arXiv:2302.13764v1 [cs.CC] 27 Feb 2023

整环上的代数电路类的逻辑刻画

Timon Barlag、Florian Chudigiewitsch 和 Sabrina A. Gaube

Abstract. 我们将 Cucker 和 Meer 提出的实数域上代数电路的构造推广到任意无限整环，并将 $\mathrm{AC}_{\mathbb{R}}$ 和 $\mathrm{NC}_{\mathbb{R}}$类推广到这一情形。我们给出一个类似 I mmerman 定理的定理，表明对于这些推广后的形式体系，由常数深度、多项式规模电路判定的集合，与通过对一阶逻辑作适当推广而定义的集合相同。此外，我们讨论 Durand、Haak 和 Vollmer 提出的受限谓词逻辑的一种推广，并给出 AC_R 和 NC_R 层级的刻画。这些推广同样适用于布尔 AC 和 NC 层级。此外，我们还引入一种形式体系，以便比较上述某些复杂性类在不同底层整环上的情形。

1. Introduction

作为一种计算模型，布尔电路是理论计算机科学研究中的一个基本概念。布尔电路在计算复杂性领域尤其发挥着核心作用：它们有助于分析并行算法及相应的复杂性类，提供更精细的视角和新的证明方法，尤其适用于次多项式复杂性类。布尔电路的一种自然推广是：不再以真值作为输入——或者对代数研究者而言，以域 $\mathbb{Z}_2$ 作为输入——而是考虑来自某种其他代数结构的元素。这一方法源于 Blum、Shub 和 Smale 的研究，他们提出的 BSS 机器模型能够描述实数上的计算。随后，Blum、Shub、Smale 和 Cucker [Blu+98] 还将其推广到实数上的代数电路。

1.1. 我们的贡献。

本文给出了整环上电路复杂性类的逻辑刻画。具体而言，我们定义了经典电路复杂性层级 AC^i 和 NC^i 在任意整环上的自然推广。由此得到的复杂性类分别记为 AC^i_R 和 NC^i_R。我们调整元有限模型论的框架，定义一阶逻辑的若干扩展，以刻画这些新的复杂性类。

我们证明了一个 Immerman 风格定理，即 $\mathrm{FO}_R = \mathrm{AC}^0_R$，并提供一个框架，用于比较底层整环不同的类在复杂性理论上的差异，同时举例说明在哪些整环上 AC^0-类相等。

我们改造 Durand、Haak 和 Vollmer 用来对 AC^1 和 NC^1 [DHV18] 进行逻辑刻画的 GPR 算子，使其适用于元有限结构上的逻辑，并将其扩展到能够刻画整个 AC_R 和 NC_R 层级。

Date：2023年2月28日。*Key words and phrases.* 代数电路，描述复杂性。

最后，我们定义了一种形式体系，用于比较基于不同底层整环的复杂性类，并证明由复杂性类集合构成的层级随之出现；该层级中的每个集合都能够“模拟”层级中更低层的集合所包含的复杂性类。

1.2. 相关工作。另一种通常被称为“代数电路”的计算模型是 Valiant 电路 [Val79], ，它们是代数复杂性理论和几何复杂性理论这两个新兴领域中的基础模型。它们与我们在本文中分析的模型的区别在于，我们使用 $<$-门，而 Valiant 模型中并不存在这种门。这一区别在复杂性理论上具有重要意义，因为对于我们的模型，我们有 $\mathsf{NC}^i_{\mathbb{R}} \subsetneq \mathsf{NC}^{i+1}_{\mathbb{R}}$ [Cuc92]，但在 Valiant 模型中，对于任意域，特别是在实数集上，我们有 $\mathsf{VNC}^2 = \mathsf{VNC}^3 = \ldots = \mathsf{VP}$ [B¨ur13]。

1.3. 论文概要。我们首先概述代数和电路复杂性中的一些核心概念，这些概念是定义我们模型所必需的。随后，我们为任意整环建立代数电路模型，并类比布尔情形，定义由这些模型诱导出的相应复杂性类。之后，我们借鉴经典描述复杂性理论的思想，对所介绍的电路类给出逻辑刻画。然而，由于我们的模型现在包含一个无限成分，我们将以元有限模型论的基础为依据。

接下来，我们定义 R 上的一阶逻辑，并证明与经典情形一样，$\mathsf{AC}^0_R = \mathsf{FO}_R[\mathsf{Arb}_R] + \mathsf{SUM}_R + \mathsf{PROD}_R$。

在第4节中，我们给出 AC^i 和 NC^i的逻辑刻画。我们选择的工具是对 [DHV18]的受限谓词逻辑所作的改编。

在第5节中，我们探讨不同整环上的 AC^0_R-类之间的联系。

致谢。感谢 Sebastian Berndt 和 Anselm Haak 与我们进行了富有成效的讨论。

2. Preliminaries

首先，我们讨论本文的理论背景，并给出基本定义以及对本文所用符号的说明。

记号 2.1. 在本文中，我们通常用带上划线的字母（例如 $\overline{x}$）表示元组。

顾名思义，代数电路类会用到源自抽象代数的概念。我们先概述本文中使用的一些基本概念。关于这一主题的更多信息，读者可参阅 Aluffi [Alu09] 或 Lang [Lan02] 的著作。

定义2.2（环）。带单位元的环（$R, +, \times$）是由一个集合 R和两个二元运算 $+$ 和 $\times$ 组成的元组，并满足

- $1 \in R$
- $(R, +)$ 是一个阿贝尔群，即：– 对任意 $a, b, c \in R$，有 $(a + b) + c = a + (b + c)$ – 对任意 $a, b \in R$，有 $a + b = b + a$– 存在一个 $0 \in R$，有 $a + 0 = a$– 对任意 $a \in R$，存在一个逆元 $-a \in R$：$a + (-a) = 0$
- $(R, \times)$ 是一个幺半群，即：– $(a \times b) \times c = a \times (b \times c)$ – 存在一个 $1 \in R$ 使得 $a \times 1 = 1 \times a = a$

- 乘法关于加法具有分配性:

$$a \times (b + c) = a \times b + a \times c, \text{ for all } a, b, c \in R$$
$$(a + b) \times c = a \times c + b \times c, \text{ for all } a, b, c \in R$$

如果对所有 a、$b \in R$，$a \times b = b \times a$，则称环是交换的。 ◁

下文中，环均默认具有单位元。

定义 2.3（整环）。具有单位元的环 R 称为整环，如果它是交换环，且不存在作为零因子的元素 $a \in R$，也就是说，不存在 $a \in R \setminus \{0\}$ 使得存在 $b \in R \setminus \{0\}$ 满足 $a \times b = 0$。◁

在本文中，除非另有说明，我们均用 R 表示无限整环。

备注 2.4。特别地，我们要求对每个 $r \in R$，以下方程

$$r \times 0 = 0$$
$$0 \times r = 0$$

保留。

定义 2.5（多项式）。设 R 是一个环。未定元 x 上、系数取自 R 的多项式 $f(x)$ 是由 x 的非负指数项构成的有限线性组合，系数属于 R:

$$f(x) = \sum_{i \geq 0} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots,$$

其中，所有 a_i 都是 R 的元素（即系数），并且我们要求对于无穷多个 i，$a_i = 0$ 成立。若两个多项式的所有系数都相等，则称这两个多项式相等:

$$\sum_{i \geq 0} a_i x^i = \sum_{i \geq 0} b_i x^i \iff (\forall i \geq 0) \colon a_i = b_i.$$

在 R 上以 x 为系数的多项式集合记为 $R[x]$。 ◁

定义 2.6（多项式环）。设 $f(x) = \sum_{i \geq 0} a_i x^i$ 和 $g(x) = \sum_{i \geq 0} b_i x^i$ 是两个多项式。多项式环是集合 $R[x]$，并配备以下两种运算:

$$f(x) + g(x) \coloneqq \sum_{i \geq 0} (a_i + b_i) x^i$$

and

$$f(x) \times g(x) \coloneqq \sum_{k \geq 0} \sum_{\substack{0 \leq i \leq k, \\ 0 \leq j \leq k, \\ i+j=k}} a_i b_j x^{i+j}$$

这个环的单位元是 $1_{R[x]} = 1_R + 0x + 0x^2 + \cdots$。 ◁

注 2.7. 定义 2.6 中的多项式环也是一个带单位元的环。类似于定义 2.6，我们可以通过归纳应用定义 2.6，定义一个有有限个变量 x_1、.....、x_n 的多项式环 S：

$$S \coloneqq R[x_1, \ldots, x_n] \coloneqq (R[x_1, \ldots, x_{n-1}])[x_n],$$

其中 R 是一个环。注意，域上的多项式环是整环。

定义 2.8（环的伴随元素）。将数 $f \notin R$ 添入环 R 后，我们得到集合

$$R[f] \coloneqq \{a + f \cdot b \mid a, b \in R\}.$$

集合 $R[f]$ 以及以下两个运算

$$(a + f \cdot b) + (c + f \cdot d) \coloneqq (a + c) + f \cdot (b + d)$$

and

$$(a + f \cdot b) \cdot (c + f \cdot d) \coloneqq ac + f(ad + bc) + f^2(bd),$$

其中 $+$ 和 $\cdot$ 是环上的二元运算，该结构仍是一个环，其中 $1_{R[x]} = 1_R + 0f$ 是此环的单位元。$\lhd$

例 2.9. 环的常见例子包括 $\mathbb{Z}$、$\mathbb{Q}$、$\mathbb{R}$ 和 $\mathbb{C}$，以及添加了元素的集合，例如 $\mathbb{Z}[i]$、$\mathbb{Z}[\sqrt[k]{-1}]$、$\mathbb{Z}[\sqrt{p}, \sqrt{q}, \ldots]$、$\mathbb{R}[i_k]$、$\mathbb{Z}[i_k]$、.....，其中 $k \in \mathbb{N}$、$p \neq q$ 是素数，i_k 表示 -1的第 k个根，即 $i_k^k = -1$。此外，我们还可以添加 ℓ 个代数无关的素数根元素，其中 $2^\ell = k$，从而得到类似的构造。

注 2.10. $\mathbb{R}[i_k]$ 和 $\mathbb{Z}[i_k]$ 的记法并不唯一，但在底层环上构造电路的方法是类似的，添加的数所对应的占位符除外。例如，$\mathbb{Z}^4$ 可以表示 $\mathbb{Z}[i_4] = \mathbb{Z}[i_4, i_4^2, i_4^3]$或 $\mathbb{Z}[\sqrt{2}, \sqrt{5}] = \mathbb{Z}[\sqrt{2}, \sqrt{5}, \sqrt{10}]$，也可以表示许多其他环。由于我们只关注元组的结构，即伴随元素的系数，或者之后关注底层电路，因此，我们对任意 $k > 1$ 使用的 $\mathbb{Z}^k$ 简写并不唯一，也可能有所不同，例如，$\mathbb{Z}^2$ 可以表示 $\mathbb{Z}[i]$，也可以表示 $\mathbb{Q}$。在占位符记法的语境中，如果具体环的算术性质很重要，就必须明确说明。

定义 2.11（域）。域 $(F, +, \times)$ 是由一个集合 F和两个二元运算 $+$ 和 $\times$ 组成的元组，满足

- $1 \in F$
- $0 \in F$
- $(F, +)$ 是阿贝尔群（见 2.2）
- $(F, \times)$ 是阿贝尔群（见 2.2）
- 乘法关于加法满足分配律：

$$a \times (b + c) = a \times b + a \times c, \text{ for all } a, b, c \in F$$

分配律无需写成两个等式，因为乘法和加法都是交换的。$\lhd$

注 2.12。域的另一种刻画是：一个满足 $0 \neq 1$ 且每个元素 $a \neq 0 \in F$ 都可逆的交换环。

我们需要在整环 R 上定义某种序 $<$。在某些情况下，例如 $\mathbb{R}$ 和 $\mathbb{Z}$，我们有自然的序可供使用。在其他情况下，例如 $\mathbb{C}$，我们必须构造一种序。这种序不必在乘法下封闭；我们只需能够区分不同的数。因此，可以对元组定义一种序（$z_1 = a_1 + b_1 i, z_2 = a_2 + b_2 i \in \mathbb{C}$：$(a_1, b_1) <_{\mathbb{C}} (a_2, b_2) \iff a_1 <_{\mathbb{R}} a_2$ 或 $a_1 = a_2$ 和 $b_1 <_{\mathbb{R}} b_2$）。

定义 2.13。集合 S 上的严格全序是 S 上的二元关系 $<$，它具有非自反性、传递性和完全性。$\lhd$

例 2.14。对于某个域 $\mathbb{F}_{p^k}$、某个素数 p 和一个自然数 k，我们将有限多个元素列成一个列表，然后在这个列表上使用字典序。例如，设 $R = \mathbb{Z}_3$。那么我们定义 $<_{\mathbb{Z}_3}$ 为 $0 < 1 < 2$。

在复数域上，这种 $<$ 有多种可能的定义。令 $z = a + bi \in \mathbb{C}$，并令 $<_1$ 为有序对（$\sqrt{a^2 + b^2}$，a）上的字典序。此外，令 $<_2$ 为形如（a，b）的有序对上的字典序。由于这两种变体都能区分不同的复数，因此二者均可行。

定义 2.15。环 R 上的严格全序 $<$ 诱导出如下符号函数：

$$\operatorname{sign}_{(R,<)}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases} \qquad \lhd$$

下文中，除非另有明确说明，符号 R 表示一个带有严格全序 $<$ 的无限整环 R。我们所考虑的大多数环都具有自然序。在这种情况下，我们省略序符号。

2.1. R上的代数电路。由于本文是对成熟的布尔电路的推广，因此假定读者具备电路复杂性的相关背景知识。介绍这一主题的标准文献是 Vollmer 所著的书 [Vol99]。Cucker [Cuc92]首次引入了实数上的代数电路这一推广。类比于布尔电路，Barlag 和 Vollmer 定义了无界扇入版本的代数电路 [BV21]。

定义 2.16。我们将整环 R 上带有严格全序 $<$的代数电路 C，或简称为 R-电路，定义为一种有向无环图。它包含以下类型的门：

输入节点的入度为 0，并包含电路相应的输入值。常量节点的入度为 0，并标记有 R 的元素
算术节点的入度可以是任意值 ≥ 1，但不超过电路中的节点数，并且标记为 $+$ 或 $\times$。

比较（$<$）节点的入度为 2。
输出节点的入度为 1，并包含电路计算完成后的输出值。

节点不能多次成为同一节点的前驱，因此这些代数电路中节点的出度受电路中门的数量限制。

在代数电路的计算过程中，算术门将其前驱门的值作为函数自变量，计算各自对应的函数；比较门也以同样的方式计算 $<$ 的特征函数。计算结束时，输出门的值即为计算结果。$\lhd$

与经典情形不同，经典情形将字母表 Σ 上的词作为输入，因此语言被定义为字母表的克莱尼闭包的子集（记作 $L \subseteq \Sigma^*$）；而我们将整环元素构成的向量作为输入。类比于 Σ^*，对于整环 R，我们记：

$$R^* = \bigcup_{k \in \mathbb{N}_0} R^k$$

用 $|x|$ 表示 x 的长度，也就是说，如果 $x \in R^k$，那么 $|x| = k$。

注 2.17。我们将使用“代数电路”一词来描述 Blum、Cucker、Shub 和 Smale 意义下的电路 [Blu+98]，而不是例如 [BV21] 中的算术电路，以便将它们与 Valiant 意义下的算术电路区分开来，参见例如 [BCS97]。Valiant 电路本质上是不含符号门的代数电路 [Blu+98，第 350] 页。

注 2.18。在特殊情形 $R = \mathbb{Z}_2$ 下，上述定义恰好得到布尔电路。

定义 2.19。我们将电路中的门数称为电路的规模，并将从输入门到输出门的最长路径称为电路的深度。$\triangleleft$

注 2.20。与先前引入具有无界扇入门的代数电路的方式不同 [，BV21], 在此语境下的代数电路使用比较门而非符号门。这是因为在处理实数或复数时，我们可以通过定义 2.15 从 $<$ 构造符号函数，并通过符号函数构造序关系，方法如下

$$x < y \iff \operatorname{sign}(\operatorname{sign}(y - x) \cdot (2 + \operatorname{sign}(x - y))) = 1.$$

然而，如果现在考虑有限整环，那么如何从符号函数构造 $<$ 关系就突然变得不那么明确了，而反过来仍然可行，依据定义 2.15。

鉴于我们将电路和逻辑片段定义为相对于某个序而言，让二者都能使用该序是很自然的。因此，我们选择使用 $<$ 门，而非符号门。因此，下文中所有对符号的使用，都隐含地使用了定义 2.15 所述的序关系。

在其他文献所讨论的情形（$\mathbb{R}$ 或 $\mathbb{F}_2$）中，这不会产生复杂性理论影响，因为在使用一种形式体系模拟另一种形式体系时，电路规模的开销是线性的，而深度的开销是常数级的。

为了定义关于代数电路的复杂性类，我们必须定义此类电路所计算的函数，并定义电路族这一术语。

定义2.21。由一个 R-电路C （具有 n 个输入门和 m 个输出门）计算的（n元）函数 $f_C\colon\ R^n \to R^m$ 定义为

$$f_C(x_1, \ldots, x_n) = (y_1, ..., y_m),$$

其中， y_1、.....、 y_m 是将 x_1、.....、 x_n 作为输入时， C的输出门的值。$\triangleleft$

定义2.22。 R-电路族 $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ 是一个电路序列，其中每个输入长度 $n \in \mathbb{N}$对应一个电路。由电路族 $\mathcal{C}$ 计算的函数 $f_{\mathcal{C}}\colon\ R^* \to R^*$定义为

$$f_{\mathcal{C}}(x) = f_{C_{|x|}}(x)$$

电路族 $(C_n)_{n\in\mathbb{N}}$ 的规模（分别为深度）定义为一个函数，它将$n \in \mathbb{N}$ 映射到 C_n的规模（分别为深度）。$\triangleleft$

与经典情形类似，如果 $\mathcal{C}$ 能够计算集合 S的特征函数，我们就说集合 $S \subseteq R^*$ 可由电路族 $\mathcal{C}$判定。

定义2.23。类 AC^i_R 是由规模为 $\mathcal{O}(n^{\mathcal{O}(1)})$、深度为 $\mathcal{O}((\log_2 n)^i)$的 R-电路族可判定的集合类。◁

定义2.24。NC^i_R 类是由大小为 R-电路族 $\mathcal{O}(n^{\mathcal{O}(1)})$且深度为 $\mathcal{O}((\log_2 n)^i)$的电路所判定的集合类，其中每个算术门的入度上界为2（我们称之为有界扇入）。◁

备注2.25。我们刚刚引入的电路族并未对给定输入长度时构造任意单个电路的难度作出限制。如果需要了解获得特定电路的难度，可以考虑所谓的一致电路族。这些电路族要求其电路满足关于构造难度的限制。关于一致性的更多信息，参见 [第99]卷。

代数电路类的一致性准则也可以用类似方式定义。例如参见 [第+98节。

2.2. 整环上的结构与一阶逻辑。由于我们希望用逻辑片段刻画电路复杂性类，这项工作总体上属于有限模型理论的范畴，尤其属于描述复杂性领域。相关主题的基础知识被视为已知，参见Gr¨adel等人、Libkin和Immerman所著书籍 [Gr¨a+07; Lib04；Imm99]。传统上，描述复杂性被视为有限模型理论的一个子学科，因为允许无限结构通常会使问题变得不可判定。然而，在我们的研究设置中，我们希望将无限结构（谨慎地）重新引入推理。为此，我们使用元有限模型论；它是对有限模型形式体系的扩展，最早由Gr¨adel和Gurevich引入 [GG98]。Gr¨adel等人所著书籍中也对元有限模型论作了简短介绍 [Gr¨a+07，见第 210]页。Gr¨adel和 Meer [GM95] 采用了这种方法来描述实数集上的一些基本复杂性类。随后，Cucker 和 Meer [CM99], 扩展了这些描述，其中包括定义实数集上的NC 层级。为进行刻画，这些论文引入了所谓的带算术的一阶逻辑，我们将对其进行调整，以便在我们的研究设置中使用。

为使证明更容易，我们采用一个众所周知的技巧：仅由函数组成的签名可以通过特征函数来模拟我们可能想要定义的任意谓词，该特征函数对应于该谓词所解释的关系。此外，我们还可以用零元函数模拟常元。

定义 2.26. 设 L_s, L_f 是仅包含函数符号的有限词汇表。签名为 $\sigma = (L_s, L_f)$ 的 R-结构是一个有序对 $\mathcal{D} = (\mathcal{A}, F)$，其中

(1) $\mathcal{A}$ 是词汇表为 L_s 的有限结构，我们称之为 $\mathcal{D}$ 的骨架；其论域 A 称为 $\mathcal{D}$ 的论域，其基数记为 $|A|$。(2) F 是一个有限集，其中包含形如 X: $A^k \to R$ 的函数（$k \in \mathbb{N}$），这些函数解释 L_f 中的函数符号。

我们用 $\mathrm{STRUC}_R(\sigma)$ 表示所有签名为 σ 的 R-结构构成的集合，并假设对于任意签名为 $\sigma = (L_s, L_f)$ 的结构， L_s 和 L_f 中的符号都是有序的。◁

注 2.27。本文仅考虑有序结构，即骨架有序的结构。

定义2.28（ R上的一阶逻辑）。整环 R 上的一阶逻辑语言针对每个签名 $\sigma = (L_s, L_f)$都包含一组公式和

数项。数项分为索引项和数项：索引项的取值位于骨架的宇宙中，数项的取值位于 R中。这些项按如下方式归纳定义：

(1) 索引项集合定义为变量集合 V在 L_s的函数符号作用下的闭包。(2) R 的任意元素都是数项。(3) 对于索引项 $h_1, ..., h_k$ 以及一个 k 元函数符号 $X \in L_f$, $X(h_1, ..., h_k)$ 是数项。(4) 如果 t_1, t_2 是数项，那么 $t_1 + t_2, t_1 \times t_2$ 和 $\mathsf{sign}(t_1)$ 也都是数项。

原子公式是索引项 $h_1 = h_2$ 和数项 $t_1 = t_2$ 的等式、数项 $t_1 < t_2$ 的不等式，以及形如 $P(h_1, ..., h_k)$ 的表达式，其中 $P \in L_s$ 是一个 k 元谓词符号， $h_1, .., h_k$ 是索引项。

集合 FO_R 是包含原子公式在逻辑联结词 $\{\wedge, \vee, \neg, \rightarrow, \leftrightarrow\}$ 和量化 $\exists v\psi$ 与 $\forall v\psi$ 下闭包的最小集合，其中 v 的取值范围为 $\mathcal{A}$。$\lhd$

为便于阅读，本文将始终使用形如 $x \leq y$ 的不等式，以及将等式$\overline{x} = \overline{y}$ 和不等式 $\overline{x} < \overline{y}$ 、 $\overline{x} \leq \overline{y}$ 推广到元组的形式。注意，这些推广形式很容易由一阶逻辑中的 $=$ 和 $<$ 定义。

注 2.29。我们将 R 中的任意元素称为数项，尽管某些整环可能包含 ζ、 $\aleph$ 或 $\circledast$ 之类并非数的元素。由于我们想对这些元素进行一些计算，因此也将它们称为数。

一阶逻辑$_R$公式与由一阶逻辑$_R$公式定义的集合之间的等价性，按照通常方式定义；也就是说，公式 φ 定义集合 S ，当且仅当 S 中的元素恰好是使 φ 成立的 R结构的编码；而两个这样的公式等价，当且仅当它们定义相同的集合。

为了构造一种能够定义由无界扇入电路判定的集合的逻辑，我们引入了构造数项的新规则：求和规则和乘积规则。我们还将定义另一条规则，称为最大化规则。不过，这条规则已经可以在一阶逻辑$_R$ 中定义，因此使用它并不会增强表达能力。我们将利用它来证明，特征函数可以在一阶逻辑$_R$中表示。

定义 2.30（求和、求积与最大化规则）。设 t 是一个数项，其中变量 $\overline{x}$ 和变量 $\overline{w}$ 均自由出现，并设 A 表示给定输入结构的论域。那么

$\mathbf{sum}_{\overline{x}}(t(\overline{x}, \overline{w}))$ 也属于数项，其解释为 $\sum_{\overline{x} \in A^{|\overline{x}|}} t(\overline{x}, \overline{w})$。数项 $\mathsf{prod}_{\overline{x}}(t(\overline{x}, \overline{w}))$ 和 $\mathsf{max}_{\overline{x}}(t(\overline{x}, \overline{w}))$ 的定义类似。

我们称这些算子为聚合器；对于任何包含上述形式聚合器的公式 φ， $\overline{x}$ 中的变量均视为在 φ 中受约束。$\lhd$

例 2.31。设 $\sigma = (\{\}, f_E^2)$ 为加权图的签名，也就是说， $f_E(x, y)$ 给出从 x 到 y 的边的权重；如果不存在这样的边，则为 0。设 $\mathcal{G}$ 是 σ上的一个（图）结构。那么，以下 $\mathrm{FO}_R + \mathrm{SUM}_R$ 句子 φ 表示： $\mathcal{G}$的骨架中存在一个节点，其出边权重之和大于任何其他节点出边权重之和的两倍。

$$\varphi := \exists x \forall y (x \neq y \rightarrow \mathbf{sum}_a(f_E(x, a)) > 2 \times \mathbf{sum}_b(f_E(y, b)))$$

观察 2.32。$\mathrm{FO}_R = \mathrm{FO}_R + \mathrm{MAX}_R$。

证明。$\max_i(F(i))$ 的一次出现实质上保证存在一个元素x，使得对于所有元素 y, $F(x) \geq F(y)$，并取值为 $F(x)$。显然，可以用形式为 $\forall x \exists y F(x) \geq F(y)$ 的一阶逻辑公式来定义这一点。□

此外，任何逻辑公式的特征函数都可以用 FO_R 描述。其证明与 Cucker 和 Meer [CM99，命题 2]，的证明类似，因为该证明没有用到实数的任何特殊性质。

3. $\mathrm{AC}^0_R = \mathrm{FO}_R$

本节中，我们证明 FO_R 刻画了 AC^0_R 类。证明思路类似于 Immerman 提出的已确立结果的证明，该结果通过 FO 刻画 AC^0 。

使用逻辑来检验给定的 R-元组是否会被某个R-电路接受时，需要考虑如何将这个 R-元组解释为一个R-结构 $\mathcal{A}$。可以将其解释为该电路输入门的集合，以及一个一元函数 f_{element}: $A \to R$ 将第 i 个输入门映射到电路的第 i 个输入。我们将这种结构称为电路结构。

下面我们将通过添加输入结构中未给出的函数和关系，扩展一阶逻辑$_R$ 。为此，我们对定义 R做少许补充，该定义中我们给出了R-结构的定义。今后谈到签名为(L_s, L_f)的L_s-结构时，我们也将考虑签名形如(L_f, L_a, L_a)的结构。额外的（同样有序的）词汇表 R不会对 R-结构产生任何影响，但其中包含可在具有该签名的逻辑公式中使用的函数符号。这意味着，对于任意词汇表L_s，任何签名为(L_f, R)的L_s-结构，也是签名为(L_f, L_a, L_a)的 L_a -结构。$_R$ 中的符号表示一些函数，我们将利用这些函数扩展一阶逻辑的表达能力，以刻画各种复杂性类。

定义 3.1。设 F 为一个有限函数集合。我们用 $\mathrm{FO}_R[F]$ 表示这样一个集合类：其中的集合可以由一阶逻辑$_R$句子定义，并且这些句子除了可以使用结构中给出的函数外，还可以使用 F 中的函数。◁

形式上，这意味着一阶逻辑$_R[F]$ 恰好描述那些集合 $S \subseteq R^*$ ： 存在一个签名为 $\sigma = (L_s, L_f, L_a)$的一阶逻辑$_R$句子 φ，使得对每个长度 n，都存在一种解释 I_n ，将 L_a 中的符号解释为 F 的元素，并且对于所有长度为 n 的 R^*元组 s ，当且仅当 $s \in S$ 编码了一个基于(L_s, L_f, L_a)的 R结构，且该结构在使用 I_n时满足 φ。

定义 3.2。我们用 Arb_R 表示所有函数 f: $R^k \to R$的集合，其中 $k \in \mathbb{N}$。◁

定理 3.3。设 R 是一个无限整环。那么 $\mathrm{AC}^0_R = \mathrm{FO}_R[\mathrm{Arb}_R] + \mathrm{SUM}_R + \mathrm{PROD}_R$。

证明。该定理的证明与 $\mathrm{FO}_{\mathbb{R}}[\mathrm{Arb}_{\mathbb{R}}] + \mathrm{SUM}_{\mathbb{R}} + \mathrm{PROD}_{\mathbb{R}} = \mathrm{AC}^0_{\mathbb{R}}$ [BV21], 的构造类似，因为这一构造并未利用实数的任何特殊性质。

这一证明的基本思路是，首先证明，对于任意 $\mathrm{FO}_R[\mathrm{Arb}_R] + \mathrm{SUM}_R + \mathrm{PROD}_R$-句子 φ，我们都可以构造一个电路族，当且仅当输入编码了一个满足 φ 的 R-结构时，该电路族才接受此输入。基本做法是利用可用的

门类型，并逐层计算公式，以此模仿 的逻辑算子的行为。例如，像 $\forall x\varphi(x)$ 这样的全称量词，是通过在一个乘法门上方使用一个符号门来实现的（该符号门按照定义 2.15 从 $<$ 得到）；这个乘法门的前驱是针对编码结构骨架中每个 a 的 $\varphi(a)$ 所对应的电路。

要将一阶逻辑$_R$ 的语义转换为电路，我们基本上可以沿用实数情形证明中的转换方法，因为其中大部分只用到了整环的性质。对于这些转换中的大多数，我们需要环性质；尤其是对于全称量词，我们还需要乘法交换性且不存在零因子，因此要求底层结构为整环。

不过，我们确实需要更改存在量词和 $\vee$的转换。在实数情形中， $\exists x\varphi(x)$的电路构造方式与全称量词的情形类似：先接一个符号门，再接一个加法门，并将骨架中每个 a对应的 $\varphi(a)$电路作为其前驱。由于存在特征大于 0 的无限整环，也就是说，在这些整环中，将正数个 1 相加可能得到 0，因此这种做法并不总能产生所需结果。不过，我们可以轻松克服这一问题：将 $\exists x\varphi(x)$转换为 $\neg\forall x\neg\varphi(x)$，并将 $x\vee y$转换为 $\neg(\neg x\wedge\neg y)$，因为否定、全称量词和 $\wedge$ 都可以利用整环的性质构造。

对于反向包含，给定一个电路族 $(C_n)_{n\in\mathbb{N}}$ 时，会构造一个数项 $\mathrm{val}_d(g)$，使得对每个 n，$\mathrm{val}_d(g)$ 的求值结果等于门 g 的值，前提是 g 位于相应电路 C_n的第 d 层。对于这一构造，在整环情形下，无需对实数情形中的公式作任何修改。□

4. Algebraic Circuits and Guarded Functional Recursion

在本节中，我们将 Durand、Haak 和 Vollmer [DHV18] 对 NC^1 和 AC^1 的逻辑刻画推广到整环上的相应复杂性类 NC^1_R 和 AC^1_R。此外，我们还将这一方法扩展到刻画整个 NC_R 和 AC_R 层次。

在他们的论文中，Durand、Haak 和 Vollmer 使用他们称为守卫谓词递归的方法扩展一阶逻辑，以刻画对数深度电路复杂性类 NC^1 和 AC^1。这本质上相当于一个递归算子：它将每个递归步骤中传递下来的变量元组（在其数值解释下）折半。这确保了总递归深度至多为对数级。变量元组的折半是通过以下方式利用这样一个事实来完成的：如果 BIT 和 $<$ 可表达，则加法在一阶逻辑中可表达 [Imm99]

$$x \leq \frac{y}{2} \iff x + x \leq y.$$

请注意，我们不将折半的公式定义为等式，因为对于奇数来说这是不可能的。不过，这并不成问题，因为我们只想界定最坏情况递归深度。为了刻画具有多对数深度的类，我们希望找到一个合适的因子来替换 $1\ 2$，使递归过程也具有多对数深度。事实证明，对于任意 $i \in \mathbb{N}$，通过使用因子 $2^{-\frac{}{\text{对数}_2 n\,(\text{对数}_2 n)^i}}$，我们可以确保递归深度为 $\mathcal{O}((\text{对数}_2 n)^i)$。

$\frac{}{\text{对数}_2 n}$观察 4.1. 任意数 $n \in \mathbb{N}$ 在达到 1 之前，都可以恰好乘以因子 $2^{\frac{}{(\text{对数}_2 n)^i}}$，共 $(\text{对数}_2 n)^i$ 次。

Proof. $n \cdot \left(2^{-\frac{\log_2 n}{(\log_2 n)^i}}\right)^{(\log_2 n)^i} = n \cdot 2^{-\frac{\log_2 n}{(\log_2 n)^i}\cdot(\log_2 n)^i} = n \cdot 2^{-\log_2 n} = n \cdot \frac{1}{n} = 1$ □

这一观察的一个更一般版本可见于附录中的引理 A.2。遗憾的是，虽然在 BIT 谓词

444	244	144	044
442	242	142	042
441	241	141	041
440	240	140	040
424	224	124	024
422	222	122	022
421	221	121	021
420	220	120	020
414	214	114	014
412	212	112	012
411	211	111	011
410	210	110	010
404	204	104	004
402	202	102	002
401	201	101	001
400	200	100	000

FIGURE 1. 对五进制数 444 逐位除法直到得到 0 的示意图，该过程需要 $63 = (\lfloor \log_2 5 \rfloor + 2)^3 - 1$ 步。

$\log_2 n$可用时，除以 2 很简单，但是否能在一阶逻辑中乘以诸如 $2^{\overline{(\log_2 n)^i}}$ 这样的因子，则完全不明朗。

不过，我们可以利用除以 2 的能力来实现多对数递归深度：不再对一个数进行除法，而是实质上逐位单独除以进制为 n 的数，并在结果达到 0 时进行进位。

例如，考虑五进制数 444。前面提到的过程如图 1 的表格所示。表格应按从上到下、再从左到右的顺序阅读。

我们从最低位到最高位依次将 444 的各位除以 2，直到得到 0。因此，第一步是将 444 最右侧的一位除以 2，使 444 变为 442。再进行两次同类除法后，我们得到 0；在接下来的一步中，我们重置最右侧的一位，并将第二位除以 2。这与倒数的过程类似：不同之处在于，每一步不是减去 1，而是除以 2，并以类似的方式进位。值得注意的是，从 444 得到 000 需要 $63 = (\lfloor \log_2 5 \rfloor + 2)^3 - 1$ 步。这并非巧合：很容易证明，对于任意进制为 n 、包含 i 位的数，这类过程在少于 $(\lfloor \log_2 n - 1 \rfloor + 2)^i - 1$ 步后即可到达最小可能元素。

为了执行前文所述的除以 2 的操作，我们需要能够在一阶逻辑中表达 BIT 谓词。在包含自然数的经典情形下，BIT 的定义如下；其中，我们假设 j 的二进制表示中最高位的位索引为 1：

定义 4.2。令关系 $\mathrm{BIT}^2 \subseteq \mathbb{N} \times \mathbb{N}$ 定义如下：

$$\mathrm{BIT} := \{(i, j) \mid j\text{的二进制表示中的第 } i\text{位为 } 1,\ \ i,\ \ j \in \mathbb{N}\}. \quad \lhd$$

由于我们希望从逻辑上刻画由电路族判定的语言，因此有必要简要讨论描述复杂性中的数的表示。在描述复杂性中，有限有序论域 A 中的元素元组通常与数相关联。这通常通过将元组解释为一个以 $|A|$ 为进制的数来实现，其中元组中的每个元素表示其在 A的序中的位置。

例4.3。例如，设 $D = \{a, b, c, d\}$ 按如下方式排序： $a < b < c < d$。那么，元组（b， d，c， a） $=$（1，3，2，0）将被解释为四进制数1320，对应于十进制数60。

每当我们谈到元组的数值解释时，指的都是这种解释。在此设定下，我们按照自身需要调整 BIT 的定义如下；同样， j 的二进制表示中，最高有效位是索引为 1 的位：

定义 4.4。对于任意论域为 D 的有序结构，关系 $\mathrm{BIT}^2 \subseteq D^* \times D^*$ 定义如下：

$$\mathrm{BIT} \coloneqq \{(i,j) \mid \text{ when } i \text{ and } j \text{ are taken as their numerical interpretations, the } i\text{th bit of the binary representation of } j \text{ is } 1,\ i,j \in D^*\} \quad \lhd$$

有了 BIT 谓词和序关系，我们现在就能用一阶逻辑表达除以 2。我们利用这样一个事实：只要具备 BIT 和序关系，就能表达元组 [Imm99] 的数值解释之间的乘法和加法。这个结果已针对普通一阶逻辑证明；由于忽略次要分量时，我们的双排序一阶逻辑 FO_R 等价于一阶逻辑，因此这里也可以应用该结果。

我们如下表示除以2：

$$\overline{x} \leq \overline{y}/2 \iff \exists \overline{z}\ \overline{x} + \overline{x} = \overline{z} \wedge \overline{z} \leq \overline{y}$$

再次注意，由于元组的数值解释是自然数，因此表示 $\overline{x} \leq \overline{y}/2$ 已经是我们所能做到的最好方式，因为 $\overline{x} = \overline{y}/2$ 对奇数不起作用。

接下来，我们将定义本节开头提到的递归算子。首先，我们需要一些附加记号，即相对化聚合器。聚合器的相对化是一个公式，用于限定聚合器所考虑的元素。

记号 4.5。对于数项 t 和一阶逻辑（FO）$_R$ 公式 φ，我们写作

$$\mathbf{max}_{\overline{x}}.(\varphi(\overline{x}))t(\overline{x})$$

作为以下表达式的简写

$$\mathbf{max}_{\overline{x}}(\chi[\varphi(\overline{x})] \times t(\overline{x})).$$

类似地，我们写作

$$\mathbf{sum}_{\overline{x}}.(\varphi(\overline{x}))t(\overline{x})$$

for

$$\mathbf{sum}_{\overline{x}}(\chi[\varphi(\overline{x})] \times t(\overline{x}))$$

and

$$\mathbf{prod}_{\overline{x}}.(\varphi(\overline{x}))t(\overline{x})$$

作为.....的简写

$$\mathbf{prod}_{\overline{x}}(\chi[\varphi(\overline{x})] \times t(\overline{x}) + \chi[\neg\varphi(\overline{x})] \times 1).$$

现在我们定义 GFR^i_R 算子，以及通过形如$\mathcal{L}$ +GFR_R的 GFR^i_R 扩展的逻辑。其思路是模仿图 1 所展示的行为。

定义 4.6（GFR^i_R）。设 F 是一组函数，使得 BIT 可用一阶逻辑$_R[F]$定义；并设 $\mathcal{L}$ 是一阶逻辑$_R[F]$，或是通过加入某些构造规则（例如定义 2.30 中的求和规则或乘积规则）扩展一阶逻辑$_R[F]$而得到的逻辑。

对于 $i \geq 0$， $\mathcal{L}+\text{GFR}_R^i$ 上关于 σ 的公式集合，是由 $\mathcal{L}$ 关于 σ 的语法通过以下规则扩展所得的公式集合：

$$\varphi := [f(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}) \equiv t(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}, f)]\psi(f),$$

其中， ψ 是一个 $\mathcal{L}$ 公式， f 是一个函数符号， t 是一个 $\mathcal{L}$ 数项，其自由变量为 $\overline{x}$、$\overline{y_1}$、.....、$\overline{y_i}$、$\overline{y_{i+1}}$；并且对于 $1 \leq j \leq i$，所有 $\overline{y_j}$ 都包含相同数量（且为正数）的变量，而每个涉及符号 f 的（子）数项在 t 中

(1) 的形式为 $f(\overline{x}, \overline{z_1}, \ldots, \overline{z_i}, \overline{z_{i+1}})$，其中 $\overline{z_1}, \ldots, \overline{z_i}, \overline{z_{i+1}}$ 位于受保护的聚合的作用域内

$$A_{\overline{z_1},\ldots,\overline{z_i},\overline{z_{i+1}}}. \left(\bigvee_{j=1}^{i} \left(\overline{z_j} \leq \overline{y_j}/2 \wedge \bigwedge_{k=1}^{j-1} \overline{z_k} \leq \overline{y_k} \right) \wedge \xi(\overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}, \overline{z_1}, \ldots, \overline{z_i}, \overline{z_{i+1}}) \right)$$

其中 $A \in \{\text{max,sum,prod}\}$, $\xi \in \mathcal{L}$ 满足 ξ 不包含输入结构中给定关系的任何关系符号，并且

(2) 不会出现在以这种方式受保护的任何聚合（或量化）的作用域内。

函数符号 f 在以下内容中被视为有界

$$\varphi := [f(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}) \equiv t(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}, f)]\psi(f).$$

GFR_R^i 算子的语义定义如下：设 φ 是一个关于 σ 的 $\mathcal{L}+\text{GFR}_R^i$ 公式，其中仅出现一次 GFR_R^i

$$[f(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}) \equiv t(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}, f)]\psi(\overline{z}, f)$$

并且 $\mathcal{D} \in \text{STRUC}_R(\sigma)$。于是，应用于公式 ψ的算子，即 $[f(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}) \equiv t(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}, f)]$,, 以下述方式定义了 f在 ψ中的解释：对于 $\mathcal{D}$的论域 D中的所有元素元组 $\overline{a}, \overline{b_1}, \ldots, \overline{b_i}, \overline{b_{i+1}}$，它们分别与 $\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}$具有相同的元数，

$$f(\overline{a}, \overline{b_1}, \ldots, \overline{b_i}, \overline{b_{i+1}}) = t(\overline{a}, \overline{b_1}, \ldots, \overline{b_i}, \overline{b_{i+1}}, f).$$

这意味着，公式 $[f(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}) \equiv t(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}, f)]\psi(\overline{c}, f)$对与 $\overline{z}$ 具有相同元数的元组 $\overline{c} \in D^{|\overline{z}|}$ 成立，当且仅当 $\mathcal{D} \models \psi(\overline{c}, f_I)$成立，其中 f_I 是由 GFR_R^i 算子定义的 f 的解释。含有多个 GFR_R^i 算子的公式的语义以类似方式定义。

注意，第 $(i+1)$ 个元组不受受保护的聚合限制。

◁

定义了 GFR_R^i 算子后，还需证明它确实能以我们所期望的方式保证多对数递归深度。

引理 4.7. 设 $\mathcal{D} \in \text{STRUC}_R(\sigma)$，且 φ 是一个包含 GFR_R^i 算子的公式。那么，GFR_R^i 算子的递归深度上界为 $\mathcal{O}((\log_2 n)^i)$，其中 n 是 $\mathcal{D}$ 的论域大小。

证明。设 n 是对 GFR_R^i 算子进行解释的结构的论域大小。GFR_R^i 算子的递归深度界来自对聚合变量加以保护的相对化。设 $f(\overline{x}, \overline{z_1}, \ldots, \overline{z_i}, \overline{z_{i+1}})$ 是 GPR^i 算子的一个出现。那么，$\overline{z_1}, \ldots, \overline{z_i}, \overline{z_{i+1}}$ 中的变量处于如下形式的受保护聚合的作用域内

$$A_{\overline{z_1},\ldots,\overline{z_i},\overline{z_{i+1}}}. \left(\bigvee_{j=1}^{i} \left(\overline{z_j} \leq \overline{y_j}/2 \wedge \bigwedge_{k=1}^{j-1} \overline{z_k} \leq \overline{y_k} \right) \wedge \xi(\overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}, \overline{z_1}, \ldots, \overline{z_i}, \overline{z_{i+1}}) \right)$$

按照定义 4.6。

令 z_j 为所有 $1 \leq j \leq i$ 的 $\overline{z_j}$ 的数值解释。我们将元组 $\overline{z_1}, \ldots, \overline{z_i}$ 解释为一个以 n 为进制的自然数 z，其中 z 的第 j 位数字为 z_j。

首先注意，在这种解释下，对递归调用中所用变量的相对化确保 z 在每一步都严格递减。大的合取项 $\bigvee_{j=1} \overline{z_j} \leq \overline{y_j}/2 \wedge \bigwedge_{k=1} \overline{z_k} \leq \overline{y_k}$ 确保存在一个下标 j，使得 $\overline{z_j} \leq \overline{y_j}/2$，这意味着 $\overline{z_j}$ 的数值解释至多是 $\overline{y_j}$ 数值解释的一半。它还确保所有下标较小的元组 $\overline{z_k}$（即，在将 $\overline{z_1}$、……、$\overline{z_i}$ 解释为进制为 n 的数时，数值更显著的元组）都不会增加。

由于每个元组 $\overline{z_j}$（按其数值解释）在达到 0 之前最多只能减半 $\lfloor \log_2 n \rfloor + 1$ 次，因此至多需要 $\lfloor \log_2 n \rfloor + 2$ 个递归步骤，才能使第 i 个以外的某个元组减半。最坏情况下，该元组是第（$i-1$）个。然后可以至多重复这一过程 $\lfloor \log_2 n \rfloor + 1$ 次，之后下一个较小下标处的元组才会减半。总的来说，最坏情况下，这一过程需要 $(\lfloor \log_2 n \rfloor + 2)^j$ 个递归步骤，第 $i-j$ 个元组才会减半。

这意味着，经过 $(\lfloor \log_2 n \rfloor + 2)^i - 1$ 个递归步骤后，第一个元组已达到 0。因此，递归深度的最大总值为 $(\lfloor \log_2 n \rfloor + 2)^i - 1 \in \mathcal{O}((\log_2 n)^i)$。

这一过程可以理解为对一个以 $\log_2 n$ 为进制的数进行倒数。其思想已在图 1 中进行了可视化。对于我们将在定义 4.12 中很快定义的一个序列，为了使用恰好这种过程，图 2 和图 3 还对其进行了明确说明。□

由于我们的最终目标是刻画 AC^i_R 和 NC^i_R，因此还需要定义能够模拟 NC_R 电路性质的聚合器。为此，我们引入有界聚合器，即相对化聚合器，其中我们只考虑满足相对化条件且大小最大的两个元素。

定义 4.8。我们定义有界聚合器 $\mathsf{sum}_{\overline{x},\ \mathrm{bound}}$ 和 $\mathsf{prod}_{\overline{x},\ \mathrm{bound}}$。它们的使用方式与定义 2.30 中先前定义的聚合器相同。其语义定义如下：

$$\mathsf{sum}_{\overline{x},\ \mathrm{bound}}\cdot(\varphi(\overline{x}))t(\overline{x},\overline{w}) \equiv$$
$$\mathbf{sum}_{\overline{x}}.(\varphi(\overline{x}) \wedge \forall \overline{y}\ \forall \overline{z}((\overline{y} \neq \overline{z} \wedge \overline{x} < \overline{y} \wedge \overline{x} < \overline{z}) \rightarrow (\neg\varphi(\overline{y}) \vee \varphi(\overline{z}))))t(\overline{x},\overline{w})$$

有界聚合器 $\mathsf{prod}_{\overline{x},\ \mathrm{bound}}$ 和 $\mathsf{max}_{\overline{x},\ \mathrm{bound}}$ 以类似方式定义。◁

借助这种有界聚合，我们现在可以定义一个稍弱版本的定义 4.6 中的受保护函数递归，称为有界受保护函数递归 $\mathsf{GFR}^i_{R,\ \mathrm{bound}}$。这样，我们便可以定义形如 $\mathsf{FO}_R[F] + \mathsf{GFR}^i_R$ 或 $\mathsf{FO}_R[F] + \mathsf{GFR}^i_{R,\ \mathrm{bound}}$ 的逻辑。

定义 4.9（$\mathsf{GFR}^i_{R,\ \mathrm{bound}}$）。如果满足定义 4.6 中的相同条件，但将（1）中的受保护的聚合替换为有界受保护的聚合，则公式属于 $\mathsf{FO}_R[F] + \mathsf{GFR}^i_{R,\ \mathrm{bound}}$。◁

以下我们的目标是使用一阶逻辑和受保护函数递归刻画 AC_R 和 NC_R 层级。为此，我们现在定义一个序列，稍后将把它作为门的数的一部分，以便将门的深度编码进其数中。构造该序列背后的思想是：对于深度有界于 $c \cdot (\log_2 n)^i$ 的电路族 $(C_n)_{n \in \mathbb{N}}$，序列中的每个元素本质上都是一个 i 位的进制 $c \cdot \lfloor \log_2 n \rfloor -$

1 数，其中每一位都采用一元编码，并用零填充。为便于阅读，我们将这种编码简称为一元编码。在这种解释下，可以将该序列看作倒数到 0。对于固定的 i，其长度将为 $c^i \cdot \lfloor \log_2 n \rfloor^i \in \mathcal{O}((\log_2 n)^i)$。我们首先介绍关于数的一元编码的约定。

定义 4.10. 设 $n, \ell \in \mathbb{N}$ 满足 $\ell \geq n$。那么，我们将函数 $\mathrm{unary}_\ell \colon \mathbb{N} \to \{0,1\}^\ell$ 定义如下

$$\mathrm{unary}_\ell(n) \coloneqq 0^{\ell - n} 1^n$$

称为 n 的（长度为 ℓ）一元编码。 ◁

定义 4.11. 对于任意形如以下形式的二进制字符串 $\overline{a}$

$$\overline{a} = 0^{k-m} 1^m$$

对于某些 k, m，我们定义函数 $\mathrm{uval} \colon \{0,1\}^\star \to \mathbb{N}$ 如下

$$\mathrm{uval}(\overline{a}) \coloneqq m$$

并称 $\mathrm{uval}(\overline{a})$ 为该一元编码的值 中 $\overline{a}$。 ◁

现在我们继续定义上述序列 d，之后将用它实质性地把电路中各门的深度编码到门编号中。

定义 4.12。对于每个 n、c、$i \in \mathbb{N}$，我们定义序列 $d(n, c, i)$ 如下。为便于阅读，我们在序列的定义中省略参数 (n, c, i)，只写 d，而不写 $d(n, c, i)$。

(1) d 中的每个元素 d_ℓ 由 i 个元组 $d_{\ell,j}$ $(1 \leq j \leq i)$ 组成，每个元组都是一个数在 $[0, \lfloor c \cdot \log_2 n \rfloor - 1]$ 中的一元编码，其长度为 $\lfloor c \cdot \log_2 n \rfloor - 1$。(2) $d_1 = 1 \ldots 1$（即，对于所有满足 $1 \leq j \leq i$ 的 j，$d_{1,j} = \mathrm{unary}_{\lfloor c \cdot \log_2 n \rfloor - 1}(c \cdot \lfloor \log_2 n \rfloor - 1) = 1_{\ell} \ldots 1$。）(3) 对于所有满足 $d_\ell \neq 0 \ldots 0$ 的 ℓ，若 $\mathrm{uval}(d_{\ell,i}) = 0$，则 $d_{\ell+1,i} = \begin{cases} \mathrm{unary}_{\lfloor c \cdot \log_2 n \rfloor - 1}(\lfloor c \cdot \log_2 n \rfloor - 1); \\ \text{否则为 } \mathrm{unary}_{\lfloor c \cdot \log_2 n \rfloor - 1}(\mathrm{uval}(d_{\ell,i}) - 1)。 \end{cases}$

(4) 对于 $j < i$，若 $\mathrm{uval}(d_{\ell,j+1}) = 0$，则 $d_{\ell+1,j} = \begin{cases} \mathrm{unary}_{\lfloor c \cdot \log_2 n \rfloor - 1}(\mathrm{uval}(d_{\ell,j} - 1)); \\ \text{否则为 } d_\ell \qquad\qquad ,j。 \end{cases}$ ◁

序列 d 的示例见图 2 和图 3。

备注 4.13。注意，在这种一元编码中减去数值 1，可以看作是在二进制中进行除以2的整数除法。稍后将受保护函数递归与 BIT 联系起来时，这一点会很有用。

注意，$d(8, 1, 2)$ 的长度（即元素的数目）为

$$9 = 1^2 \cdot \lfloor \log_2 8 \rfloor^2 (= c^i \cdot \lfloor \log_2 n \rfloor^i).$$

而 $d(8, 2, 2)$ 的长度为

$$36 = 2^2 \cdot \lfloor \log_2 8 \rfloor^2 (= c^i \cdot \lfloor \log_2 n \rfloor^i).$$

这并非巧合。接下来，我们将证明这一观察结果一般成立。

引理 4.14. 设 $n, c, i \in \mathbb{N}$。那么，$d(n, c, i)$ 的长度为 $c^i \cdot \lfloor \log_2 n \rfloor^i$。

ℓ \ i	1	2
1	11	11
2	11	01
3	11	00
4	01	11
5	01	01
6	01	00
7	00	11
8	00	01
9	00	00

FIGURE 2. 序列 $d(8, 1, 2)$。由于 $c \cdot \lfloor \log_2 8 \rfloor - 1 = 2$， $d(8, 1, 2)$ 中的每个元素都包含 $i = 2$ 个长度为 2 的元组。每一行对应序列中的一个元素，各列则决定了元素中的元组。这意味着，这里的第一个元素是 (11, 11)，第二个是 (11, 01)，依此类推。

证明。由于对于 $d(n, c, i)$ 中的每个元素 e，后继规则都可以解释为：当 e 被视为一个以 $c \cdot \lfloor \log_2 n \rfloor$ 为进制、具有 i 位的数时，从 e 中减去 1，因此我们从这种意义下可能的最大元素开始（即 $1 \ldots 1$，对应于 $(c \cdot \lfloor \log_2 n \rfloor)^i - 1$），并倒数至可能的最小元素（即 $0 \ldots 0$，对应于 0），所以 $d(n, c, i)$ 中恰有 $(c \cdot \lfloor \log_2 n \rfloor)^i = c^i \cdot \lfloor \log_2 n \rfloor^i$ 个元素。□

妨碍我们使用序列 d 为电路描述中的门编号的剩余问题是，d 中各元素的长度取决于 n（也就是电路的输入门数量）。不过，我们可以解决这个问题，因为在包含 n 个输入门的电路描述中，我们实际上可以使用进制为 n 的数（这是通过将电路输入解释为电路结构实现的）。将这些数与 BIT 谓词结合起来，并将 d 中元素的一元编码元组解释为二进制数，就可以用常数个位来编码 d 中的元素。

观察 4.15。设 n、$c \in \mathbb{N}$ 和 $1 \leq \ell \leq c \cdot \lfloor \log_2 n \rfloor$。数 $2^\ell - 1$ 可以编码为长度为 c的 n 进制数。

证明。该形式所能表示的最大数是 $2^{c \cdot \lfloor \log_2 n \rfloor} - 1 \leq n^c - 1$，它在 n进制下对应于“$\underbrace{n-1 \ldots n-1}$”。因此，$2^{c \cdot \lfloor \log_2 n \rfloor} - 1$ 可以用长度为 c 的 n 进制位编码c 次，所以所有更小的自然数也都可以用这种方式编码。□

因此，我们可以将元组的二进制赋值编码为 d 中的元素，并将其表示为长度为 c的 n 进制数。因此，d 中的每个元素都可以用长度为 c 的 n进制数（或 $i \cdot c$ 个 n 进制位）编码。

在继续使用序列 d 描述电路之前，我们还需要一个引理，它给出了 AC^i_R（分别为 NC^i_R）电路的一个有用性质。我们希望能够讨论门的深度，即某个门到电路输入门的距离。为此，我们将证明，对于所研究的电路族，存在这样的电路：对每个门 g，每条输入-g路径的长度都相同。

ℓ \ i	1	2
1	11111	11111
2	11111	01111
3	11111	00111
4	11111	00011
5	11111	00001
6	11111	00000
7	01111	11111
8	01111	01111
9	01111	00111
10	01111	00011
11	01111	00001
12	01111	00000
13	00111	11111
14	00111	01111
15	00111	00111
16	00111	00011
17	00111	00001
18	00111	00000
19	00011	11111
20	00011	01111
21	00011	00111
22	00011	00011
23	00011	00001
24	00011	00000
25	00001	11111
26	00001	01111
27	00001	00111
28	00001	00011
29	00001	00001
30	00001	00000
31	00000	11111
32	00000	01111
33	00000	00111
34	00000	00011
35	00000	00001
36	00000	00000

FIGURE 3. 序列 $d(8, 2, 2)$。由于 $c \cdot \lfloor \log_2 8 \rfloor - 1 = 5$，$d(8, 2, 2)$ 中的每个元素都包含 $i = 2$ 个长度为 5 的元组。每一行是序列中的一个元素，各列则确定这些元素中的元组。这意味着，此处的第一个元素是 (11111, 11111)，第二个是 (11111, 01111)，依此类推。

引理 4.16。设 L 属于 AC^i_R 或 NC^i_R，其电路族为 $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$。那么存在一个电路族 $\mathcal{C}' = (C'_n)_{n\in\mathbb{N}}$ 来判定 L，使得对于所有 $n \in \mathbb{N}$ 以及其中的每个门 g C'_n，从输入门到 g （位于 C'_n 中）的每条路径长度都相同。我们称 C'_n 为平衡有向无环图。

证明。设 L 属于 AC^i_R 或 NC^i_R，通过 $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$。对于每个电路 $C_n \in \mathcal{C}$，我们构造一个电路 C'_n，使得 $f_{C_n} = f_{C'_n}$，且对于 C'_n中的每个门 g，C'_n 中所有输入-g-路径的长度都相同。我们通过创建由虚设门构成的路径来替换跨越多个深度层级的边，从而将 C_n 转换为 C'_n。

设 C_n 的深度有界于 $c_1 \cdot (\log_2 n)^i$，且其规模有界于$c_2 \cdot n^{c_3}$。我们对 C_n中门 g的深度 d进行结构归纳法，即从输入门到 g的路径的最大长度。

$d = 1$：深度为 d 的每个门 g 都是输入门的直接后继。因此，无需进行任何更改，因为深度为 d 的所有门的输入-g路径长度都为 1，从而具备所需性质。

$d \to d + 1$：对于深度为 $d+1$ 的每个门，其所有前驱都是深度为 $< d+1$ 的门，并且从输入门到这些前驱的所有路径长度都相同。将深度为 d 的所有此类前驱保持原样，并将深度较小的前驱 $d' < d$ 到 g 的边替换为一条由长度为 $d - d'$ 的虚拟（一元加法）门组成的路径。这样，从输入门到 g 的所有路径长度都恰好为 $d + 1$，而且对于 g 的每个前驱，我们最多只添加了 $c_1 \cdot (\log_2 n)^i$ 个门。总计，为 g 添加的虚拟门数量至多为 $c_2 \cdot n^{c_3} \cdot c_1 \cdot (\log_2 n)^i$。

所得电路为 C'_n。由于只有一个前驱的加法门本质上是恒等门，因此在 C'_n 的任何一次计算中，每个门的值都保持不变。因此，$f_{C_n} = f_{C'_n}$。

此外，对于 C_n 中的每个门，我们最多添加 $c_1 \cdot (\log_2 n)^i \cdot c_2 \cdot n^{c_3}$ 个门，以得到 C'_n。因此，C'_n 的大小至多为 $c_1 \cdot (\log_2 n)^i \cdot c_2 \cdot n^{c_3} \cdot c_2 \cdot n^{c_3} = c_1 \cdot (\log_2 n)^i \cdot (c_2 \cdot n^{c_3})^2 \in \mathcal{O}(n^{\mathcal{O}(1)})$。$C'_n$ 的深度不变，因为我们只会在 C_n 中存在更长路径、因而这些路径可以被延长至相同长度时才添加门。

总之，C'_n 计算与 C_n 相同的函数——因此判定 L ——并且具有如下性质：对于其每个门 g，所有输入-g-路径都具有相同的长度。□

如前所述，每当我们处理平衡有向无环图时，我们将输入门到某个门 g 之间无歧义的长度称为 g的深度。

现在我们将介绍一个引理，借助它，我们最终能够使用此前定义的序列 d，将门的深度编码到其电路编号中。结合引理 4.14，这为我们提供了一种逻辑上确保给定为电路结构的电路具有多对数深度的方法。

引理 4.17。设 L 通过电路族 $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$属于 AC^i_R 或 NC^i_R。则存在一个 AC^i_R（分别为 NC^i_R）电路族 $\mathcal{C}' = (C'_n)_{n\in\mathbb{N}}$ 判定 L，其深度由 $c \cdot (\log_2 n)^i$界定，并且 $\mathcal{C}'$ 中每个电路的门都被编号为 n 进制的数；对于从输入门到输出门的每条路径，前 $c \cdot i$ 位按照元素在序列中的出现顺序编码 d 中的元素。

前 $c \cdot i$ 位之后的编号可以任意选择。

证明。设 $L \in \mathsf{AC}^i_R$ 或 $L \in \mathsf{NC}^i_R$ 可由电路族 $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ 判定，其深度被 $c_1 \cdot (\log_2 n)^i$ 所界。不失一般性，根据引理 4.16，设 $\mathcal{C}$ 的电路是平衡有向无环图。这意味着，对于 $\mathcal{C}$ 中的每个电路 C_n，以及 C_n 中的每个门 g，从输入门到 g 的所有路径长度都相同。令 $c \in \mathbb{N}$ 满足 $c^i \cdot \lfloor \log_2 n \rfloor^i$ 大于 C_n 的深度（该深度被 $c_1 \cdot (\log_2 n)^i$ 所界）。我们将 C_n 中的每条路径补齐至长度 $c^i \cdot \lfloor \log_2 n \rfloor^i$：把每条从输入门到 C_n 中某个门的边替换为一条由虚拟门构成、长度为 $c^i \cdot \lfloor \log_2 n \rfloor^i - \mathsf{depth}(C_n)$ 的路径，从而使所得电路的深度恰为 $c^i \cdot \lfloor \log_2 n \rfloor^i$。

现在，我们用任意的进制 n 对 C_n 中的门进行编号，并对每个门 g，将 $d(n,c,i)$ 的第 $\mathrm{depth}(g)$ 个元素前置到 g 的编号上。由于我们确保了 C_n 中每条输入到输出的路径长度恰为 $c^i \cdot \lfloor \log_2 n \rfloor^i$，因此可以在每条输入到输出的路径上的编号中准确编码序列 $d(n,c,i)$。因此，现在对于每条输入到输出的路径，门编号的前 $c \cdot i$ 位按其在序列中出现的顺序编码了 $d(n,c,i)$ 的各个元素。□

借助引理 4.17 的标准形式和前面的定义，我们现在可以将这些内容串联起来，进而给出一个从逻辑上刻画 AC^i_R 和 NC^i_R 的定理。

定理 4.18。

(1) $\mathsf{AC}^i_R = \text{一阶逻辑}_R[\mathsf{Arb}_R] + \mathsf{SUM}_R + \mathsf{PROD}_R + \mathsf{GFR}^i_R$(2)
$\mathsf{NC}^i_R = \text{一阶逻辑}_R[\mathsf{Arb}_R] + \mathsf{SUM}_R + \mathsf{PROD}_R + \mathsf{GFR}^i_{R,\ \mathrm{bound}}$

证明。我们首先证明电路类包含于相应的逻辑中，然后再证明反向包含关系。

步骤 1：$\mathsf{AC}^i_R \subseteq \text{一阶逻辑}_R[\mathsf{Arb}_R] + \mathsf{SUM}_R + \mathsf{PROD}_R + \mathsf{GFR}^i_R$:
令 L 是通过非一致电路族 $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$，并令 $\mathcal{C}$ 的深度由 $c\cdot(\log_2 n)^i$ 界定。我们构造一个定义 L 的 $\mathrm{FO}_R[\mathsf{Arb}_R] + \mathsf{SUM}_R + \mathsf{PROD}_R + \mathsf{GFR}^i_R$ 句子 φ。由于电路输入被解释为电路结构，φ 的签名 σ 仅包含单个一元函数符号 f_{element}。

我们定义以下附加关系和函数，它们将实质性地编码给定的电路。由于逻辑的 Arb_R 扩展，我们可以使用这些关系和函数；这里使用关系而不是函数是为了便于阅读，因为如果将关系各自的特征函数纳入考虑，在函数结构中我们实质上也可以使用关系。

- $G_+(\overline{x}) \iff \overline{x}$ 是一个加法门。
- $G_\times(\overline{x}) \iff \overline{x}$ 是一个乘法门。
- $G_<(\overline{x}) \iff \overline{x}$ 是一个 $<$门，其左前驱的字典序低于右前驱。

- $G_{\text{输入}}(\overline{x}) \iff \overline{x}$ 是一个输入门。
- $G_E(\overline{x},\overline{y}) \iff \overline{y}$ 是 $\overline{x}$的一个后继门。
- $G_{\text{输出}}(\overline{x}) \iff \overline{x}$ 是输出门。
- $G_{\text{常数}}(\overline{x}) \iff \overline{x}$ 是一个常数门。
- $f_{\text{常数值}}(\overline{x}) = y \in R \iff y$ 在 $\overline{x}$ 是常数门时取其值，否则取 $\overline{x}$。

不失一般性，设 $\mathcal{C}$ 的所有电路都处于引理 4.17 的标准形式，并设 C_n 的编号满足：对于 $1 \le j \le n$，第 j个输入门的编号的最后一位是 j。回想一下，这意味着，对于表示为元组 $\overline{g}$的门 g 的每个门编号，$\overline{g}$ 的前 $c\cdot i$ 个元素编码 $d(n,c,i)$ 的第 $\mathrm{depth}(g)$ 个元素。下面的句子 φ 定义了 L:

$$\varphi := [f(\overline{y}) \equiv t(\overline{y}, f)]\exists \overline{a}\ G_{\mathrm{output}}(\overline{a}) \wedge f(\overline{a}) = 1$$

其中 t 定义如下（$\overline{z_j}$ 表示 $\overline{z}$ 的 $c\cdot i$ 长前缀中的第 j 个长度为 c 的子元组；根据引理 4.17 的标准形式，它编码了第 j 个

$d(n, c, i)$ 中某个元素的元组）：

$$
\begin{aligned}
t(\overline{y}, f) \coloneqq \quad & \chi[G_{+}(\overline{y})] \times \mathbf{sum}_{\overline{z}}. \left(\bigvee_{j=1}^{i} \left(\overline{z_j} \leq \overline{y_j}/2 \wedge \bigwedge_{k=1}^{j-1} \overline{z_k} \leq \overline{y_k} \right) \wedge G_{\mathrm{E}}(\overline{y}, \overline{z}) \right) f(\overline{z}) + \\
& \chi[G_{\times}(\overline{y})] \times \mathbf{prod}_{\overline{z}}. \left(\bigvee_{j=1}^{i} \left(\overline{z_j} \leq \overline{y_j}/2 \wedge \bigwedge_{k=1}^{j-1} \overline{z_k} \leq \overline{y_k} \right) \wedge G_{\mathrm{E}}(\overline{y}, \overline{z}) \right) f(\overline{z}) + \\
& \chi[G_{<}(\overline{y})] \times \mathbf{max}_{\overline{z}}. \left(\bigvee_{j=1}^{i} \left(\overline{z_j} \leq \overline{y_j}/2 \wedge \bigwedge_{k=1}^{j-1} \overline{z_k} \leq \overline{y_k} \right) \wedge G_{\mathrm{E}}(\overline{y}, \overline{z}) \right) \\
& \qquad \left(\mathbf{max}_{\overline{b}}. \left(\bigvee_{j=1}^{i} \left(\overline{b_j} \leq \overline{y_j}/2 \wedge \bigwedge_{k=1}^{j-1} \overline{b_k} \leq \overline{y_k} \right) \wedge G_{\mathrm{E}}(\overline{y}, \overline{b}) \wedge \overline{b} < \overline{z} \right) \right. \\
& \qquad \left. \left(\chi[f(\overline{b}) < f(\overline{z})] \right) \right) + \\
& \chi[G_{\mathrm{input}}(\overline{y})] \times f_{\mathrm{element}}(y_{|\overline{y}|}) + \\
& \chi[G_{\mathrm{const}}(\overline{y})] \times f_{\mathrm{const\_val}}(\overline{y}) + \\
& \chi[G_{\mathrm{output}}(\overline{y})] \times \mathbf{sum}_{\overline{z}}. \left(\bigvee_{j=1}^{i} \left(\overline{z_j} \leq \overline{y_j}/2 \wedge \bigwedge_{k=1}^{j-1} \overline{z_k} \leq \overline{y_k} \right) \wedge G_{\mathrm{E}}(\overline{y}, \overline{z}) \right) f(\overline{z}).
\end{aligned}
$$

这里，关系 $G_g(\overline{y})$（对应 $g \in \{+, \times, <, \mathsf{input}, \mathsf{const}, \mathsf{output}\}$）提供了由 $\overline{y}$ 编码的门的类型信息，并由 FO_R 的 Arb_R-扩展提供。它们按上述方式解释。BIT 谓词也以相同方式提供。

现在我们将证明 φ 确实定义了 L。设 $\overline{a} \in R^n$ 是 C_n 的输入。我们将证明，对于所有 $\overline{g} \in R^l$，其中 l 是 $\mathcal{C}$ 中一个门的编码长度，当 C_n 以 $\overline{a}$ 作为输入时，在 C_n 的计算过程中，由 $\overline{g}$ 编码的门的值为 $f(\overline{g})$。设 g 是由 $\overline{g}$ 编码的门。我们将对门 g 的深度进行归纳，即依据 g 与一个输入门之间的距离进行归纳。

$d = 0$：若 $d = 0$，则 g 是一个输入门。因此，$t(\overline{g}, f)$ 中唯一不显然等于 0 的加项是第四项，它等于 $f_{\mathrm{element}}(g_{|\overline{g}|})$（即第 $g_{|\overline{g}|}$ 个输入门的值）。

$d \to d+1$：由于 $d + 1 > 0$，g 不是输入门。这意味着对于 g，有以下 5 种可能性：

(1) g 是一个加法门：在这种情况下，$t(\overline{g}, f)$ 中唯一不显然为 0 的被加数是第一个。由于引理 4.17 的标准形式，g 的所有前驱都有一个数，其前 $c \cdot i$ 位是序列 $d(n, c, i)$ 中g的前 $c \cdot i$ 位的后继。此外，相对化确保相应被加数中由 $\overline{z}$ 编码的门，其前 $c \cdot i$ 位恰好是序列 $d(n, c, i)$ 中g的前 $c \cdot i$ 位的后继。这意味着根据归纳法假设

$$
\mathbf{sum}_{\overline{z}}. \left(\bigvee_{j=1}^{i} \left(\overline{z_j} \leq \overline{y_j}/2 \wedge \bigwedge_{k=1}^{j-1} \overline{z_k} \leq \overline{y_k} \right) \wedge G_{\mathrm{E}}(\overline{y}, \overline{z}) \right) f(\overline{z})
$$

恰好等于 g 的所有前驱门的值之和。

(2) g 是一个乘法门：与上述情况类似，

$$\mathbf{prod}_{\overline{z}}.\left(\bigvee_{j=1}^{i}\left(\overline{z_j}\le\overline{y_j}/2\wedge\bigwedge_{k=1}^{j-1}\overline{z_k}\le\overline{y_k}\right)\wedge G_{\mathrm{E}}(\overline{y},\overline{z})\right)f(\overline{z})$$

恰好得到 g 的所有前驱门的乘积。

(3) g 是一个 $<$ 门：在这种情况下，相对化

$$\mathbf{max}_{\overline{z}}.\left(\bigvee_{j=1}^{i}\left(\overline{z_j}\le\overline{y_j}/2\wedge\bigwedge_{k=1}^{j-1}\overline{z_k}\le\overline{y_k}\right)\wedge G_{\mathrm{E}}(\overline{y},\overline{z})\right)$$

确保 $\overline{z}$ 是 g 的某个前驱门的最大门编号，并且

$$\left(\mathbf{max}_{\overline{b}}.\left(\bigvee_{j=1}^{i}\left(\overline{b_j}\le\overline{y_j}/2\wedge\bigwedge_{k=1}^{j-1}\overline{b_k}\le\overline{y_k}\right)\wedge G_{\mathrm{E}}(\overline{y},\overline{b})\wedge\overline{b}<\overline{z}\right)\left(\chi[f(\overline{b})<f(\overline{z})]\right)\right)$$

确保 $\overline{b}$ 是 g 的另一个前驱门的门编号，因此 $\chi[f(\overline{b})<f(\overline{z})]$ 是 $t(\overline{g},f)$ 的值，这恰好是 C_n 的计算中 g 的值。

(4) g 是一个常数门：由于 $\varphi_{\mathrm{const\_val}}(\overline{g})$ 恰好返回 g 的值，而该值由 $t(\overline{g},f)$ 取得。

（5） g 是输出门：在这种情况下， g 只有一个前驱，因此

$$\mathbf{sum}_{\overline{z}}.\left(\bigvee_{j=1}^{i}\left(\overline{z_j}\le\overline{y_j}/2\wedge\bigwedge_{k=1}^{j-1}\overline{z_k}\le\overline{y_k}\right)\wedge G_{\mathrm{E}}(\overline{y},\overline{z})\right)f(\overline{z})$$

确保 g 取该前驱的值，因为只有一个元素满足相对化条件。

最后，通过

$$\varphi\coloneqq[f(\overline{y})\equiv t(\overline{y},f)]\exists\overline{a}\;G_{\mathrm{output}}(\overline{a})\wedge f(\overline{a})=1$$

我们确保存在一个输出门，其在计算结束时的值为 1。

步骤 2：$\mathsf{NC}^i_R\subseteq\mathsf{FO}_R[\mathsf{Arb}_R]+\mathsf{SUM}_R+\mathsf{PROD}_R+\mathsf{GFR}^i_{R,\ \mathrm{bound}}$：
该包含关系的证明与 AC^i_R 情形的证明相同，只需将公式中的所有受保护聚合替换为有界受保护聚合。

步骤 3：$\mathsf{FO}_R[\mathsf{Arb}_R]+\mathsf{SUM}_R+\mathsf{PROD}_R+\mathsf{GFR}^i_R\subseteq\mathsf{AC}^i_R$：
设 $L\in\mathsf{FO}_R[\mathsf{Arb}_R]+\mathsf{SUM}_R+\mathsf{PROD}_R+\mathsf{GFR}^i_R$ 由签名$\sigma=(L_s,L_f,L_a)$上的某个公式 φ 表示，并且其中 GFR^i_R算子只出现一次。这个证明很容易推广到一般情形。这意味着 φ 具有如下形式

$$\varphi=[f(\overline{x},\overline{y_1},\ldots,\overline{y_i},\overline{y_{i+1}})\equiv t(\overline{x},\overline{y_1},\ldots,\overline{y_i},\overline{y_{i+1}},f)]\psi(f).$$

现在，我们构造一个 AC^i_R 电路族 $\mathcal{C}=(C_n)_{n\in\mathbb{N}}$ 来判定 L。

我们首先构造一个 AC^i_R 电路族，用于计算不含 f 的出现的 φ ，如定理 3.3 所示。这是可行的，因为 φ 是一个定义在 $(L_s,L_f,L_a\cup\{f\})$ 上的 $\mathsf{FO}_R[\mathsf{Arb}_R]+\mathsf{SUM}_R+\mathsf{PROD}_R$公式；根据定理 3.3，存在一个这样的电路族来计算 φ。

接下来，我们将解释如何从这里开始，为整个公式 φ构造一个 AC^i_R 电路族。为此，我们需要构造一个 AC^i_R 电路族，对满足 $\overline{a}\in A^k$ 的某些 $k\in\mathbb{N}$ 和 $\overline{b}\in A^i$的每一对 $\overline{a}$、$\overline{b}$ 计算 $f(\overline{x},\overline{y})$ 的值，其中元组 $\overline{b}$ 的长度恰好是限定电路深度的多对数的指数。注意， t 是一个 $\mathsf{FO}_R[\mathsf{Arb}_R]+\mathsf{SUM}_R+\mathsf{PROD}_R+\mathsf{GFR}^i_R$

数项，因此可以由一个 AC^0_R 电路族进行计算，但 f的出现除外。现在，我们通过从所有这些（数量为多项式个）AC^0_R 电路族中取第 n个电路，并对所有 $\overline{a}$、$\overline{b}$，将标记为 $f(\overline{a},\overline{b})$ 的门替换为计算 $t(\overline{a},\overline{b},f)$ 的电路的输出门，从而得到 C_n 。

由于所有 $f(\overline{x},\overline{y})$ 的出现都处于受保护的聚合的作用范围内

$$A_{\overline{z_1},\ldots,\overline{z_i}}.(\bigvee_{j=1}^{i}\overline{z_j}\leq\overline{y_j}/2\wedge\bigwedge_{k=1}^{j-1}\overline{z_k}\leq\overline{y_k}\wedge G_{\mathrm{E}}(\overline{y_1},\ldots,\overline{y_i},\overline{y_{i+1}},\overline{z_1},\ldots,\overline{z_i})),$$

根据引理 4.7，从任意 $f(\overline{a},\overline{b})$ 到达 $f(\overline{a},0)$（递归终止）之前的步数上界为 $\mathcal{O}((\log_2 n)^i)$。

由于每一步——在给定下一次递归调用 $f(\overline{c},\overline{d})$ 的值时计算 $f(\overline{a},\overline{b})$——都是由 AC^0_R 电路完成的，因此深度为常数。总之，从第一次递归调用到递归终止的任意路径长度均为 $\mathcal{O}((\log_2 n)^i)$。由于用于判定 φ 的起始电路深度为常数，我们构造的电路总深度为多对数级。并且，鉴于我们仅添加了多项式数量的子电路，且每个子电路都包含多项式数量的门，因此整个电路是一个用于判定 φ 的 AC^i_R 电路。

对于多个 GFR^i_R 算子的常见情况，我们以相同的方式为每个算子构造一个电路，并将它们连接到计算 φ 的电路。

步骤 4：$\mathsf{FO}_R[\mathsf{Arb}_R]+\mathsf{SUM}_R+\mathsf{PROD}_R+\mathsf{GFR}^i_{R,\mathrm{bound}}\subseteq\mathsf{NC}^i_R$:
这一情形可以用与 AC^i_R 情形类似的方法证明。我们不再使用用于计算 φ 和 t 的 AC^0_R 族，而是需要使用 NC^1_R 族。这样一来，计算 t 的深度为对数级；通常这会带来问题，因为重复执行该过程（$\log_2 n$）^{i} 次会得到一个 NC^{i+1}_R 族。然而，根据定义，f 不会出现在无界聚合器或量词的作用域内。对于有界聚合器，我们仍然为聚合变量的所有可能取值构造电路，但只连接满足相对化条件的两个最大电路的输出门。我们之所以能够这样做，是因为 ξ 不包含结构中给定的函数符号或关系符号，所以可以预先确定哪些电路符合相对化条件，并据此将这些连接硬连线。一般而言，不含无界量词或聚合器的 $\mathsf{FO}_R[\mathsf{Arb}_R]+\mathsf{SUM}_R+\mathsf{PROD}_R$ 公式可以在 NC^0_R 中求值。因此，标记为 f（$\overline{a}$，$\overline{b}$）的门只会在 t（$\overline{a}$，$\overline{b}$，f）的子电路中出现在常数深度处。这意味着，总体而言，该构造得到的深度为 $\mathcal{O}((\log_2 n)^i)$。
□

如前所述，受保护函数递归这一思想的基础，是用于普通一阶逻辑 [DHV18] 的守卫谓词递归。本文中针对 GFR_R 展示的将递归深度扩展至多对数递归深度的方法，也可应用于 GPR，从而得到以下结果。（GPR^i 的形式化定义见附录中的定义 B.2）。

推论 4.19。
(1) 一阶逻辑$[\mathsf{Arb}]+\mathsf{GPR}^i=\mathsf{AC}^i$(2) 一阶逻辑$[\mathsf{Arb}]+\mathsf{GPR}^i_{\mathrm{bound}}=\mathsf{NC}^i$

5. Relationship between versions of AC^0_R over different integraldomains

直观而言，判定 $\mathsf{AC}^0_{\mathbb{R}}$ 中某个问题的电路也应能够模拟判定例如 $\mathsf{AC}^0_{\mathbb{Z}}$ 和 $\mathsf{AC}^0_{\mathbb{Z}_2}$ 中问题的电路。此外，我们应能够通过模拟运算，用 $\mathsf{AC}^0_{\mathbb{Z}_2}$ 电路模拟 $\mathsf{AC}^0_{\mathbb{Z}_3}$ 电路，具体来说，模拟这些运算

即用 $\mathbb{Z}_2$ 的元组运算来模拟定义在整环 $\mathbb{Z}_3$ 中的运算。为了将这一想法形式化，我们提出了 $\mathfrak{C}$-模拟映射的概念。

定义 5.1. 设 $f_1, f_2\colon \mathbb{N} \to \mathbb{N}$ 是两个函数，并设 $\mathfrak{C}$ 是一个复杂性类，满足 $\mathfrak{C} \in \{\textsc{TimeSpace}(f_1, f_2)\ \textsc{SizeDepth}(f_1, f_2)\ \textsc{UnbSizeDepth}(f_1, f_2)\}$。

从整环 R_1 到整环 R_2 的 $\mathfrak{C}$-模拟映射是一个单射函数 $f\colon R_1^* \to R_2^*$，满足以下条件：

对于所有 $k \in \mathbb{N}_{>0}$ 和 $A \in \mathfrak{C}_{R_1^k}$，都存在一个 $\ell \in \mathbb{N}_{>0}$ 和一个语言 $B \in \mathfrak{C}_{R_2^\ell}$，使得对于所有满足 $x_i \in R_1^k$ 的 $\overline{x} = (x_1, x_2, \ldots, x_{|x|})$，对于所有 i：

$$\overline{x} \in A \iff f(\overline{x}) = \big(f(x_1), f(x_2), \ldots, f(x_{|x|})\big) \in B.$$

如果存在一个从整环 R_1 到整环 R_2的 $\mathfrak{C}$-模拟映射，我们也记作 $\mathfrak{C}_{R_1} \subseteq_{\mathrm{sim}} \mathfrak{C}_{R_2}$。关系 $=_{\mathrm{sim}}$ 和 $\subsetneq_{\mathrm{sim}}$ 的定义类似。$\lhd$

与经典复杂性理论中的归约形式体系类似，由 $\subseteq^{\mathfrak{C}}_{\mathrm{sim}}$ 诱导的关系具有自反性和传递性。由于本文主要关注 $\mathbf{AC}_R$ 和 $\mathbf{NC}_R$ 层级，我们将把讨论范围限制在这些复杂性类中。但请注意，我们展示的模拟方法可以直接推广到更大的复杂性类。

这一形式体系实质上将我们的整环划分为三层层级。第一层由有限整环上的复杂性类构成，第二层由可通过 $\mathbb{Z}$ 模拟的整环上的复杂性类构成，第三层则由可通过 $\mathbb{R}$ 模拟的整环上的复杂性类构成。在此框架下，某个整环上的复杂性类能够模拟同一层或更低层中整环上的复杂性类。为明确说明这一点，我们给出以下三个等式：

定理 5.2. 以下三个等式成立：

(1) 对所有素数幂 p、q 和 $i \in \mathbb{N}$，$\mathbf{NC}^i_{\mathbb{F}_p} =_{\mathrm{sim}} \mathbf{NC}^i_{\mathbb{F}_q}$。

(2) 对所有 $i \in \mathbb{N}$，$\mathbf{NC}^i_{\mathbb{Z}} =_{\mathrm{sim}} \mathbf{NC}^i_{\mathbb{Z}[j_k]}$。 (3) 对所有 $i \in \mathbb{N}$，$\mathbf{NC}^i_{\mathbb{R}} =_{\mathrm{sim}} \mathbf{NC}^i_{\mathbb{R}[j_k]}$。

证明。我们将证明分为两个主要部分：有限情形，其中整环的模拟是显然的；以及无限情形，在该情形中，我们必须更谨慎地处理模拟电路所执行的新运算，以确保电路所模拟的结构仍然是整环。

有限情形。对有限整环的模拟相当显然。对于 $\mathbf{NC}^i_{\mathbb{F}_p} \subseteq_{\mathrm{sim}} \mathbf{NC}^i_{\mathbb{F}_q}$ 其中 $p > q$，选择 $\mathbf{NC}^i_{\mathbb{F}_q}$-电路所使用的元组长度为满足 $p \leq q^k$ 的最小 $k \in \mathbb{N}$。将这 p 个元素映射到 $\mathbb{F}_q^k$ 中的前 p 个元素，并定义大小恒定的加法和乘法表，以模拟 $\mathbb{F}_p$中的加法和乘法。

无限情形。我们将证明，对于任意固定的 $k \in \mathbb{N}$，$\mathbf{NC}^i_{\mathbb{Z}} =_{\mathrm{sim}} \mathbf{NC}^i_{\mathbb{Z}[j_k]}$。请注意，以下证明不依赖于集合的可数性，因此不可数情形的证明也可类似进行。方向 $\mathbf{NC}^i_{\mathbb{Z}} \subseteq_{\mathrm{sim}} \mathbf{NC}^i_{\mathbb{Z}[j_k]}$ 显然成立。对于另一个方向，请注意，在模拟无限整环时，我们无法再以简单的方式将加法和乘法表硬编码进去。此外，整环对笛卡尔积并不封闭，因为对于两个整环 R_1、R_2，我们有，对于 $R_1 \times R_2$，使用逐分量乘法可得 $(1, 0) \times (0, 1) = (0, 0)$。我们通过仍然使用 k-元组来模拟伴随元素，并使用逐分量加法来解决这一问题，同时调整乘法，使其模拟两个具有伴随元素的元素相乘，其中

元组的第 m 个下标表示伴随元的 m 次幂，我们在此将其称为 j。具体而言，我们使用整环 $(\mathbb{Z}^k, +_{\mathbb{Z}^k}, \times_{\mathbb{Z}^k})$，其中 $+_{\mathbb{Z}^k}$ 是逐分量加法，而 $\times_{\mathbb{Z}^k}$ 定义如下：

如果我们要将两个长度为 k 的元组相乘

$$(x_1, x_2, \ldots, x_k) \times_{\mathbb{Z}^k} (y_1, y_2, \ldots, y_k),$$

我们就在原始的伴随整环中模拟乘法

$$\left(x_1 + x_2 j + \cdots + x_k j^{k-1}\right) \times_{\mathbb{Z}[j_k]} \left(y_1 + y_2 j + \cdots + y_k j^{k-1}\right),$$

方法是构造以下一个固定大小的矩阵，它对应于乘法的展开：

$$A = (a_{uv}) = \begin{pmatrix} x_1y_1 & x_1y_2j & \cdots & x_1y_kj^{k-1} \\ x_2y_1j & x_2y_2j^2 & \cdots & x_2y_kj^k (= -x_2y_k) \\ \vdots & \ddots & & \\ x_ky_1j^{k-1} & x_ky_2j^k (= -x_ky_2) & \cdots & x_ky_kj^{2k-2} \end{pmatrix}.$$

注意，由于 $j^k = -1$，矩阵的每个元素 a_{uv} 都会对元组中索引为 $(u + v - 2) \mod k$ 的项产生贡献。因此，所得元组中索引为 ℓ 的元素为

$$\sum_{\substack{1 \le u,v \le k \\ (u+v-2) \mod k = \ell}} a_{uv}. \qquad \square$$

定理 5.3。以下三个等式成立：

(1) 对所有素数幂 p、q 和 $i \in \mathbb{N}$，$\mathsf{AC}^i_{\mathbb{F}_p} =_{\mathrm{sim}} \mathsf{AC}^i_{\mathbb{F}_q}$。

(2) 对所有 $i \in \mathbb{N}$，$\mathsf{AC}^i_{\mathbb{Z}} =_{\mathrm{sim}} \mathsf{AC}^i_{\mathbb{Z}[j_k]}$。 (3) 对所有 $i \in \mathbb{N}$，$\mathsf{AC}^i_{\mathbb{R}} =_{\mathrm{sim}} \mathsf{AC}^i_{\mathbb{R}[j_k]}$。

证明。我们采用定理 5.2 证明中的策略，并说明无界加法和乘法可以在复杂度没有显著增加的情况下实现。

对于 n 给定的长度为 k 的元组

$$(x_{1,1}, x_{1,2}, \ldots, x_{1,k}), (x_{2,1}, x_{2,2}, \ldots, x_{2,k}), \ldots, (x_{n,1}, x_{n,2}, \ldots, x_{n,k}),$$

用无界逐分量元组加法来模拟无界加法是直接的。注意，对于无界乘法，简单地反复进行二进制乘法会导致线性深度。但与二进制乘法的方法类似，我们可以预先计算出哪些值会对结果元组中给定位置的值产生贡献。由于结果中元组项的值只能依赖于 $x_{i,\ j}$，因此，要模拟一个无界乘法门，我们只需要一个有线性个前驱的无界乘法门（但连线复杂度可能会呈指数级增加）。$\square$

推论 5.4. 对所有 $i \in \mathbb{N}$，我们有 $\mathsf{NC}^i_{\mathbb{R}} =_{\mathrm{sim}} \mathsf{NC}^i_{\mathbb{C}}$ 且 $\mathsf{AC}^i_{\mathbb{R}} =_{\mathrm{sim}} \mathsf{AC}^i_{\mathbb{C}}$。

证明。注意到 $\mathbb{C} = \mathbb{R}[j_{(2)}]$。 $\square$

引理 5.5. 对所有 $i \in \mathbb{N}$，我们有 $\mathsf{NC}^i_{\mathbb{Z}} =_{\mathrm{sim}} \mathsf{NC}^i_{\mathbb{Q}}$ 且 $\mathsf{AC}^i_{\mathbb{R}} =_{\mathrm{sim}} \mathsf{AC}^i_{\mathbb{C}}$。

证明。对于 $\mathsf{NC}^i_{\mathbb{Z}} \subseteq_{\mathrm{sim}} \mathsf{NC}^i_{\mathbb{Q}}$（分别为 $\mathsf{AC}^i_{\mathbb{Z}} \subseteq_{\mathrm{sim}} \mathsf{AC}^i_{\mathbb{Q}}$），取 $f(x) := x$；对于 $\mathsf{NC}^i_{\mathbb{Q}} \subseteq_{\mathrm{sim}} \mathsf{NC}^0_{\mathbb{Z}}$（分别为 $\mathsf{AC}^i_{\mathbb{Z}} \subseteq_{\mathrm{sim}} \mathsf{AC}^i_{\mathbb{Q}}$），取 $f(x) := (a, b)$，其中 $x = \frac{a}{b}$。$\square$

引理 5.6. $\mathsf{NC}^i_{\mathbb{Q}} \subsetneq \mathsf{NC}^i_{\mathbb{R}}$。

证明。我们利用这样一个事实： $\mathbb{R}$ 是 $\mathbb{Q}$的超越域扩张，也就是说，不存在一个有限的数集 M，将其添加到 $\mathbb{Q}$ 中便能得到 $\mathbb{Q}[M] = \mathbb{R}$。在我们的设定中，这意味着我们无法用由数构成的有限元组集合（a_0、$\cdots$、a_n）来模拟所有数 $r \in \mathbb{R}$，其中 a_0、$\cdots$、$a_n \in \mathbb{Q}$。□

6. Conclusion

在本文中，我们针对整环上的代数电路引入了代数复杂性类。我们利用 Durand、Haak 和 Vollmer [DHV18] 的 GPR算子的一种推广，给出了 AC^0_R 的逻辑刻画，以及 AC_R 和 NC_R 层级的进一步刻画。我们构造了一种形式体系，以便比较基于不同底层整环的复杂性类的表达能力。随后我们证明，利用这一形式体系，可以得到一个复杂性类集合的层级，其中每个集合都能够“模拟”其下方集合中的复杂性类。

对于未来的工作，研究本文在一致设置下给出的逻辑刻画将会很有意义。我们知道，对于实数，刻画 $\mathsf{AC}^0_{\mathbb{R}} = \mathsf{FO}_{\mathbb{R}}[\mathsf{Arb}_{\mathbb{R}}]{+}\mathsf{SUM}_{\mathbb{R}}{+}\mathsf{PROD}_{\mathbb{R}}$ 无论在非一致情形下，还是对于由多项式时间可计算电路（$\mathsf{P}_{\mathbb{R}}$-uniform）、对数时间可计算电路（$\mathsf{LT}_{\mathbb{R}}$-uniform）和一阶逻辑可定义电路（$\mathsf{FO}_{\mathbb{R}}$-uniform）给出的一致性准则下，都成立 [BV21]。我们相信，我们在此处给出的结果在类似的一致设置下同样成立，不过这仍需进一步研究。

另一个尚待探索的方向，是为这些新的复杂性类寻找有趣的问题（甚至可能是完全问题）。这甚至可能为经典情形带来新的见解。

一种有望分离代数电路复杂性类的方法，或许是改进 Cucker 所采用的方法。Cucker 证明了问题 FER 能够区分 $\mathsf{NC}^i_{\mathbb{R}}$-类 [Cuc92]；该问题本质上是在询问一个点是否位于某条费马曲线上。同一证明或许也适用于 $\mathsf{NC}^i_{\mathbb{C}}$-类。

另一种值得称为“代数电路”的模型，是 Valiant 意义下的算术电路 [Val79]。该模型与本文提出的模型相似，不同之处在于，通常只允许使用加法门和乘法门。本文提出的思路或许也能进一步加深我们对这种计算模型的认识。

在布尔情形中，除了 AC 和 NC 层级外，人们还常研究所谓的 SAC层级。该层级通过限制一种门类型的扇入来定义，即限制合取门或析取门的扇入。已知在这种情况下，限制哪种门类型并无区别。接下来可以尝试在代数情形中定义一种合理的 SAC 层级类比。我们认为，在代数情形中，限制哪种门类型确实会产生区别。随后，这种模型或许有助于研究那些相应运算不遵循相同公理的代数结构。

Appendix A. Bounding circuit depth

定义 A.1。设 C 是一个电路，令 L_k 为所有满足 $\mathsf{depth}(g) = k$ 的门 $g \in C$ 所构成的集合。则称 L_k 为 C 的第 k 层。$\triangleleft$

引理 A.2. 设 $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ 是一个电路族， f 是一个次线性函数，而 $g_C(k)$ 是电路 $C \in \mathcal{C}$ 在第 k 层的门数。此外，设 $\alpha := 2^{-\frac{\log_2 n}{f\ n}()}$。

如果要求对每一层 k，不等式 $g_C(k) \leq \lfloor \alpha \cdot g_C(k-1) \rfloor$ 都成立，那么电路 C 的深度为 $\mathcal{O}(f(n))$。

证明. 我们希望用次线性函数 f 来界定电路的深度（即层数 k）。因此，对于任意输入规模 n，我们令 $k = f(n) < n$。此外，若要求不等式 $g_C(k) \leq \lfloor \alpha \cdot g_C(k-1) \rfloor$ 成立，则当 $n\alpha^k \leq 1$ 成立时，电路的深度达到最大，因为因子 α 会应用 k 次，每一层应用一次。代入可得

$$\begin{aligned} n \cdot \alpha^k \leq 1 &\iff n \cdot 2^{-\frac{\log_2 n}{f(n)} k} \leq 1 \\ &\iff n \cdot 2^{-\log_2 n} \leq 1. \end{aligned}$$

□

推论 A.3. 如果要求对于每一层 k，不等式 $g_C(k) \leq \left\lfloor 2^{-\log_2 n\,(\log_2 n)^i} \cdot g_C(k-1) \right\rfloor$ 成立，则电路 C 的深度为 $\mathcal{O}((\log_2 n)^i)$。

Appendix B. Guarded Predicative Recursion

记法 B.1. 与记法 4.5 中的相对化聚合器类似，对于相对化量词，我们写作

$$(\exists x_1, \ldots, x_k.\varphi)\psi$$

作为 $\exists x_1 \ldots \exists x_k(\varphi \wedge \psi)$ 的简写，并将

$$(\forall x_1, \ldots, x_k.\varphi)\psi$$

作为 $\forall x_1, \ldots, x_k(\varphi \to \psi)$ 的简写。

对于一阶公式 φ 和关系变量 P，如果 P 在 φ 中不出现在否定的辖域内，我们记作 $\varphi(P^+)$。

定义 B.2（GPR^i）。设 $\mathcal{R}$ 为一组关系，使得 BIT 可在 FO$[\mathcal{R}]$ 中定义。关于 σ 的 FO$[\mathcal{R}] + \mathrm{GPR}^i$-公式集合，是由关于 σ 的 FO$[\mathcal{R}]$-公式语法加上以下规则扩展得到的公式集合：

$$\varphi ::= [P(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}) \equiv \theta(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}, P^+)]\psi(P^+),$$

其中， ψ 和 θ 是关于 σ 的 FO$[\mathcal{R}]$-公式， $\overline{x}$、$\overline{y_1}$、.....、$\overline{y_i}$、$\overline{y_{i+1}}$ 是变量元组，P 是关系变量，并且涉及 P 的每个原子子公式在 θ 中

（1）具有 $P(\overline{x}$、$\overline{y_1}$、.....、$\overline{y_i}$、$\overline{y_{i+1}})$ 的形式， $\overline{y_1}$、.....、$\overline{y_i}$、$\overline{y_{i+1}}$ 处于受保护量化的作用域内

$$Q\overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}. \left(\bigvee_{j=1}^{i} \left(\overline{z_j} \leq \overline{y_j}/2 \wedge \bigwedge_{k=1}^{j-1} \overline{z_k} \leq \overline{y_k} \right) \wedge \xi(\overline{y_1}, \ldots, \overline{y_i}, \overline{y_{i+1}}, \overline{z_1}, \ldots, \overline{z_i}, \overline{z_{i+1}}) \right)$$

其中 $Q \in \{\forall, \exists\}$、 $\xi \in$ 是带有 ξ 的 FO$[\mathcal{R}]$，且不包含输入结构中给定关系的任何关系符号，并且

（2）从不出现在未以这种方式受保护的任何量化的作用域内。

GPR^i 的语义以类似于定义 4.6 中 GFR^i_R 的语义的方式定义。$\triangleleft$

References

[Alu09] Paolo Aluffi。《代数：第0章》。数学研究生丛书。美国数学会，2009年。 ISBN: 9780821847817（引自第2页）。[BCS97] Peter B¨urgisser、Michael Clausen 和 Mohammad A min Shokrollahi。《代数复杂性理论》。第315卷。数学科学基础丛书。Springer，1997年。 ISBN: 3-540-60582-7（引自第6页）。[Blu+98] Lenore Blum 等。《复杂性与实数计算》。Springer New York，1998年。ISBN: 978-1-4612-6873-4。URL: https://doi.org/10.1007/978-1-4612-0701-6（引自第1、6、7页）。[B¨ur13] Peter B¨urgisser。《代数复杂性理论中的完备性与归约》。第7卷。施普林格科学与商业媒体，2013年。 ISBN: 978-3- 540-66752-0（引自第2页）。[BV21] Timon Barlag 和 Heribert Vollmer。“实数上常深度电路的逻辑刻画”。载于：《逻辑、语言、信息与计算——第27届国际研讨会，WoLLIC 2021，虚拟活动，2021年10月5日至8日，论文集》。Alexandra Silva、Renata Wassermann 和 Ruy J. G. B. de Queiroz 编。第13038卷。计算机科学讲义。Springer，2021年，第16–30页。DOI: 10.1007/978-3-030-88853-4_2。URL: https://doi.org/10.1007/978-3-030-88853-4_2（引自第5、6、9、25页）。[CM99] Felipe Cucker 和 Klaus Meer。“刻画实数上复杂性类的逻辑”。载于：《符号逻辑杂志》64.1（1999），第363–390页。URL: https://doi.org/10.2307/2586770（引自第7、9页）。[Cuc92] Felipe Cucker。“$P_{\mathbb{R}} \neq NC_{\mathbb{R}}$”。载于：《复杂性杂志》8.3（1992），第230–238页。URL: https://doi.org/10.1016/0885-064X(92)90024-6（引自第2、5、25页）。[DHV18] Arnaud Durand、Anselm Haak 和 Heribert Vollmer。“布尔与算术小深度电路类的模型论刻画”。载于：《第33届ACM/IEEE计算机科学逻辑研讨会年度会议论文集》。LICS '18。英国牛津：计算机协会，2018年，第354–363页。ISBN: 9781450355834。DOI: 10.1145/3209108.3209179。URL: https://doi.org/10.1145/3209108.3209179（引自第1、2、10、22、25页）。[GG98] Erich Gr¨adel 和 Yuri Gurevich。“元有限模型论”。载于：《信息与计算》140.1（1998），第26–81页。DOI: 10.1006/inco.1997.2675。URL: https://doi.org/10.1006/inco.1997.2675（引自第7页）。[GM95] Erich Gr¨adel 和 Klaus Meer。“实数上的描述复杂性理论”。载于：《第27届ACM计算理论研讨会年度会议论文集》，1995年5月29日至6月1日，美国内华达州拉斯维加斯。Frank Thomson Leighton 和 Allan Borodin 编。ACM，1995年，第315–324页。DOI: 10.1145/225058.225151。URL: https://doi.org/10.1145/225

（引自第 7 页）。[Gr¨a+07] Erich Gr¨adel 等人。《有限模型理论及其应用》。理论计算机科学丛书。EATCS 系列。Springer，2007。 ISBN: 978-3-540-00428-8。DOI: 10.1007/3-540-68804-8。URL: https://doi.org/10.1007/3-540-68

（引自第 7 页）。[Imm99] Neil Immerman。描述复杂性。计算机科学研究生教材。Springer，1999 年。ISBN: 978-1-4612-6809-3。DOI: 10.1007/978-1-4612-0539-5。URL: https://doi.org/10.1007/978-1-4612-0539-5（引自第 7、10、12 页）。

[Lan02] Serge Lang。代数。Springer New York，2002 年。DOI：10.1007/978-1-4613-0041-0。URL：https://doi.org/10.1007/978-1-4613-0041-0（引自第 2 页）。[Lib04] Leonid Libkin。有限模型理论基础。理论计算机科学文献。EATCS 丛书。Springer，2004 年。ISBN：3-540-21202-7。DOI：10.1007/978-3-662-07003-1。URL：https://doi.org/10.1007/978-3-662-07003-（引自第 7 页）。[Val79] Leslie G. Valiant。“代数中的完备性类”。载于：第十一届 ACM计算理论研讨会年会论文集（1979 年）（引自第 2、25 页）。[Vol99] Heribert Vollmer。电路复杂性导论——一种一致方法。Springer，1999 年。ISBN：978-3-540-64310-4。URL：https://doi.org/10.1007/978-3-662-0392

（引自第 5、7 页）。

Institut für Theoretische Informatik, Leibniz Universität Hannover, 30167 Han-nover, Germany*Email address*: gaube@thi.uni-hannover.de*Email address*: barlag@thi.uni-hannover.de

Institut für Theoretische Informatik, Universität zu Lübeck, 23562 Lübeck, Ger-many*Email address*: fch@tcs.uni-luebeck.de