. c_1 < r < c_2, \exists n_0 \in \mathbb{N}^* \\ s.t.\ \ c_1 \cdot f(n) \leq f'(n) \leq c_2 \cdot f(n)\ ,\ \forall n \geq n_0\}$$

使用代换 $f(n) \longleftarrow \frac{1}{r} \cdot f(n)$ -

$$\Theta_r\left(\frac{1}{r} \cdot f(n)\right) = \{f' \in \mathcal{F} \mid \forall c_1, c_2 \in \mathbb{R}_+^*\ s.t. c_1 < r < c_2, \exists n_0 \in \mathbb{N}^* \\ s.t.\ \frac{1}{r} \cdot\ c_1 \cdot f(n) \leq f'(n) \leq \frac{1}{r} \cdot c_2 \cdot f(n)\ ,\ \forall n \geq n_0\}$$

通过选择 $c_1' = \frac{1}{r} \cdot c_1$, $c_2' = \frac{1}{r} \cdot c_2$

$$\Theta_r\left(\frac{1}{r}\cdot f(n)\right) = \{f' \in \mathcal{F} \mid \forall c_1', c_2' \in \mathbb{R}_+^* \ s.t. c_1' < 1 < c_2', \exists n_0 \in \mathbb{N}^* \\ s.t. \ \ c_1' \cdot f(n) \le f'(n) \le c_2' \cdot f(n) \ , \ \forall n \ge n_0\}$$

对于任意 $x \in \mathbb{R}^*, \forall c_1, \ c_2 \ \ s.t. c_1 \le 1 \le c_2$, 我们有 $x \cdot c_1 \le x \le x \cdot c_2$。因此，若$f$: $\mathbb{N} \longrightarrow \mathbb{R}, \forall c_1, \ c_2 \ \ s.t. c_1 \le 1 \le c_2$, 则有 $f(n) \cdot c_1 \le f(n) \le f(n) \cdot c_2 \ \ \forall n \in \mathbb{N}^*$ 或 $f(n) \cdot c_1 \ge f(n) \ge f(n) \cdot c_2 \ \ \forall n \in \mathbb{N}^*$

因此:

$$\forall c_1', c_2' \in \mathbb{R}_+^* \ s.t. c_1' < 1 < c_2', \exists n_0 = 1 s.t. \ c_1' \cdot f(n) \le f(n) \le c_2' \cdot f(n) \quad \forall n \ge n_0 = 1 \Rightarrow \\ f \in \Theta_r\left(\frac{1}{r}\cdot f(n)\right) \ \forall r \ne 0$$

■

定理 4.2. 大 *r-O* 符号在 *r*-复杂度中的自反性

$$f \in \mathcal{O}_r\left(x \cdot f(n)\right) \ \forall x \ge \frac{1}{r}$$

证明。 根据 $\mathcal{O}_r(f(n))$ 的定义

$$\mathcal{O}_r(f(n)) = \{f' \in \mathcal{F} \mid \forall c \in \mathbb{R}_+^* \ s.t. \ r < c, \exists n_0 \in \mathbb{N}^* \ s.t. \ f'(n) \le c \cdot f(n) \ \ \forall n \ge n_{\ 0}\}$$

使用代换 $f(n) \longleftarrow x \cdot f(n) \ \forall x \ge \frac{1}{r}, \ \ \forall r \ne 0$

$$\mathcal{O}_r(x \cdot f(n)) = \{f' \in \mathcal{F} \mid \forall c \in \mathbb{R}_+^* \ s.t. \ r < c, \exists n_0 \in \mathbb{N}^* \ s.t. \ f'(n) \le c \cdot x \cdot f(n) \ \ \forall n \ge n_0\}$$

如果 $r < c$ 且 $x \ge \frac{1}{r}$, 那么 $c \cdot x \ge 1, \ \ \forall r, \ \ c, \ \ x \ \ r \ne 0$ 因此，若 f: $\mathbb{N} \longrightarrow \mathbb{R}$, 则有$f(n) \le 1 \cdot f(n) \le c \cdot x \cdot f(n) \quad \forall n \in \mathbb{N}^*$。

因此:

$$\forall x \ge \frac{1}{r}, \forall c \in \mathbb{R}_+^* \ s.t. \ r < c, \exists n_0 = 1 s.t. \ \ f(n) \le c \cdot x \cdot f(n) \ \ \forall n \in \mathbb{N}^*, \forall n \ge n_0 = 1 \Rightarrow \\ f \in \mathcal{O}_r\left(x \cdot f(n)\right) \ \forall x \ge \frac{1}{r}$$

■

定理 4.3。 大 *r-Omega* 符号中 *r*-复杂度的自反性

$$f \in \Omega_r\left(x \cdot f(n)\right) \ \forall x \le \frac{1}{r}$$

证明。 根据 $\Omega_r(f(n))$ 的定义

$$\Omega_r(f(n)) = \{f' \in \mathcal{F} \mid \forall c \in \mathbb{R}_+^* \ s.t. \ c < r, \exists n_0 \in \mathbb{N}^* \ s.t. \ f'(n) \ge c \cdot f(n) \ \ \forall n \ge n_0\}$$

使用代换 $f(n) \longleftarrow x \cdot f(n) \ \forall x \le \quad \frac{1}{r} \ \forall r \ne 0$

$$\Omega_r(x \cdot f(n)) = \{f' \in \mathcal{F} \mid \forall c \in \mathbb{R}_+^* \ s.t. \ c < r, \exists n_0 \in \mathbb{N}^* \ s.t. \ f'(n) \ge x \cdot c \cdot f(n) \ \ \forall n \ge n_0\}$$

如果 $r > c$和 $x \leq \frac{1}{r}$，那么 $c \cdot x \leq 1$， $\forall r$， c， x $r \neq 0$

因此：

$\forall x \leq \frac{1}{r}$, $\forall c \in \mathbb{R}_+^* \; s.t. \; c < r, \exists n_0 = 1 s.t. \; f(n) \geq c \cdot x \cdot f(n) \;\; \forall n \in \mathbb{N}^*, \forall n \geq n_0 = 1 \Rightarrow$

$$f \in \Omega_r \left(x \cdot f(n) \right) \; \forall x \leq \frac{1}{r}$$

■

注 4.1. 在 *r*-复杂度中，对于小 *r-O* 和小 *r-Ω* 符号，自反性不成立，因为这些集合与经典的 *Bachmann–Landau* 符号集合相同，而小 *Omega* 类也不满足自反性。

$$f \notin o_r(f(n))$$
$$f \notin \omega_r(f(n))$$

定理 4.4. *r*-复杂度中的传递性——大 *r-Theta* 符号：

$$f \in \Theta_r(g(n)), g \in \Theta_{r'}(h(n)) \Rightarrow f \in \Theta_{r \cdot r'}(h(n))$$

证明。 $f \in \Theta_r(g(n)) \; \Rightarrow \forall c_1, c_2 \in \mathbb{R}_+^* \; s.t. c_1 < r < c_2, \exists n_0 \in \mathbb{N}^*$

$s.t. \;\; c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \quad \forall n \geq n_0$

$g \in \Theta_{r'}(h(n)) \Rightarrow \forall c_1, \;\; c_2 \in \mathbb{R}_+^* \; s.t. c_1 < r' < c_2, \;\; \exists n_0' \in \mathbb{N}^*$

$s.t. \;\; c_1 \cdot h(n) \leq g(n) \leq c_2 \cdot h(n) \quad \forall n \geq n_0'$

因此：

$\forall c_1, c_2, c_1', c_2' \in \mathbb{R}_+^* \; s.t. c_1 < r < c_2, c_1' < r' < c_2', \exists n_0'' = max(n_0, n_0') \in \mathbb{N}^* \; s.t.$

$c_1' \cdot c_1 \cdot h(n) \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \leq c_2' \cdot c_2 \cdot h(n) \;,\; \forall n \geq n_0''$

For $c_1'' = c_1 \cdot c_1', c_2'' = c_2 \cdot c_2'$, we have: $c_1'' < r \cdot r' < c_2''$.

As a consequence:

$$\forall c_1'', c_2'' \in \mathbb{R}_+^* \; s.t. c_1'' < r \cdot r' < c_2'' \;\; \exists n_0'' = max(n_0, n_0') \in \mathbb{N}^* s.t.$$
$$c_1'' \cdot h(n) \leq f(n) \leq c_2'' \cdot h(n) \;,\; \forall n \geq n_0'' \Rightarrow f \in \Theta_{r \cdot r'}(h(n))$$

■

r-复杂度演算中的其余传递性也成立，其证明与上文给出的证明结构类似：

Theorem 4.5. *Transitivity in r-Complexity - Big r-O notation:*
$f \in \mathcal{O}_r(g(n)), g \in \mathcal{O}_r(h(n)) \Rightarrow f \in \mathcal{O}_{r \cdot r'}(h(n))$

定理 4.6. *r*-复杂度中的传递性 - 大 *r-Omega* 符号：$f \in \Omega_r(g(n)) \;\; g \in \Omega_r(h(n)) \;\Rightarrow f \in \Omega_{r \cdot r'}(h(n))$

定理 4.7。 *r*-复杂度中的传递性 - 小 *o_r* 记号：$f \in o_r$（g（n））
$g \in o_{r'}$（h（n））$\Rightarrow f \in o_{r \cdot r'}$（$h$（$n$））

定理 4.8。 *r*-复杂度中的传递性 - 小 *r*-Ω 记号$f \in \omega_r$（g（n））　$g \in \omega_{r'}$（h（n））$\Rightarrow f \in \omega_{r \cdot r'}$（$h$（$n$））

对称性是一种基本代数性质，在各种数学语境中都发挥着关键作用，因为它在许多情况下能够简化问题求解，也为更优雅地进行计算流程提供了方法。

定理 4.9. *r*-复杂度中的对称性：　$f \in \Theta_r(g(n)) \Rightarrow g \in \Theta_1(f(n))_r$

证明。　$f \in \Theta_r(g(n)) \Rightarrow \forall c_1, c_2 \in \mathbb{R}_+^* \ s.t. c_1 < r < c_2, \exists n_0 \in \mathbb{N}^* s.t. \ c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \quad \forall n \geq n_0$使用代换 $c_1' = \frac{c_1}{r}, c_2' = \frac{c_2}{r} \Rightarrow \forall c_1', c_2' \in \mathbb{R}_+^* \ s.t. c_1 < 1 < c_2, \exists n_0 \in \mathbb{N}^* s.t. \ c_1' \cdot g(n) \leq \frac{1}{r} \cdot f(n) \leq c_2' \cdot g(n) \quad \forall n \geq n_0$

前一个不等式可以改写为：

$$\begin{cases} g(n) \leq \frac{1}{c_1'} \cdot \frac{1}{r} \cdot f(n) \\ g(n) \geq \frac{1}{c_2'} \cdot \frac{1}{r} \cdot f(n) \end{cases}$$

使用记号 $c_1'' = \frac{1}{c_2'}$，$c_2'' = \frac{1}{c_1'}$ 该不等式变为：

$\forall c_1'', c_2'' \in \mathbb{R}_+^* \ s.t. c_1'' < r < c_2'', \ \exists n_0 \in \mathbb{N}^*$

$$c_1'' \cdot \frac{1}{r} \cdot f(n) \leq g(n) \leq c_2' \cdot \frac{1}{r} \cdot f(n) \ \ \forall n \geq n_0$$

因此，根据 $\Theta_r(f(n))$ 的定义，　$f \in \Theta_r(g(n)) \Rightarrow g \in \Theta_1(f(n))_r$ ■

定理 4.10。 *r*-复杂度中的转置对称性：　$f \in \mathcal{O}_r$（g（n））$\Leftrightarrow g \in \Omega_1$（$f$（$n$））$_r$

证明。 根据 $\mathcal{O}_r(f(n))$ 和 $f \in \mathcal{O}_r(g(n))$（$\mathcal{O}_r(g(n)) \supseteq \{f\}$）的定义：

$$\forall c \in \mathbb{R}_+^* \ s.t. \ r < c, \exists n_0 \in \mathbb{N}^* \ s.t. \ g(n) \leq c \cdot f(n), \ \forall n \geq n_0$$

我们可以将前面的关系改写如下：

$$\forall c \in \mathbb{R}_+^* \ s.t. \ r < c, \exists n_0 \in \mathbb{N}^* \ s.t. \ \frac{1}{c} \cdot g(n) \leq f(n), \ \forall n \geq n_0$$

代入常数 $c' = \frac{1}{c}$：

$$\forall c' \in \mathbb{R}_+^* \ s.t. \ c' < \frac{1}{r}, \exists n_0 \in \mathbb{N}^* \ s.t. \ f(n) \geq c' \cdot g(n), \ \forall n \geq n_0$$

因此：

$$\Omega_{\frac{1}{r}}(f(n)) \supseteq \{g\}$$

■

备注 4.2. 对于 *Smallr-o* 和 *Smallr-Omega*，转置对称性成立，因为这两个集合与 *Bachmann–Landau* 符号中定义的经典集合相等。

定理 4.11. *r*-复杂度中的投影性质：$f \in \Theta_r(g(n))$
$\Leftrightarrow f \in \mathcal{O}_r(g(n))$ 以及 $f \in \Omega_r(g(n))$
证明。 证明很直接，只需使用 Big$r$-Theta、Big$r$-O 和 Big$r$-Omega 的定义。

”$\Rightarrow$” $f \in \Theta_r(g(n)) \Rightarrow f \in \mathcal{O}_r(g(n))$ $f \in \Omega_r(g(n))$ 基于
$\Theta_r(g(n))$，$f \in \Theta_r(g(n)) \Rightarrow$
$\forall c_1, c_2 \in \mathbb{R}_+^* \; s.t. c_1 < r < c_2, \exists n_0 \in \mathbb{N}^* s.t. \; c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \quad \forall n \geq n_0$
通过拆分不等式：

$$\begin{cases} \forall c_1 \in \mathbb{R}_+^* \; s.t. c_1 < r, \exists n_0 \in \mathbb{N}^* \; s.t. \; c_1 \cdot g(n) \leq f(n), \; \forall n \geq n_0 \\ \forall c_2 \in \mathbb{R}_+^* \; s.t. r < c_2, \exists n_0 \in \mathbb{N}^* \; s.t. \; f(n) \leq c_2 \cdot g(n) \; , \; \forall n \geq n_0 \end{cases}$$

因此：

$$\begin{cases} f \in \mathcal{O}_r(g(n)) \\ f \in \Omega_r(g(n)) \end{cases}$$

”$\Leftarrow$” $f \in \mathcal{O}_r(g(n))$ $f \in \Omega_r(g(n)) \Rightarrow f \in \Theta_r(g(n))$ 根据大 r-O 和大 r-Ω 的定义：

$$\begin{cases} \forall c_1 \in \mathbb{R}_+^* \; s.t. c_1 < r, \exists n_0 \in \mathbb{N}^* \; s.t. \; c_1 \cdot g(n) \leq f(n), \; \forall n \geq n_0 \\ \forall c_2 \in \mathbb{R}_+^* \; s.t. r < c_2, \exists n_0' \in \mathbb{N}^* \; s.t. \; f(n) \leq c_2 \cdot g(n) \; , \; \forall n \geq n_0' \end{cases}$$

通过选择 $n_0'' = max(n_0, n_0')$：

$$\begin{cases} \forall c_1 \in \mathbb{R}_+^* \; s.t. c_1 < r, \Rightarrow \quad c_1 \cdot g(n) \leq f(n), \; \forall n \geq n_0'' \\ \forall c_2 \in \mathbb{R}_+^* \; s.t. r < c_2, \Rightarrow \; f(n) \leq c_2 \cdot g(n) \; , \; \forall n \geq n_0'' \end{cases}$$

合并这些不等式后，对于 $\forall n \geq n_0''$，有：
$\forall c_1, c_2 \in \mathbb{R}_+^* \; s.t. c_1 < r < c_2, \exists n_0'' = max(n_0, n_0') \; s.t. \; c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$
因此 $f \in \Theta_r(g(n))$。 ■

5. **加法性质**

类似于 Bachmann-Landau 记号中的微积分（包括大 O 运算），通过考察 r-复杂度类在加法下的行为，可以获得有价值的见解。对于每个给出的定理，我们都会给出一个推论，并采用宽松记法：将任意 r-复杂度类记号视为该集合中的一个任意函数。这样便可以写出如下所示的方程。

为了分析大 r-Theta 中的加法性质，以下关系适用于任意正确界定的函数 f、g、f'、g'、h：$\mathbb{N} \longrightarrow \mathbb{R}$，其中

$$h(n) = f'(n) + g'(n)$$

$\forall n \in \mathbb{N}$，其中 f'、g'是两个任意函数，满足 $f' \in \Theta_r(f)$ $g' \in \Theta_q(g)$ 和 r、$q \in \mathbb{R}_+$:

定理 5.1. 如果 $\lim_{n\to\infty} f(n)\ g(n) = 0 \Rightarrow h \in \Theta_q(g)$。

证明。 $\lim_{n\to\infty} f(n)\ g(n) = 0 \Rightarrow \lim_{n\to\infty} f'(n)\ g'(n) = 0 \Rightarrow \lim_{n\to\infty} g'(n) + f'(n)\ g'(n) = 1$并利用渐近分析部分的结果，我们有 $g'(n) + f'(n) = h(n) \in \Theta_1(g')$。

利用自反性， $h(n) \in \Theta_q(g)$。 ■

推论 5.1。 如果 $\lim_{n\to\infty} f(n)\ g(n) = 0$，则有以下关系:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \Rightarrow \Theta_r(f) + \Theta_q(g) = \Theta_q(g)$$

定理 5.2. 如果 $\lim_{n\to\infty} f(n)\ g(n) = \infty \Rightarrow h \in \Theta_r(f)$。

证明。 利用上一个结果，进行交换：$f \leftarrow g$， $g \leftarrow f$ 并考虑加法交换律，即可得到该命题的证明。 ■

推论 5.2。 如果 $\lim_{n\to\infty} f(n)\ g(n) = \infty$，则有以下关系:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \Rightarrow \Theta_r(f) + \Theta_q(g) = \Theta_r(f)$$

定理 5.3。 如果 $\lim_{n\to\infty} f(n)\ g(n) = t$，
$t \in \mathbb{R}_+ \Rightarrow h \in \Theta_r\left(f + \frac{r}{q} \cdot g\right)$。

证明。 $\lim_{n\to\infty} f(n)\ g(n) = t \Rightarrow \lim_{n\to\infty} f'(n)\ g'(n) = t \cdot \frac{r}{q} = t' \Rightarrow \lim_{n\to\infty} g'(n) + f'(n)\ g'(n) = t' + 1$ 并利用渐近分析一节中的结果，可得 $g'(n) + f'(n) = h(n) \in \Theta_{t'+1}(g')$。

利用自反性， $h(n) \in \Theta_{t'+1}(q \cdot g)$。

利用转换技巧，我们有 $h(n) \in \Theta_r(\frac{1}{r} \cdot t \cdot \frac{r}{q} + 1) \cdot q \cdot g)$。

通过换回 t，我们可以渐近地得到:

$$h(n) \in \Theta_r\left(\frac{1}{r} \cdot (\frac{f(n)}{g(n)} \cdot \frac{r}{q} + 1) \cdot q \cdot g\right)$$

Therefore $h \in \Theta_r\left(f + \frac{r}{q} \cdot g\right)$. ■

推论 5.3。 如果极限$_{n\to\infty}$ $\underline{f(n)}$ $g(n) = t$，$t \in \mathbb{R}_+$，则以下关系成立：

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = t,\ t \in \mathbb{R}_+ \Rightarrow \Theta_r(f) + \Theta_q(g) = \Theta_r(f + \frac{r}{q} \cdot g)$$

对于大 r-O 中的加法，任意定义正确的函数 f, g, f', g', h都满足以下关系： $\mathbb{N} \longrightarrow \mathbb{R}$，其中 $h(n) = f'(n) + g'(n)$ $\forall n \in \mathbb{N}$，其中 f', g'是两个任意函数，满足 $f' \in \mathcal{O}_r(f)$ $g' \in \mathcal{O}_q(g)$，且$r, q \in \mathbb{R}_+$：

定理 5.4。若 $\lim_{n\to\infty}$ $\underline{f(n)\, g(n)} = 0 \Rightarrow h \in \mathcal{O}_q(g)$。

推论 5.4。

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \Rightarrow \mathcal{O}_r(f) + \mathcal{O}_q(g) = \mathcal{O}_q(g)$$

定理 5.5。

推论 5.5。

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \Rightarrow \mathcal{O}_r(f) + \mathcal{O}_q(g) = \mathcal{O}_r(f)$$

定理 5.6。若$\lim_{n\to\infty}$ $\underline{f(n)\, g(n)} = t$， $t \in \mathbb{R}_+ \Rightarrow h \in \mathcal{O}_r\left(f + \frac{r}{q} \cdot g\right)$。

推论 5.6。

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = t,\ t \in \mathbb{R}_+ \Rightarrow \mathcal{O}_r(f) + \mathcal{O}_q(g) = \mathcal{O}_r(f + \frac{r}{q} \cdot g)$$

最新结果的证明与大 Θ 中加法的证明类似，使用了大 O_r 类所介绍的转换。

对于大 r-Omega 的加法，以下关系适用于任意定义良好的函数 f， g， f'， g'， h： $\mathbb{N} \longrightarrow \mathbb{R}$，其中 $h(n) = f'(n) + g'(n)$ $\forall n \in \mathbb{N}$，其中 f'， g' 是两个任意函数，使得 $f' \in \Omega_r(f)$ $g' \in \Omega_q(g)$ 且r， $q \in \mathbb{R}_+$：

定理 5.7。若$\lim_{n\to\infty}$ $\underline{f(n)\, g(n)} = 0 \Rightarrow h \in \Omega_q(g)$。

推论 5.7。

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \Rightarrow \Omega_r(f) + \Omega_q(g) = \Omega_q(g)$$

定理 5.8。如果 $\lim_{n\to\infty}$ $\underline{f(n)\, g(n)} = \infty \Rightarrow h \in \Omega_r(f)$。

推论 5.8。

$$\lim_{n\to\infty}\frac{f(n)}{g(n)}=\infty \Rightarrow \Omega_r(f)+\Omega_q(g)=\Omega_r(f)$$

定理 5.9。若 $\lim_{n\to\infty} f(n)\,g(n)=t,\quad t\in\mathbb{R}_+ \Rightarrow h\in\Omega_r\left(f+\frac{r}{q}\cdot g\right)$。

推论 5.9。

$$\lim_{n\to\infty}\frac{f(n)}{g(n)}=t,\ t\in\mathbb{R}_+ \Rightarrow \Omega_r(f)+\Omega_q(g)=\Omega_r(f+\frac{r}{q}\cdot g)$$

最新结果的证明与大 Θ 中加法的证明类似，使用了针对大 r-Omega 类所介绍的转换。

小 r-O 和小 r-Ω 中的加法具有巴赫曼-朗道记号中所述的相同性质。

6. 结论

r-复杂度 [1] 模型提出了一种不同的算法复杂度度量方法，旨在更精细地评估算法性能：它不仅进行渐近分析，还为有限输入提供相关估算。

与传统的巴赫曼-朗道记号相比，r-复杂度能够区分原本属于同一复杂度类的算法，为开发者选择最优方案提供更细致的见解。此外，r-复杂度类具有灵活性：只需采用 [1]中提出的简单转换技术，就能根据r的不同取值轻松转换定义这些类的函数。因此，r-复杂度框架有望改进工程师和开发者评估及优化算法的标准方法。第 2 节中展示的应用表明，r-复杂度提供了一个有价值的框架，能够超越渐近复杂度分析算法性能，从而实现更准确的预测（如矩阵乘法所示），并加深对程序行为的理解（如其在构建代码嵌入方面的适用性所示）。这一复杂度模型带来的关键见解之一是：尽管 Strassen 算法在理论上具有渐近优势，但对于实际输入规模，其速度仍不如传统矩阵乘法。

本文的主要贡献在于列举并证明 r-复杂度类中的一些微积分性质，例如自反性、传递性、对称性和投影。此外，本文还研究了 r-复杂度类在加法下的行为，以及 r-复杂度类与更传统的 Bachmann-Landau 复杂度类之间的转换规则。这些发现旨在帮助 r-复杂度的使用者

加快微积分运算，并有望推动这种算法分析与权衡比较方法得到更广泛的应用。

参 考 文 献

[1] Folea, Rares 和 Slusanschi, Emil-Ioan，《一种评估计算机程序性能与复杂度的新度量指标：评估算法复杂度和估算运行时间的传统方法的新途径》（2021年），2021年第23届控制系统与计算机科学国际会议（CSCS），157–164页。[2] Bachmann, Paul，《解析数论》[《解析数论》]（1894年），第2卷。莱比锡：Teubner，第402–403页。[3] Landau, Edmund，《素数分布理论手册》[《素数分布理论手册》]（1909年），莱比锡：B. G. Teubner，第883页。[4] Knuth, Donald E.，《大 O、大 Ω与大Θ》。ACM Sigact News 8.2（1976年）：18–24。[5] Giumale, C. A.《算法分析导论：理论与应用》（2004年），Polirom，罗马尼亚布加勒斯特。[6] Cormen, T. H.、Leiserson, C. E.、Rivest, R. L.、Stein, C.《算法导论》（2022年）。麻省理工学院出版社。[7] Wilf, Herbert S.《算法与复杂性》（2002年），AK Peters/CRC出版社。[8] Bovet, Daniel Pierre、Pierluigi Crescenzi 和 D. Bovet。《复杂性理论导论》（1994年），第7卷。伦敦：Prentice Hall。[9] Mogoşs, Andrei-Horia、Bianca Mogoşs 和 Adina Magda Florea，《一种新的渐近记号：弱 Θ》（2015年）。《工程数学问题》。[10] Affeldt, Reynald、Cyril Cohen 和 Damien Rouhling，《经典分析中渐近推理的形式化技术》。《形式化推理期刊》（2018年）。[11] Mala, Firdous Ahmad 和 Rouf Ali，《数学与计算机科学中的大 O》。《应用数学与计算》6.1（2022年）：1–3。[12] Phalke, Swapnil、Yogita Vaidya 和 Shilpa Metkar，《算法的大 O 时间复杂度分析》。2022年信号与信息处理国际会议（IConSIP）。IEEE，2022年。[13] Vaz, Rayner 等，《算法的大 O 自动分析》。2017年工程新兴技术国际会议（ICNTE）。IEEE，2017年。[14] Folea, R.、Iacob, R.、Slusanschi, E.、Rebedea, T.，《基于复杂度的代码嵌入》（2023年）。载于：Nguyen, N.T. 等，《计算集体智能》。2023年国际计算集体智能会议。《计算机科学讲义》，第14162卷。施普林格，Cham。[15] Goloveshkin, V. A. 和 M. V. Uljanov，《函数渐近增长的度量与计算机算法的复杂度分类》。《计算与信息科学新兴趋势期刊》3.11（2012年）。[16] Papadimitriou, Christos H.，《计算复杂性》，载于《计算机科学百科全书》，第260–265页，2003年。[17] Wegener, Ingo，《复杂性理论：探索高效算法的极限》。施普林格科学与商业媒体，2005年。