arXiv:2601.03201v1 [cs.LO] 6 Jan 2026

通过加权结构递归查询神经网络

Martin Grohe[1],Christoph Standke[1],Juno Steegmans[2],和 Jan Van den Bussche[2]

[1]亚琛工业大学[2]哈瑟尔特大学

摘要

将机器学习模型视为一种内涵数据形式，对其进行表达性查询，可以使用声明式语言实现模型的验证与解释，从而使数据的学习得到的表示更加易于理解。受对前馈神经网络进行查询的启发，我们研究了加权结构逻辑。在不存在神经网络深度上界的情况下，这类逻辑必须纳入递归；为此，我们重新审视了 Gr¨adel 和 Gurevich 提出的函数式不动点机制。我们采用类 Datalog 语法来表达该机制；将不动点逻辑的范式扩展到加权结构；并展示一种等价的“宽松”不动点机制，该机制允许覆盖归纳定义的权重函数的值。我们提出了函数式不动点逻辑的“标量”限制，该限制具有多项式时间数据复杂度，并证明它能够表达约简网络上所有权重为多项式有界的、与模型无关的 PTIME 查询。相比之下，我们证明非常简单的与模型无关的查询已经是 NP 完全的。最后，我们考察通过迭代转导对加权结构进行的变换。

1. 引言

从多个角度来看，查询 机器学习模型都有其必要性：

- 数据科学项目会生成大量模型制品，这些制品应像其他任何类型的数据一样，使用数据库技术进行管理。具体来说，我们应该能够查询这些数据。MLflow 或 W&B 等平台提供基于实验元数据的管理性筛选和搜索功能，但无法深入查询模型本身。

- 在机器学习术语中，“查询”模型通常只是指将模型应用于新的输入。不过，我们可以有更大的抱负。考虑一种典型的

用于数值特征元组（向量）的布尔分类器。此类模型表示由所有被分类为真的可能元组构成的潜在无限关系。我们希望能够像查询关系数据库中的普通关系一样查询这些关系。

- 在数据库理论中，以约束查询语言 [33, 35, 37]为名，对由约束有限表示的无限关系进行查询，早已得到广泛研究。

- 模型可解释性或可解释人工智能 [40, 44] 的众多方法，可以视为针对模型和数据的各种不同查询（例如，寻找反事实、计算沙普利值），但它们并不属于一个涵盖一切的结构化查询语言框架。

- 在神经网络验证 [6, 12], 中，模型表示实数上的函数，而待验证的性质则用关于这些函数的全称量化约束表达式来表示。这些表达式已经可以被视为一种最简查询语言。

上述考量促使我们研究神经网络查询语言的理论基础。事实上，相关方向的研究已经开始。Arenas 等人 [8, 7]研究了布尔决策树以及定义在任意长度布尔向量上的一阶逻辑。他们将这些逻辑与可对决策树节点进行量化的逻辑相结合，从而通过 SAT 求解实现查询求值。Grohe 等人 [28] 研究了前馈神经网络（FNN）以及用于加权结构的一阶逻辑 FO(SUM)，并通过 SQL [24]进行查询求值。

聚焦于与模型无关的查询，[1] Grohe 等人展示了一个引人注目的二分结果。若不限制网络的深度，FO(SUM)由于缺乏递归，只能表达平凡的与模型无关的查询。然而，对于固定深度前馈神经网络，FO(SUM)具有相当强的表达能力，能够表达约束查询语言 FO($\mathbf{R}$ f) 的线性算术片段中的所有查询。在这里，FO($\mathbf{R}$ f) 表示带有一个额外函数符号 f的实数上的一阶逻辑。这种语言是表达与模型无关查询的一种自然标尺：模型通过符号 f作为黑箱访问。

本文探讨了提升 Grohe 等人工作中固定深度假设的方法与障碍，并作出以下贡献。我们首先构建扩展 FO(SUM) 以支持递归所需的理论。其核心是，FO(SUM) 是一种用于定义权重函数的逻辑。用于归纳定义关系的不动点逻辑，早已为逻辑、数据库理论和有限模型理论所熟知。相比之下，对于权重函数，我们只知道一种方案，即函数不动点 [25], ；本文将重新审视这一方案，并以更系统的方式加以发展。我们考察了替代语义，并证明它们的表达能力相同。我们定义了 FO(SUM)的递归扩展，称为 IFP(SUM)，并分别采用类 Datalog 语法和不动点逻辑风格语法来表述，

[1]如果一个查询无法区分两个在结构上可能不同、但恰好表示相同函数或分类器的模型，那么该查询就是与模型无关的 [43]。

并建立了一种范式，扩展了已知的不动点逻辑和带否定的膨胀式 Datalog 的范式 [3, 18, 36]。

我们对 IFP(SUM) 的处理通常适用于所有加权结构。为了将其应用于前馈神经网络，我们首先定义一个受限片段，称为标量 IFP(SUM)，其中禁止将乘法与递归结合使用。我们证明，这个片段具有 PTIME 数据复杂度。由于递归与权重上的算术运算相结合，这并非易事。接着，我们证明，对于权重有多项式上界的“约化”前馈神经网络，标量 IFP(SUM) 可以表达所有 与模型无关的 PTIME 查询。我们将这一条件精确定义，并解释其必要性。

接下来，我们回到黑盒逻辑 FO($\mathbf{R}$ f)。相关问题是：在不作固定深度假设的情况下，IFP(SUM) 是否能够表达所有线性 FO($\mathbf{R}$ f)。然而，答案很可能是否定的。我们将证明，非常简单的 FO($\mathbf{R}$ f) 查询已经是 NP-困难的。因此，至少对于标量 IFP(SUM)，答案是否定的，除非 P = NP。对于不受限制的 IFP(SUM)，这个问题仍未解决。

最后，我们考察 IFP(SUM) 的一个表达能力很强的扩展，它允许通过迭代来定义将结构映射为结构的转导。通过回顾 Grohe 等人关于表达能力结果的证明，我们得出结论：每个线性 FO($\mathbf{R}$ f) 查询确实都可以通过迭代 IFP(SUM) 转导来计算。

本文安排如下。第2 节讨论相关工作。第3 节介绍预备知识。第4节给出关于 IFP(SUM)的结果，第5节介绍标量片段的相关结果。第6 节给出 NP 完全性结果。第7 节讨论迭代转导。第8 节作出总结。附录中提供了证明细节。

2. 相关工作

用于加权结构的逻辑FO(SUM) 是 Gr¨adel 和 Gurevich [25]所定义的亚有限结构逻辑的一种具体化。Torunczyk 显然并不了解这项工作，却提出了一个非常相似的方案，并研究了在一般数值论域（半环）类上查询求值和枚举的组合复杂性 [46]。在统计学习的背景下，Van Bergerem 和 Schweikardt [9, 10]研究了通过“权重聚合”扩展一阶逻辑这一密切相关的方向。当然，FO(SUM) 也与带有聚合和算术运算、用于形式化基本 SQL 的关系演算颇为相似 [38]。区别在于，在 SQL 中，关系可以包含多个数值列；而亚有限结构拥有一个独立的抽象论域，在该论域上定义关系和权重函数（其取值属于一个独立的数值论域）。

权重函数与半环标注关系也非常相似 [26, 4]。不过，在这一领域，研究重点通常是权重通过相应的关系运算符隐式地相加、相乘或求和的语言。相比之下，FO(SUM)则显式地处理数值。[2] Geerts 和 Reutter 也提出了显式张量逻辑 [23, 21] ，旨在刻画图神经网络无法区分权重函数的情形。相比之下，我们在此关注的是对前馈神经网络进行模型无关查询。

[2]另请参阅关于非数值注释的隐式处理与显式处理的相关研究 [13, 22]。

为此，我们在FO(SUM)中加入递归。将递归与数值计算结合时，终止性便成为一个需要关注的问题。我们采用并研究了能够保证终止的膨胀不动点变体。相比之下，在一项非常有意思的相关工作中，Abo Khamis 等人研究了基于半环标注关系的 Datalog，并探讨了能够保证收敛的半环条件 [5]。我们还提及了关于带聚合的逻辑程序语义的研究 [42]。该研究着重为涉及否定和聚合的一般递归规则集提供自然语义，并寻找刻画此类规则集何时不存在歧义（即具有唯一良基模型）的条件。

第 6 节中关于我们 NP 困难性结论的相关研究，考察了判定前馈神经网络的各种性质的复杂性，包括其所表示函数的单射性和满射性 [20]。此外，已有研究表明，神经网络验证的其他多种形式都是 NP 完全的 [49]。然而，现有结果都关键依赖于这样一个假设：待验证网络的 宽度 （输入维度）会计入验证问题的输入规模。相比之下，我们的 NP 困难性结论更强，因为它适用于宽度为一的网络。

3. 预备知识

3.1. 前馈神经网络

前馈神经网络 [45]，（简称 FNN）是一种有向无环图，其节点和边均带有权重。源节点称为输入节点，编号从 1 到 m；汇节点称为输出节点，编号从 1 到 p。其他所有节点均称为隐藏节点。我们始终假设输入节点和输出节点彼此不同，也就是说，不允许图仅由孤立节点构成。节点的权重也称为其偏置；输入节点是例外，它们没有偏置。

设 $\mathcal{N}$ 是上述前馈神经网络。由 $\mathcal{N}$ 表示、使用 ReLU 激活和线性输出的函数 $f^{\mathcal{N}}\colon \mathbb{R}^m \to \mathbb{R}^p$ 定义如下。我们首先按 u 的深度（从输入节点到 u 的路径的最大长度）进行归纳，为每个节点 u 定义一个函数 $f_u^{\mathcal{N}}\colon \mathbb{R}^m \to \mathbb{R}$。如果 u 是第 i 个输入节点，则 $f_u^{\mathcal{N}}(x_1,\dots,x_m) = x_i$。如果 u 是偏置为 b 的隐藏节点，且其输入边为 $(v_1,u),\dots,(v_k,u)$，对应权重分别为 $w_1,\dots,w_k$，则 $f_u^{\mathcal{N}}(\boldsymbol{x}) = \mathrm{ReLU}(b + \sum_j w_j f_{v_j}^{\mathcal{N}}(\boldsymbol{x}))$。这里，$\mathrm{ReLU}\colon \mathbb{R}\to\mathbb{R}$：$z \mapsto \max(0,z)$。如果 u 是输出节点，则 $f_u^{\mathcal{N}}(\boldsymbol{x})$ 的定义与隐藏节点类似，但省略 ReLU 的应用。最后，我们可以将 $f^{\mathcal{N}}(\boldsymbol{x})$ 定义为 $(f^{\mathcal{N}}\mathrm{out}_1(\boldsymbol{x})\dots, f_{\mathrm{out}_p}^{\mathcal{N}}(\boldsymbol{x}))$，其中 $\mathrm{out}_1,\dots,\mathrm{out}_p$ 是输出节点。

$\mathbf{K}(m,\ p)$ 表示具有 m 个输入节点和 p 个输出节点的前馈神经网络类。当输入节点数、输出节点数或两者均未固定时，我们也分别记作$\mathbf{K}(*,\ p)$、$\mathbf{K}(m,\ *)$ 和 $\mathbf{K}(*,\ *)$。

3.2. 模型无关的查询与 FO($\mathbf{R}$ f)

一般而言，我们可以将 r元查询定义为 $\mathbf{K}(m,p)$，其 k 参数定义为关系$Q \subseteq \mathbf{K}(m,p) \times \mathbb{R}^k \times \mathbb{R}^r$。如果 $Q(\mathcal{N},\boldsymbol{z},\boldsymbol{y})$ 成立，我们就称 $\boldsymbol{y}$是 Q 在 $\mathcal{N}$上的一个可能结果

以及 $\boldsymbol{z}$。在特殊情况下 $r=0$，我们得到一个布尔查询（如果 $Q(\mathcal{N},\boldsymbol{z})$ 成立则为 true，否则为 false）。

示例 3.1。查询最简单的示例是推理，即在给定输入上评估模型。形式化地，对于推理，我们有 $k=m$ 且 $r=p$ 且 $Q(\mathcal{N},\ \boldsymbol{z},\ \boldsymbol{y})$ 当且仅当 $\boldsymbol{y}=f^{\mathcal{N}}(\boldsymbol{z})$。然而，可解释机器学习中的许多其他任务也 [40]符合上述查询概念。例如，返回反事实解释或对抗样本、检查鲁棒性、计算某一点处的沙普利值、计算某一点处的梯度，或检查由参数给定的某些值域之间的可微性。

情况 $k=0$涵盖了对神经网络全局行为的分析或验证，而不是分析其在给定输入向量上的行为。例如，回顾一下，由带有 ReLU 和线性输出节点的前馈神经网络表示的函数始终是分段线性的。针对 $m=1=p$ 的一个可能的二元查询可以询问前馈神经网络 $\mathcal{N}$ 上满足以下条件的结果 $(b,\ s)$：b 是 $f^{\mathcal{N}}$ 的一个断点，且 s 是 b处的右侧斜率。对于更高的输入维度，例如，我们可以询问由 $f^{\mathcal{N}}$诱导的输入空间划分的支撑超平面的系数。 □

刚才提到的所有查询示例都是模型无关的，这意味着对于表示同一函数的两个网络 $\mathcal{N}$ 和 $\mathcal{N}'$，查询会得到相同的结果，即 $f^{\mathcal{N}}=f^{\mathcal{N}'}$。模型无关方法是可解释机器学习和可解释人工智能 [40]中的一个成熟类别。[3]

作为针对 $\mathbf{K}(m,p)$ 上与模型无关的查询的衡量标准，我们可以使用实数上的一阶逻辑，其中包含 p 个元数为 m 的函数符号 $f_1,\dots,f_p$，它们共同表示一个函数$f:\mathbb{R}^m\to\mathbb{R}^p$。为简单起见，本文主要考察 $p=1$ 的情形；这本身已是机器学习中的典型情况，涉及回归和二元分类等任务。

形式化地，令 $\mathbf{R}=(\mathbb{R},+,\cdot,(q)_{q\in\mathbb{Q}},<)$ 表示实数结构，其中包含所有有理数对应的常数。$\mathrm{FO}(\mathbf{R}\ f/m)$ 表示实数结构 $\mathbf{R}$ 词汇表上的一阶逻辑，并额外包含一个 m元函数符号 f。当乘法仅限于项与常数之间（即数乘）时，我们将其记为 $\mathbf{R}_{\mathrm{lin}}$ 和 $\mathrm{FO}(\mathbf{R}_{\mathrm{lin}},f/m)$。当 m 已明确时，我们会在记号中省略它。

一个带有 $k+r$ 个自由变量 $z_1,\dots,z_k,y_1,\dots,y_r$ 的 $\mathrm{FO}(\mathbf{R}\ f/m)$ 公式 φ，现在自然地表示了在 $\mathbf{K}(m,1)$ 上的一个 r 元查询 Q_φ，其具有 k 个参数。具体而言，当且仅当 $\varphi(\boldsymbol{z},\boldsymbol{y})$ 在 $(\mathbf{R}\ f^{\mathcal{N}})$ 中成立时，$Q_\varphi(\mathcal{N},\boldsymbol{z},\boldsymbol{y})$ 成立。根据定义，这个查询是模型无关的。

示例 3.2。我们给出两个公式来说明 $\mathrm{FO}(\mathrm{R}\ f)$ 的语法。推理查询（示例 3.1）可表示为公式 $y=f(\boldsymbol{z})$。针对 K(1, 1) 的查询询问 $\lim_{x\to+\infty}f^{\mathcal{N}}(x)=-\infty$ 是否成立，可表示为 $\forall u<0\,\exists x_0>0\,\forall x>x_0\,f(x)<u$。这两个公式都属于 $\mathrm{FO}(\mathrm{R}_{\mathrm{lin}},f)$，因为它们不使用乘法。我们注意到，示例 3.1 中的所有查询（沙普利值除外）都可以在 $\mathrm{FO}(\mathrm{R}\ f)$ [28] 中表示。

[3]举一个非模型无关方法的例子，我们可以提到神经网络的剪枝：找出那些对所表示函数的影响微乎其微的神经元。

3.3. FO(SUM)

FO(SUM) 是一种用于查询加权结构的逻辑；加权结构是在标准关系结构的基础上额外配备了权重函数。这些函数将元素元组映射为数值。

在我们的情形中，数值取自“扩充的”有理数 $\mathbb{Q}_\perp := \mathbb{Q} \cup \{\perp\}$或实数 $\mathbb{R}_\perp = \mathbb{R} \cup \{\perp\}$。这里，$\perp$ 是表示未定义值的额外元素。我们将通常的序 $\leq$ 从 $\mathbb{R}$ 扩展到 $\mathbb{R}_\perp$，规定对于所有 $x \in \mathbb{R}_\perp$，$\perp \leq x$。我们通过规定 $x+y := \perp$、$x-y := \perp$，以及当 $x = \perp$ 或 $y = \perp$时 $x \cdot y := \perp$，来扩展加法、减法和乘法。类似地，当 $x = \perp$ 或 $y = \perp$或 $y = 0$ 时，我们规定 $x/y := \perp$，以此扩展除法。

一个加权词汇 Υ 是由有限个关系符号和权重函数符号组成的集合。每个符号 S 都有一个元数 $\mathrm{ar}(S)$，它是一个自然数。一个加权 Υ-结构 $\mathcal{A}$ 由一个称为 $\mathcal{A}$论域的有限集合 A、每个 k 元关系符号 $R \in \Upsilon$ 对应的一个 k元关系 $R^{\mathcal{A}} \subseteq A^k$，以及每个 k元权重函数符号 $F \in \Upsilon$ 对应的一个函数$F^{\mathcal{A}} \colon A^k \to \mathbb{R}_\perp$组成。为简便起见，我们通常将加权结构简称为结构。

我们通过以下语法定义逻辑 FO(SUM) 的公式 φ 和权重项 θ 集合：

$$\varphi ::= x = y \mid R(x_1, \ldots, x_{\mathrm{ar}(R)}) \mid \theta \leq \theta \mid \neg\varphi \mid \varphi * \varphi \mid Q x\, \varphi \tag{3.A}$$

$$\theta ::= r \mid F(x_1, \ldots, x_{\mathrm{ar}(F)}) \mid \theta \circ \theta \mid \mathsf{if}\ \varphi\ \mathsf{then}\ \theta\ \mathsf{else}\ \theta \mid \sum_{(x_1,\ldots,x_k):\varphi} \theta. \tag{3.B}$$

这里 x，y，x_i是变量，R是关系符号，$* \in \{\vee, \wedge, \to\}$是布尔连接词，$Q \in \{\exists, \forall\}$ 是量词，$r \in \mathbb{Q}_\perp$ 是数值常量，[4] F 是权重函数符号，而 $\circ \in \{+, -, \cdot, /\}$ 是算术运算符。语义是相对于二元组（$\mathcal{A}$，ν）定义的，其中 $\mathcal{A}$ 是一个结构，ν 是将论域 A 中的元素赋给变量的赋值。公式 φ 取布尔值 $[\![\varphi]\!]^{(\mathcal{A},\nu)} \in \{0, 1\}$，权重项 θ 取值 $[\![\theta]\!]^{(\mathcal{A},\nu)} \in \mathbb{R}_\perp$。这些值以归纳方式定义；由于大多数定义显而易见或遵循一阶逻辑的常见语义，这里略去不述。求和算子的语义如下：

$$[\![\sum_{(x_1,\ldots,x_k):\varphi} \theta]\!]^{(\mathcal{A},\nu)} := \sum_{(a_1,\ldots,a_k)\in A^k} [\![\varphi]\!]^{(\mathcal{A},\nu \frac{a_1,\ldots,a_k}{x_1,\ldots,x_k})} \cdot [\![\theta]\!]^{(\mathcal{A},\nu \frac{a_1,\ldots,a_k}{x_1,\ldots,x_k})},$$

其中 $\nu \frac{a_1,\ldots,a_k}{x_1,\ldots,x_k}$ 表示通过将 a_i 赋给 x_i而从 ν 得到的更新后的赋值，适用于 $i = 1, \ldots, k$。

FO(SUM) 表达式 要么是公式，要么是权重项。表达式 ξ 的自由变量 集合 $\mathrm{free}(\xi)$ 的定义很直接，其中求和$\sum_{(x_1,\ldots,x_k):\varphi}$将变量 $x_1, \ldots, x_k$绑定。闭表达式 是不含自由变量的表达式。闭公式 也称为句子。

[4]也可以允许任意实常数，但本文关注的是计算表达式的算法，因此合理地将讨论范围限制在有理常数上。

对于表达式 ξ，记号 $\xi(x_1,\dots,x_k)$ 表示其所有自由变量都属于 $\{x_1,\dots,x_k\}$。不难看出，值 $[\![\xi]\!]^{(\mathcal{A},\nu)}$ 只取决于自由变量的取值 $a_i := \nu(x_i)$。因此，我们可以不必明确提及赋值 ν，而写作 $[\![\xi]\!]^{\mathcal{A}}(a_1,\dots,a_k)$，而不是 $[\![\xi]\!]^{(\mathcal{A},\nu)}$。如果 ξ 是闭表达式，我们就只写 $[\![\xi]\!]^{\mathcal{A}}$。对于公式 $\varphi(x_1,\dots,x_k)$，我们也写作 $\mathcal{A}\models\varphi(a_1,\dots,a_k)$，而不写 $[\![\varphi]\!]^{\mathcal{A}}(a_1,\dots,a_k)=1$；对于句子 φ，我们写作 $\mathcal{A}\models\varphi$。

注意，我们可以在 FO(SUM) 中使用求和与除法来表示平均值。引入一种表示平均值的记法会很有用：对于项 θ 和公式 φ，我们用 $\mathsf{avg}_{\boldsymbol{x}:\varphi}\,\theta$ 来简写 $(\sum_{\boldsymbol{x}:\varphi}\theta)/\sum_{\boldsymbol{x}:\varphi}1$。注意，如果没有元组 $\boldsymbol{x}$ 满足 φ，$\mathsf{avg}_{\boldsymbol{x}:\varphi}\,\theta$ 的取值为 $\perp$。

3.4. 作为加权结构的前馈神经网络

任意前馈神经网络 $\mathcal{N}\in\mathbf{K}(*,*)$ 都可以自然地视为一个加权结构

$$\mathcal{N}=(V,E^{\mathcal{N}},\mathrm{In}^{\mathcal{N}},\mathrm{Out}^{\mathcal{N}},b^{\mathcal{N}},w^{\mathcal{N}}),$$

其中，V（论域）是节点集合；$E^{\mathcal{N}}$ 是二元边关系；$\mathrm{In}^{\mathcal{N}}$ 是一个二元关系，它在线性排序 $\mathcal{N}$ 的输入节点上构成线性序（在其余节点上未定义）；同样，$\mathrm{Out}^{\mathcal{N}}$ 是输出节点上的线性序；$b^{\mathcal{N}}$ 是节点上的一元偏置权重函数；$w^{\mathcal{N}}$ 是边上的二元权重函数。[5]

这对 [28]，的模型略作了推广。在该模型中，词汇表取决于网络的输入维度和输出维度；对于网络 $\mathcal{N}\in\mathbf{K}(p,q)$，$p$输入节点通过一个单元素 p元关系访问，q 输出节点则通过一个单元素 q元关系访问。请注意，在我们的版本中，可以轻松地还原这些关系。对于 $\mathbf{K}(p,q)$ 中的前馈神经网络，其单元素输入关系可由 FO(SUM)公式 $\varphi_{\mathrm{In}(p)}(x_1,\dots,x_p) := \bigwedge_{i=1}^{p-1}\big(\mathrm{In}(x_i,x_{i+1})\wedge x_i\neq x_{i+1}\big)\wedge\forall y\,(\mathrm{In}(y,y)\rightarrow\bigvee_{i=1}^{p}y=x_i)$ 定义，输出关系也可类似定义。我们的方法的优点在于，无论前馈神经网络的维度如何，都可以编写统一适用的查询。

我们可以用权重函数val 扩展相应的词汇表（E，In、Out b，w），为网络提供输入值。现在，我们可以基于扩展后的词汇表，给出 FO(SUM) 中公式和权重项的一些示例。

例3.3。对于任意自然数 ℓ，都存在一个一阶逻辑公式 深度$\leq\ell(u)$，用于定义任意网络中深度不超过 ℓ 的节点。接着，我们可以用权重项 $f_u^{\mathcal{N}}$(参见第 3.1节)为这类节点定义函数 $eval_{\leq\ell}(u)$，其定义如下：

求值 ≤ 0:=，若 $\mathrm{In}(u,u)$ 则 $\mathrm{val}(u)$，否则 $\perp$；求值 $\leq\ell+1$:=，若深度$\leq\ell(u)$，则求值$\leq\ell(u)$；否则，若深度$\leq\ell+1(u)$，则若 $\mathrm{Out}(u,u)$，则 $b(u)+\sum_{v:E(v,u)}w(v,u)\cdot$求值$\leq\ell(v)$；否则 $\mathrm{ReLU}(b(u)+\sum_{v:E(v,u)}w(v,u)\cdot$ 求值 $\leq\ell(v))$；否则 $\perp$

[5]确切地说，对于每个输入节点 u，$b^{\mathcal{N}}(u)=\perp$；对于每一对不属于 E 的 (u,v)，$w^{\mathcal{N}}(u,v)=\perp$。

这里，对于任意权重项 θ，ReLU(θ) 是 if $\theta \geq 0$ then θ else 0 的缩写。

□

上述示例本质上表明，FO(SUM) 能够表达固定深度网络上的推理查询（参见第3.2节）。事实证明，FO(SUM)的能力远不止于此，实际上达到了由 FO($\mathbf{R}_{\text{lin}}, f$) 所设定的标准：

定理 *3.4*（[28]）。设 m、k 和 ℓ 为自然数，且设 Q 为 *K(m, 1)* 上的一个布尔查询，具有 k 个参数，可在 *FO($\mathbf{R}_{\text{lin}}, f$)* 中表达。则存在一个在词汇表（$E$、*In*、*Out* b、w、*val*）上的 *FO(SUM)* 句子 ψ，能够在 *K(m, 1)* 中所有深度为 ℓ 的网络上表达 Q。

这一结果表明，FO(SUM) 可以模拟对实数的任意量化，这是 FO($\mathbf{R}_{\text{lin}}, f$) 的核心特性。鉴于 FO(SUM) 本身只能对网络的有限节点集进行量化，这一结果十分惊人。

4. 用递归扩展 FO(SUM)

本文探讨如何突破定理 3.4中固定深度限制的方法及其障碍。本节将探讨一项必要的发展：为 FO(SUM) 扩展递归机制，以支持归纳定义。

回顾一下，FO(SUM) 表达式可以是定义关系的公式，也可以是定义权重函数的权重项。对于关系的归纳定义，逻辑机制已广为人知；本文考虑一种带分层结构的类 Datalog 语法，并采用各层的膨胀不动点语义 [3, 34, 1]。我们通过一个例子来回顾这种语义。

例4.1。考虑由二元边关系 E 以及一个额外的一元关系 S定义的有限有向图。以下程序用于检验：将图限制在从 S中的节点可达的节点上时，该图是否无环：

可达(x) $\leftarrow S(x) \vee \exists y(Reach(y) \wedge E(y, x))$;答案 $\leftarrow \neg\exists x\, Reach(x, x)$。

关系 $Reach$ 初始为空。定义公式会被反复求值，所得结果 添加 到$Reach$ 中，直到不再发生变化。[6] 零元关系 Ans提供查询的布尔答案，并以非递归方式定义。其定义规则位于后续层中，这意味着它会在$Reach$的递归终止后求值。

□

据我们所知，权重函数的归纳定义受到的研究要少得多。[7] 我们可以采用现有的唯一方案，它与膨胀式不动点具有良好的兼容性，称为函数不动点 [25]。其思想是，归纳定义的

[6]添加操作带来了膨胀性；定义公式不必是正的或单调的。[7]在标准一阶逻辑中，其语法缺乏我们在 FO(SUM) 中所拥有的公式与权重项之间的强大交互，因此只能使用公式将函数归纳地定义为关系（函数图），并以某种方式规定：如果所得关系不是函数图，则该定义是错误的。

权重函数 F 初始化为处处未定义（$\perp$）。随后反复求值定义性权重项，并据此更新 F，但仅限于 F 尚未定义的元组。重复这一过程，直到不再发生变化。

例4.2。以下归纳定义函数 $eval$ 对任意 FNN $\mathcal{N}$给出时，都统一表示函数 $f_u^{\mathcal{N}}$，适用于所有节点。（请比较例 $eval_{\trianglelefteq\ell}$中的权重项 3.3，它实现相同功能，但仅适用于节点 u 深度不超过 ℓ。）

$$
\begin{aligned}
eval(u) \leftarrow\ & \text{if } \mathrm{In}(u,u) \text{ then } val(u) \\
& \text{else if } \mathrm{Out}(u,u) \text{ then } b(u) + \textstyle\sum_{v:E(v,u)} w(v,u)\cdot eval(v) \\
& \text{else } \mathrm{ReLU}(b(u) + \textstyle\sum_{v:E(v,u)} w(v,u)\cdot eval(v)).
\end{aligned}
$$

4.1. IFP(SUM)逻辑

我们形式化定义 IFP(SUM)：在 FO(SUM)的基础上扩展不动点和函数不动点。我们的语法遵循分层Datalog [1]的模式。本节稍后还将讨论一种语法，它更符合有限模型论中定义一阶逻辑不动点扩展的方式 [18, 36]。

规则和层

加权词汇表 Υ 上的关系规则 形式为 $R(\boldsymbol{x}) \leftarrow \varphi$，其中 $R \in \Upsilon$是关系名称，$\boldsymbol{x}$ 是由 ar(R) 个互不相同的变量组成的元组，且 $\varphi(\boldsymbol{x})$ 是关于 Υ的 FO(SUM)公式。

Υ 上的权重函数规则 形式为 $F(\boldsymbol{x}) \leftarrow \theta$，其中 $F \in \Upsilon$ 是权重函数名称，$\boldsymbol{x}$是由 ar(R) 个互不相同的变量组成的元组，且 $\theta(\boldsymbol{x})$ 是关于Υ的权重项。

设 Υ和 Γ 是两个不相交的带权词汇。类型为 $\Upsilon \to \Gamma$ 的 IFP(SUM) 层是关于 $\Upsilon \cup \Gamma$的一组规则，记作 Σ，其中 Γ中的每个符号各有一条规则。按照分层 Datalog 术语的惯例，在 Σ中，Υ 中的符号称为外延符号，而 Γ中的符号称为内涵符号。

为定义上述 IFP(SUM) 层 Σ 的语义，我们首先在 $\Upsilon \cup \Gamma$-结构上引入直接后继 T_Σ。

定义4.3。给定结构 $\mathcal{B}$，我们定义 $T_\Sigma(\mathcal{B}) := \mathcal{C}$，其中 $\mathcal{C}$ 的论域等于B，即 $\mathcal{B}$的论域，并且 $\mathcal{C}$ 在外延符号上与 $\mathcal{B}$ 一致；随后按如下方式定义 $\mathcal{C}$ 中的内涵符号。

- 设 $R(\boldsymbol{x}) \leftarrow \varphi$ 是 Σ中的一条关系规则。那么 $R^{\mathcal{C}} := R^{\mathcal{B}} \cup \{\boldsymbol{a} \in B_{\mathrm{ar}(R)} \mid \mathcal{B} \models \varphi(\boldsymbol{a})\}$.
- 设 $F(\boldsymbol{x}) \leftarrow \theta$ 是 Σ中的一个权重函数规则。那么

$$
F^{\mathcal{C}}(\boldsymbol{a}) := \begin{cases} [\![\theta]\!]^{\mathcal{B}}(\boldsymbol{a}) & \text{if } F^{\mathcal{B}}(\boldsymbol{a}) = \perp; \\ F^{\mathcal{B}}(\boldsymbol{a}) & \text{otherwise.} \end{cases} \tag{4.A}
$$

由于 $\mathcal{B}$ 是有限的，而且内涵关系和权重函数在应用 T_Σ时只会增长，因此存在一个自然数 n，使得 $T_\Sigma^n(\mathcal{B})$（即从 $\mathcal{B}$开始连续应用 T_Σ 共 n次的结果）等于 $T_\Sigma^{n+1}(\mathcal{B})$。我们将此结果记为$T_\Sigma^\infty(\mathcal{B})$。

现在我们可以将 Σ 的语义定义为从 Υ-结构到 $\Upsilon\cup\Gamma$-结构的映射。设 $\mathcal{A}$ 为一个 Υ-结构，并令 $\mathcal{A}'$ 为其扩展得到的 $\Upsilon \cup \Gamma$-结构，其中所有内涵关系均设为空，所有内涵权重函数均设为处处未定义。于是 $\Sigma(\mathcal{A}) := T_\Sigma^\infty(\mathcal{A}')$。

程序与查询

一个 IFP(SUM) 程序就是一个有限的层序列。形式上，每个类型为 $\Upsilon \to \Gamma$的层本身也是该类型的程序；此外，若 Π 是类型为 $\Upsilon \to \Gamma_1$的程序，且 Σ 是类型为 $\Upsilon \cup \Gamma_1 \to \Gamma_2$的层，那么 $\Pi;\Sigma$ 就是类型为 $\Upsilon \to \Gamma_1 \cup \Gamma_2$的程序。其语义由顺序复合直接给出。

如例4.1所示，通常会从程序的内涵符号中指定一个答案符号（关系名称或权重函数名称）。这样，我们便可以用程序来表达查询，即从结构到关系或权重函数的映射。

4.2. 一种宽松语义

刚刚定义的直接后果算子只允许在权重函数 F 尚未在元组 $\boldsymbol{a}$上定义时，才为其设置新值 $F\ \boldsymbol{a}$。由此得到的函数式不动点语义的优点是，它保证能在有限结构上终止。然而在实践中，能够对权重函数进行更新可能会更方便。

示例4.4。 考虑一个距离矩阵 W，将其表示为一个带有严格全序关系ord 和二元权重函数 W 的结构。下面给出一个几乎逐字照搬的 IFP(SUM) 中计算所有点对最短路径距离的 Floyd-Warshall算法。

$$
\begin{aligned}
&chosen(k) \leftarrow next(k);\\
&\quad D(i,j) \leftarrow \text{if } chosen = \emptyset \text{ then}\\
&\qquad\qquad \text{if } \exists k(next(k) \wedge W(i,j) > W(i,k) + W(k,j)) \text{ then}\\
&\qquad\qquad\qquad \sum_{k:next(k)} W(i,k) + W(k,j)\\
&\qquad\qquad \text{else } W(i,j)\\
&\qquad \text{else if } \exists k(next(k) \wedge D(i,j) > D(i,k) + D(k,j)) \text{ then}\\
&\qquad\qquad\qquad \sum_{k:next(k)} D(i,k) + D(k,j)\\
&\qquad\qquad \text{else } D(i,j).
\end{aligned}
$$

这里， i、 j 和 k 是变量，而$next(k)$ 是 $\forall x(ord(x,k) \leftrightarrow chosen(x))$ 的缩写。辅助内涵关系$chosen$ 用于遍历图中的所有节点；$next(k)$ 选取下一个节点。

然而，在目前采用的函数式不动点语义下，这个程序无法按预期运行。第一次迭代后， D 就已经处处有定义，因此不会再对 D 进行更新。在一种更宽松、允许更新权重函数的语义下，该程序便能按预期运行。 □

上述示例表明，可以为层采用另一种宽松不动点语义，在这种语义下，权重函数可以更新。不过需要谨慎，因为此时不再保证终止。我们提出的一种简单解决方案是，仅当内涵关系达到不动点时才停止。

为了形式化定义类型为 $\Upsilon \to \Gamma$的层 Σ 的宽松语义，我们引入另一种直接后果算子 L_Σ（用 L 表示“宽松”）。它的定义与 T_Σ（定义4.3）相同，只是将方程（4.A）替换为 $F^{\mathcal{C}}(\boldsymbol{a}) := [\![\theta]\!]^{\mathcal{B}}(\boldsymbol{a})$。内涵关系的定义保持不变。由于我们处理的是有限结构$\mathcal{B}$ ，且内涵关系只能增长，因此存在一个自然数 n，使得$L^n_\Sigma(\mathcal{B})$与$L^{n+1}_\Sigma(\mathcal{B})$在内涵关系上相同。取 n 为满足这一条件的最小数，我们定义 $L^\infty_\Sigma(\mathcal{B}) := L^n_\Sigma(\mathcal{B})$，并称 n 为宽松终止指标。请注意，只有存在内涵关系时，宽松语义才有用；否则，该指标为零。

在宽松语义下，将层 Σ 应用于 Υ-结构 $\mathcal{A}$的结果定义方式与之前相同，但使用 L^∞_Σ 代替 T^∞_Σ，并记作 $\Sigma^L(\mathcal{A})$。使用宽松语义处理各层时，程序（层序列） Π 在 $\mathcal{A}$上的结果记作 $\Pi^L(\mathcal{A})$。

4.3 比较两种语义

不难看出，函数式不动点语义可以在宽松语义中进行模拟:

命题4.5。 对于类型为 $\Upsilon \to \Gamma$ 的每个层 Σ，都存在一个类型为$\Upsilon \to \Gamma'$的层 Σ'，满足 $\Gamma \subseteq \Gamma'$，使得 $\Sigma'^L(\mathcal{A})$ 和 $\Sigma(\mathcal{A})$ 在 Γ 上一致。

证明。 对于 Γ中的关系符号，我们在 Σ' 中使用与 Σ中相同的规则。 Σ中的每条权重函数规则 $F(\boldsymbol{x}) \leftarrow \theta$，在 Σ'中都替换为：如果 $F(\boldsymbol{x}) = \bot$，则为 $F(\boldsymbol{x}) \leftarrow$，否则为 $F(\boldsymbol{x})$。最后，为了捕获函数不动点，对于上述每个 F，我们在 Γ'中加入一个额外关系符号 R_F。 R_F的规则为 $R_F(\boldsymbol{x}) \leftarrow \theta \neq \bot$，从而记录 F 有定义的元组。 □

反过来，函数式不动点语义可以模拟宽松语义。我们可以通过应用时间戳标记和延迟求值技术来证明，这些技术是为带否定的膨胀式 Datalog 开发的 [48]。在这里，宽松不动点中一个阶段的时间戳是由若干元组串接而成，每个内涵关系对应一个元组，并且至少有一个元组是新添加到其内涵关系中的。这类时间戳会成为内涵权重函数的额外参数。然后，我们可以模拟函数

更新，方法是在新的时间戳上定义函数。我们可以在时间戳上维护一个弱序，或者使用一组额外的“延迟”内涵关系名称，从而既能识别前一次迭代的时间戳，也能识别当前迭代的时间戳。模拟层终止后，可以使用后续层投影出最后的时间戳，以获得宽松不动点的最终结果。

我们得出结论：

定理 4.6。 IFP(SUM) 程序在函数式不动点语义下，以及 IFP(SUM)程序在宽松不动点语义下，构成关于加权结构的等价查询语言。

示例 4.7。 示例 4.4 中的程序在宽松不动点语义下，可由以下函数式不动点语义下的程序模拟：

$$
\begin{aligned}
&chosen(k') \leftarrow next(k');\\
&D'(k',i,j) \leftarrow \text{if } \neg next(k') \text{ then } \bot\\
&\qquad\qquad \text{else if } chosen = \emptyset \text{ then}\\
&\qquad\qquad\qquad\qquad \text{if } W(i,j) > W(i,k') + W(k',j)) \text{ then } W(i,k') + W(k',j)\\
&\qquad\qquad\qquad\qquad \text{else } W(i,j)\\
&\qquad\qquad\qquad \text{else if } \exists k(last(k) \wedge D'(k,i,j) > D'(k,i,k') + D'(k,k',j)) \text{ then}\\
&\qquad\qquad\qquad\qquad\qquad \sum_{k:last(k)} D'(k,i,k') + D'(k,k',j)\\
&\qquad\qquad\qquad\qquad \text{else } \sum_{k:last(k)} D'(k,i,j);\\
&D(i,j) \leftarrow \sum_{k:last(k)} D'(k,i,j).
\end{aligned}
$$

这里，$last(k)$ 是 $chosen(x) \wedge \forall x((chosen(x) \wedge x \neq k) \rightarrow ord(x,k))$ 的缩写，它选择前一次迭代中选定的节点。定义 D 的最终规则位于一个单独的层中。

□

我们指出，上述示例比定理 4.6 证明中的一般构造简单得多。事实上，在该示例中，我们可以假设一个全序，从而免费获得时间戳。然而，该定理一般成立。对于从宽松不动点到函数不动点而言，时间戳标记是否必要，这是一个开放问题。具体来说，是否存在这样一个从加权结构到权重函数的查询：它可以在宽松语义下使用仅一元内涵权函数的 IFP(SUM) 表达，但不能在函数式不动点语义下表达?

4.4. 一个范式

本节将在递归理论和有限模型论所研究的经典不动点逻辑框架与语言中考察 IFP(SUM)。我们将证明一个范式，其核心是指出，每个 IFP(SUM) 程序都等价于一个程序

该程序由单个层组成，其中包含一个内涵符号，后接选择和投影。

将 IFP(SUM)重新定义为一种扩展 FO(SUM)的逻辑是标准做法 [41, 18, 25], 因此我们从简介绍。我们在语法中添加一条新的权重项构造规则（3.A）–（3.B）：

$$\theta ::= \mathsf{ifp}\left(F(x_1,\dots,x_{\mathrm{ar}(F)}) \leftarrow \theta\right)(x'_1,\dots,x'_{\mathrm{ar}(F)}), \tag{4.B}$$

其中 F 是一个权重函数符号；注意， θ 可以包含内层 ifp 算子。

此类 ifp 项的自由变量为 $(\mathsf{free}(\theta) - \{x_1,\dots,x_{\mathrm{ar}(F)}\} \cup \{x'_1,\dots,x'_{\mathrm{ar}(F)}\}$。I FP(SUM) 表达式 ξ中的自由（或称外延）关系符号和函数符号，记作 ext(ξ)，其定义方式很直接：令 ifp 项 (4.B) 的 ext 为 ext(θ) $-\{F\}$。如果符号 F 出现在 ξ 的子项 if p $\left(F(\boldsymbol{x}) \leftarrow \theta\right)(\boldsymbol{x}')$ 中，那么它在 ξ 中是内涵的。我们用 int(ξ) 表示 ξ 中所有内涵符号的集合。下文中，对于所有 IFP(SUM) 表达式 ξ，我们都假设 int(ξ) $\cap$ext ξ) $=\emptyset$，且每个内涵符号只受一个 ifp 算子约束。通过重命名内涵符号，这一假设不失一般性。

IFP(SUM) 的语义扩展了 FO(SUM) 的语义。为了定义 $\eta :=$ ifp $\left(F(\boldsymbol{x}) \leftarrow \theta\right)(\boldsymbol{x}')$ 在 $(\mathcal{A},\nu)$ 上的语义，其中 $\mathcal{A}$ 为一个 Υ-结构，且 ext(η) $\subseteq \Upsilon$，我们定义函数 $F(t)$ 的序列 $(F(t))^{t\in\mathbb{N}\cup\{\infty\}}$ 如下。对所有 $\boldsymbol{a} \in A^{\mathrm{ar}(F)}$，令 $F(0)(\boldsymbol{a}) := \bot$；如果 $F(t)(\boldsymbol{a}) = \bot$，则令 $F(t+1)(\boldsymbol{a}) := [\![\theta]\!]^{(\mathcal{A}^{F(t)}_F,\nu^{\boldsymbol{a}}_{\boldsymbol{x}})}$；否则令 $F(t)(\boldsymbol{a}) := \bot$。这里，$\mathcal{A}^{F(t)}_F$ 表示这样一个 $\Upsilon \cup \{F\}$-结构：它在 $\Upsilon \setminus \{F\}$ 中的所有符号上都与 $\mathcal{A}$ 一致，并将 F 解释为 $F(t)$。对于 $t=\infty$，我们注意到，对于每个 $\boldsymbol{a} \in A^{\mathrm{ar}(F)}$，如果存在某个 t 使得 $F(t)(\boldsymbol{a}) \neq \bot$，那么对于所有 $t' \geq t$，都有 $F(t')(\boldsymbol{a}) = F(t)(\boldsymbol{a})$；在这种情况下，我们令 $F^{(\infty)}(\boldsymbol{a}) := F(t)(\boldsymbol{a})$。否则，我们令 $F(\infty)(\boldsymbol{a}) := \bot$。最后，令 $[\![\eta]\!]^{(\mathcal{A},\nu)} := F^{(\infty)}\left(\nu(\boldsymbol{x}')\right)$.

我们将 IFP(SUM) 这一名称同时用于上一节中的逻辑以及此处定义的变体。正如我们接下来将要说明的，这两种逻辑实际上基本相同。若需明确区分二者，我们将第 4.1 节定义的语法称为 IFP(SUM) 层 和 程序，而将此处定义的语法称为 IFP(SUM) （权重）项 和公式。

我们首先注意到，ifp 算子的语义与第 4.1 节中层的语义相容。具体而言，设 $\theta(\boldsymbol{x})$ 是一个 FO(SUM) 项，并考虑 IFP(SUM) 层 $\Sigma := F(\boldsymbol{x}) \leftarrow \theta$，其中 ext($\theta$) $\setminus \{F\} \subseteq \Upsilon$。那么，对于所有 Υ-结构 $\mathcal{A}$，我们有

$$[\![\mathsf{ifp}\left(F(\boldsymbol{x}) \leftarrow \theta\right)]\!]^{\mathcal{A}} = F^{\Sigma(\mathcal{A})}.$$

这里我们将 $[\![$ifp $\left(F(\boldsymbol{x}) \leftarrow \theta\right)]\!]^{\mathcal{A}}$视为从 $A^{\mathrm{ar}(F)}$到 $\mathbb{R}_\bot$的函数，它将元组 $\boldsymbol{a}'$映射为 $[\![$ ifp $\left(F(\boldsymbol{x}) \leftarrow \theta\right)(\boldsymbol{x}')]\!]^{\mathcal{A}}(\boldsymbol{a}') = [\![$ifp $\left(F(\boldsymbol{x}) \leftarrow \theta\right)(\boldsymbol{x}')]\!]^{(\mathcal{A},\nu^{\boldsymbol{a}'}_{\boldsymbol{x}'})}$，对所有赋值 ν均成立。

然而，程序与 ifp 项之间一个显而易见的区别是，每个 ifp 项定义一个内涵权重函数，而程序则可以通过归纳法定义多个内涵关系和权重函数。不过，这并不意味着程序具有更强的表达能力。通过将同时归纳引理的证明 [41, 18] 调整到IFP(SUM) 情形，我们可以证明以下结论。

引理 4.8. 对于每个带有答案符号 S 的IFP(SUM) 程序 Π，都存在一个闭IFP(SUM) 表达式 ξ，使得 $[\![\xi]\!]^{\mathcal{A}} = S^{\Pi(\mathcal{A})}$ 对于所有 Υ-结构 $\mathcal{A}$均成立。

ifp 项与程序之间剩下的区别在于，ifp 运算符可以嵌套，而层只能按顺序组合。不过，我们可以证明如下范式。我们称 选择条件 是等式与不等式的合取。

引理 *4.9*。每个 *IFP(SUM)* 公式 $\varphi(\boldsymbol{x})$，以及每个以关系符号为答案符号的 *IFP(SUM)*程序，都等价于形如 $\exists \boldsymbol{y}\big(\chi(\boldsymbol{y}) \wedge \mathit{ifp}\ \big(F(\boldsymbol{x},\boldsymbol{y}) \leftarrow \theta\big)(\boldsymbol{x},\boldsymbol{y})\big)$ 的公式，其中 χ 是一个选择条件，而 $\theta(\boldsymbol{x},\boldsymbol{y})$ 是一个 *FO(SUM)* 项。此外，每个 *IFP(SUM)* 项 $\eta(\boldsymbol{x})$，以及每个以权重函数符号为答案符号的 *IFP(SUM)*程序，也同样等价于形如 $\mathit{avg}_{\boldsymbol{y}:\chi(\boldsymbol{y})}\ \mathit{ifp}\ \big(F(\boldsymbol{x},\boldsymbol{y}) \leftarrow \theta\big)(\boldsymbol{x},\boldsymbol{y})$ 的项。

这个引理可以沿着膨胀不动点逻辑类似正规形式结果的思路来证明 [18, 第 8]章。证明的思路是，先用同时归纳模拟嵌套的ifp操作，正如 IFP(SUM) 层中的定义。在同时归纳中，我们先逐步进行内层归纳，直到它达到不动点。然后进行外层归纳的一步，再次运行内层归纳，直到它达到不动点；接着进行外层归纳的一步，如此反复，直到外层归纳达到不动点。要使这一方法奏效，关键在于我们能够在FO(SUM) 中检测归纳是否已达到不动点，并设置一个标志，将这一信息告知外层归纳。

推论 4.10。*1.* 每个答案符号为关系符号的 IFP(SUM) 程序，都等价于一个如下形式的 IFP(SUM) 公式

$$\exists \boldsymbol{y}\Big(\chi(\boldsymbol{y}) \wedge \mathsf{ifp}\ \big(F(\boldsymbol{x},\boldsymbol{y}) \leftarrow \theta\big)(\boldsymbol{x},\boldsymbol{y})\Big),$$

其中 χ 是选择条件，θ 是一个FO(SUM) 项。

2. 每个答案符号为函数符号的IFP(SUM) 程序都等价于如下形式的项

$$\operatorname*{avg}_{\boldsymbol{y}:\chi(\boldsymbol{y})}\ \mathsf{ifp}\ \big(F(\boldsymbol{x},\boldsymbol{y}) \leftarrow \theta\big)(\boldsymbol{x},\boldsymbol{y}),$$

其中 χ是选择条件，θ 是一个FO(SUM) 项。

注 *4.11.* 由于我们的动机是神经网络，我们将数值论域固定为实数。不过，本节所述结果可推广到具有聚合 [25, 38] 的任意数值论域。唯一的要求是，不含递归的逻辑能够表示构造 $\mathrm{uniq}_{x:\varphi}\,\theta$，其中 φ 是一个公式，θ 是一个权重项，其语义如下。假设存在 $a \in A$ 使得 $\mathcal{A} \models \varphi(a)$，并且对于所有这样的 a，值 $[\![\theta]\!]^{\mathcal{A}}(a)$ 都相同，记为 $r \in \mathbb{R}_{\perp}$。那么我们定义 $[\![\mathrm{uniq}_{x:\varphi}\,\theta]\!]^{\mathcal{A}}$为这个 r。否则，我们将其定义为 $\perp$。

在 FO(SUM) 中，我们确实可以将其表示为 if $\forall x \forall x'((\varphi(x) \wedge \varphi(x')) \rightarrow \theta(x) = \theta(x'))$ then $\mathrm{avg}_{x:\varphi}\,\theta$ else $\perp$；其中，$\varphi(x')$ 和 $\theta(x')$ 表示将 x' （一个新变量）代入 x 的自由出现处。

5. 多项式时间片段

在有限模型论中，不动点逻辑通常用于刻画多项式时间计算。若能使用 IFP(SUM) 刻画神经网络的多项式性质，那就再好不过了。若要做到这一点，至少在采用传统计算模型的情况下，我们必须将研究范围限制在权重为有理数的加权结构上。但即便如此，事实证明，IFP(SUM) 项也无法在多项式时间内求值，因为其值可能过大，甚至无法用与输入结构规模成多项式关系的空间来表示。

例5.1 ([25])。 考虑以下 IFP(SUM)项 $\eta(x)$:

$$\eta(x) \coloneqq \mathsf{ifp}\,\big(F(x) \leftarrow \mathsf{if}\ \exists y\, E(y,x)\ \mathsf{then}\ (\sum_{y:E(y,x)} F(y)) \cdot (\sum_{y:E(y,x)} F(y))\ \mathsf{else}\ 2\big)(x).$$

那么，如果 $\mathcal{A}$ 是一条长度为 n 的路径，且 a 是这条路径的最后一个顶点，那么 $[\![\eta]\!]^{\mathcal{A}}(a) = 2^{2^n}$。

为了避免上文所示的重复平方，我们将定义一个称为 标量 IFP(SUM) 的片段 sIFP(SUM)，它限制乘法和除法的使用方式。该片段禁止两个都包含内涵函数符号的项相乘。换句话说，当涉及内涵权重函数时，我们只允许数乘；这里对“标量”作宽泛解释，即“以非递归方式定义”。我们还禁止对递归定义的项进行除法。

形式上，对于函数符号集 $\mathcal{F}$， $\mathcal{F}$-标量公式和项的语法完全遵循 3.A、3.B、4.B中IFP(SUM)公式和项的语法规则，但乘法、除法和ifp的规则作如下修改:

- 如果 θ_1 是 $\mathcal{F}$-标量，且 θ_2 是一个含有 $\mathsf{ext}(\theta_2) \cap \mathcal{F} = \emptyset$ 的项，那么 $\theta_1 \cdot \theta_2$、 $\theta_2 \cdot \theta_1$ 和θ_1/θ_2 是 $\mathcal{F}$-标量;
- 如果 θ是 $\mathcal{F} \setminus \{F\}$-标量，那么 $\mathsf{ifp}\ \big(F(\boldsymbol{x}) \leftarrow \theta\big)(\boldsymbol{x}')$ 是 $\mathcal{F}$-标量。

现在，如果一个 IFP(SUM) 表达式 ξ 的所有子表达式都是 $\mathsf{int}(\xi)$-标量，就称它为标量。我们用 sIFP(SUM) 表示 IFP(SUM) 的标量片段。类似地，如果 IFP(SUM) 层 Σ 的规则中的所有子表达式都是 $\mathsf{int}(\Sigma)$-标量，就称该层为标量；如果一个程序的所有层都是标量，就称该程序为标量。

例5.2。 例 5.1 中的项并非标量，因为它包含两个涉及内涵符号的项相乘。例 4.2 和 4.4中的程序是标量。

□

在考虑计算 IFP(SUM) 表达式的复杂性时，我们将注意力限制在具有有理数权重的结构上。为简便起见，我们称之为有理结构。对于有理数 Υ-结构 $\mathcal{A}$，我们用 $\|\mathcal{A}\|$ 表示 $\mathcal{A}$ 的一种编码的位大小。于是

$$\|\mathcal{A}\| = O\big(|A| + \sum_{R\in\Upsilon \text{ relation symbol}} |R^{\mathcal{A}}| + \sum_{\substack{F\in\Upsilon \text{ weight-}\\ \text{function symbol}}} \sum_{\boldsymbol{a}\in A^{\mathrm{ar}(F)}} \|F^{\mathcal{A}}(\boldsymbol{a})\|\big),$$

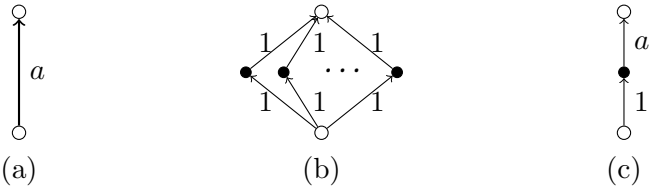


图1：将 (a) 中权重较大的边 $a \in \mathbb{N}$ 拆分为 (b) 中由权重为 1 的边连接的 a 个内部节点。对 (b) 中的网络进行化简，得到网络 (c)

其中，对于整数 $n \in \mathbb{N}$，我们令 $\|n\|$ 为 n 的二进制编码长度；对于既约形式的有理数 $p/q \in \mathbb{Q}$，我们令 $\|p/q\| = \|p\| + \|q\|$。

我们可以证明，标量 IFP(SUM) 具有多项式数据复杂性。

定理 5.3。 存在一种算法，给定一个 sIFP(SUM) 表达式 ξ、一个有理数结构 $\mathcal{A}$ 和一个元组 $\boldsymbol{a} \in A^{|\boldsymbol{x}|}$，可在关于 $\|\mathcal{A}\|$ 的多项式时间内计算 $[\![\xi]\!]^{\mathcal{A}}(\boldsymbol{a})$。

为证明此定理（见附录），我们可以从引理 4.9 的范式出发，该范式可证明保持标量性。证明的关键在于仔细验证：对于递归规则 $F(\boldsymbol{x}) \leftarrow \theta$，$[\![\theta]\!]^{\mathcal{A}}(\boldsymbol{a})$ 的分子和分母仅以线性方式依赖于 $F^{\mathcal{A}}$，这一点可以精确定义。难点在于控制分母在算术运算符作用下的增长。

备注 *5.4*。用参数化复杂性的语言来说，IFP(SUM)-求值问题若以表达式长度为参数，则属于复杂性类 XP。由于 IFP(SUM)包含标准的不动点逻辑 UFP，因此该问题实际上是（统一）XP-hard [19]。 □

即使在无权关系结构上，sIFP(SUM) 的表达能力也强于普通的不动点逻辑 IFP，因为它可以使用求和算子计数可定义集合中的元素个数，再比较这些计数以表达例如多数查询。然而，不出所料，即使对于完整的 IFP(SUM)，我们也有如下结论。

定理 5.5。 存在一个关于图的布尔查询，可在多项式时间内判定，但无法用IFP(SUM)表达。

这一结果可用标准技巧证明；我们在附录中给出一个自成一体的证明。我们证明，双射 pebble 博弈 [30]，是证明带计数的不动点逻辑中不可表达性的标准工具，也可用于证明 IFP(SUM) 中的不可表达性。对于与 FO(SUM) 相当类似的带聚合的逻辑，这一点早已为人所知 [29]。接着，通常的 Cai-F¨urer-Immerman 构造 [16]给出了一个无法用 IFP(SUM) 表达的查询。

然而，我们主要关注神经网络上与模型无关的查询。尽管如此，我们仍无法在 IFP(SUM) 中表达所有此类多项式时间查询。直观地说，这是因为前馈神经网络中的权重可以任意大，而 IFP(SUM) 对于一般的（多项式时间）数值计算而言过于弱小。事实上，甚至可以证明，这个非常简单的查询 $Q_0$where $Q_0(\mathcal{N})$ 为真，当且仅当 $f^{\mathcal{N}}(1)$ 是一个自然数，

无法表达。但情况更糟：我们可以通过将大权重的边拆分成许多小权重的边来避免大权重，如图 1所示。利用上述查询的变体 Q_0，该变体询问 $f^{\mathcal{N}}(1)$ 是否为偶自然数，我们可以证明如下结论。

定理 5.6。 在类 $\mathbf{K}(1,1)$ 上存在一个可在多项式时间内判定的模型无关布尔查询，但即使在所有权重均为1 或0的前馈神经网络上，该查询也无法用 IFP(SUM) 表达。

对大权重的依赖确实需要更细致的分析。为此，~对于每个前馈神经网络 $\mathcal{N}$，我们将定义一个等价的约简 前馈神经网络 $\mathcal{N}$。回顾一下， $b(u)$ 表示节点 u 的偏置，而 $w(v,u)$ 表示网络中边 (v,u) 的权重。对于节点集合 V，我们令 $w(V,u) := \sum_{v\in V} w(v,u)$。

我们通过对 $\mathcal{N}$中节点的深度进行归纳法，在 $\mathcal{N}$ 的节点集合上定义一个等价关系 $\sim$ 。只有深度相同的节点才可能等价。对于输入节点和输出节点 u、 u'，我们令 $u \sim u' \Leftrightarrow u = u'$，因此输入节点和输出节点只与自身等价。现在考虑两个深度相同的隐藏节点 u、 u'，并假设我们已经在所有较小深度的节点上定义了 $\sim$。那么，对于每个较小深度节点的等价类 V，若 $u \sim u'$ $b(u) = b(u')$ 且 $w(V,u) = w(V,u')$。

我们用 $\tilde{u}$ 表示节点 u 的等价类。现在定义：

~**定义5.7**。 约化网络 $\mathcal{N}$ 的节点为 $\tilde{u}$，边为所有来自 $\mathcal{N}$ 的边 (v,u)~所对应的边 $(\tilde{v},\tilde{u})$。我们定义 $b(\tilde{u}) := b(u)$，以及 $w(\tilde{v},\tilde{u}) := \sum_{v'\sim v} w(v',u)$。 $\mathcal{N}$ 的输入节点是 $\mathcal{N}$ 的输入节点所对应的单元素等价类，输出节点也类似。

~通过直接归纳可知，若 $u \sim u'$，则 $f_u^{\mathcal{N}} = f_{u'}^{\mathcal{N}}$，因此 $f^{\mathcal{N}} = f^{\mathcal{N}}$；也就是说，前馈神经网络及其约化网络表示同一个函数。

例5.8。 图 1(b) 中网络的约化结果如图 1(c) 所示。 □

设 $P(X)$ 是一个多项式， $\mathcal{N}$ 是一个前馈神经网络；我们用记号 $|\mathcal{N}|$ 表示网络的节点数。给定 $n = |\mathcal{N}|$，若 $\mathcal{N}$ 的所有节点权重和边权重均为有理数，且具有形式 r/q，其中整数 $r \in \mathbb{Z}$、 $q \in \mathbb{N}$ 满足 $|r|, q \le P(n)$，则称 $\mathcal{N}$ 的权重受 P 阶界限制 。此外，若 $\mathcal{N}$ 的权重受 P 阶界限制，则称 $\mathcal{N}$ 的约化权重受 P 阶界限制 。若存在一个多项式 P，使得类 $\mathbf{K}$ 中的每个网络 $\mathcal{N} \in \mathbf{K}$ 的权重（约化权重）都受 P 阶界限制，则称网络类 $\mathbf{K}$ 的权重（约化权重）受多项式界限制。

例 5.9. 图中描绘的网络类别1(b) 显然具有多项式有界权重（它们全都是 1），但不具有多项式有界约化权重。事实上，图 1(c) 中描绘的类别不具有多项式有界权重（因为 a 可以任意取值，而 $|\mathcal{N}| = 3$）。

□

我们为标量 IFP(SUM) 建立以下完备性结果。

定理 5.10。 设 Q 是在 $\mathbf{K}(*,*)$ 上的多项式时间可计算查询。那么，在每个具有多项式有界约化权重的 $\mathbf{K}' \subseteq \mathbf{K}(*,*)$ 上， Q 都可以用 sIFP(SUM) 表示。

证明（见附录）指出，可以在 sIFP(SUM) 中对网络$\sim\mathcal{N}$ 的节点统一定义一个准序，使其在 $\mathcal{N}$ 上导出一个线性序。~此外，由于 $\mathcal{N}$ 具有多项式有界权重，所有分子和分母都可以用 sIFP(SUM) 公式表示为由 ~ 个节点组成的字典序排列的元组。由此可得到 $\mathcal{N}$ 的一个副本，作为在 $\mathcal{N}$ 中定义的有序有限结构。根据~Immerman-Vardi 定理，我们还可以在 $\mathcal{N}$ 中定义 Q 在 $\mathcal{N}$ 上的答案。由于 Q是模型无关的，这也就是 Q 在 $\mathcal{N}$ 上的答案。

注 *5.11*。 为简单起见，定理 5.10针对不带参数的布尔查询进行了陈述和证明。可以表述并证明一个针对带参数的 r元查询的版本；在这种情况下，我们还需要将参数和结果元组限制为多项式有界的既约有理数。参数和结果元组的候选项作为额外的权重函数传递给 sIFP(SUM)公式。

6. 模型无关查询的复杂性

有了 IFP(SUM) 之后，现在很自然会重新审视定理 3.4 ，并思考能否仅通过将 FO(SUM)替换为 IFP(SUM)，就取消固定深度的限制。然而，我们猜想答案是否定的:

猜想 6.1。 设 $m > 0$ 为一个自然数。存在一个关于$\mathbf{K}(m, 1)$ 的布尔查询，该查询可用 FO($\mathbf{R}_{\text{lin}}, f$) 表示，但无法用IFP(SUM)表示。

虽然我们无法证明这一猜想，但在假设 P $\neq$ NP 的情况下，可以证明相应的标量 IFP(SUM) 猜想。事实上，尽管 sIFP(SUM) 查询可在多项式时间内计算，但即使在 $\mathbf{K}(1,1)$ 上，也存在非常简单的 FO($\mathbf{R}_{\text{lin}}, f$) 句子，能够表达一个 NP-困难查询。

定理 6.2。 判断前馈神经网络 $\mathcal{N} \in \mathbf{K}(1,1)$ 是否计算非零函数是NP-困难的，也就是说，判断是否存在某个 $x \in \mathbb{R}$使得 $f^{\mathcal{N}}(x) \neq 0$。

注意，测试非零性是一种布尔型模型无关查询，可以很容易地表示为 FO($\mathbf{R}_{\text{lin}}, f$) 句子 $\exists x\, f(x) \neq 0$。事实上，该查询也属于 NP[49,命题 1]。在所引用的工作中，Wurm 还证明了判定前馈神经网络 $\mathcal{N} \in \mathbf{K}(*, *)$ 是否为零是 co-NP 困难的（表述为等价性问题）。我们的贡献在于证明了即使输入和输出维度固定为 1，该问题仍然是困难的。证明（见附录）从 3-SAT 归约而来。我们将二进制表示为 $(0.a_1a_2\ldots a_n)_2$, 的 0 到 1 之间的有理数解释为 n 布尔变量的赋值。然后，我们构造一个网络 $\mathcal{N}$，使其在这些数上模拟关于这些变量的给定 3-CNF 公式。

7. 迭代转导

鉴于猜想6.1，我们如何超越IFP(SUM)，定义一种关于加权结构的逻辑，使其能够表达所有 FO($\mathbf{R}$ f查询，而不必固定网络深度？我们可以从定理3.4 的证明 [28]中获得启发。该证明的第一步

是将任意给定的（固定深度）前馈神经网络 $\mathcal{N} \in \mathbf{K}(m, 1)$ 映射为一种结构，用来表示分段线性函数 $f^{\mathcal{N}} : \mathbb{R}^m \to \mathbb{R}$ 的几何形状。这种几何形状由一个将 $\mathbb{R}^m$ 划分为多胞体的超平面排列，以及每个多胞体上的一个仿射函数构成。后续步骤会根据我们希望表达的 FO($\mathbf{R}_{\text{lin}}$, f) 查询 ψ 中变量的数量，将所得几何形状嵌入更高维空间，并构造该空间的柱形胞腔分解。所得胞腔分解既与 $f^{\mathcal{N}}$ 兼容，也符合 ψ 施加的约束，因此我们可以在 FO(SUM) 中表达 ψ 。

刚才描述的步骤将加权结构映射为加权结构；这些转- 导可以通过调整经典的模型论方法——在一种逻辑理论中解释另一种逻辑理论——用FO(SUM)来表达 [31]。这类解释（也称为转- 导 [17, 27] 或翻译 [28]）通过将输出结构的元素定义为输入结构中元组的等价类，并用公式和权重项定义关系和权重函数，从一个词汇表中的结构映射到另一个词汇表中的结构。

重要的观察是，在证明上述步骤可以表示为 FO(SUM) 转导时，固定深度ℓ的 $\mathcal{N}$ 这一假设只对第一步至关重要。该构造逐层进行，其中的转导迭代固定的 ℓ 次。因此，只要能够对转导进行无界次数的迭代，我们就能取消固定深度的限制；实际上， $O(\ell)$次迭代就足够了。

形式上，我们可以将从词汇表 Υ 到词汇表 Γ 的迭代FO(SUM) 转导定义为一对 (τ, φ)，其中 τ 是一个从 $\Upsilon \cup \Gamma$ 到其自身的 FO(SUM) 转导，而 φ 是 $\Upsilon \cup \Gamma$ 上的一个闭 FO(SUM) 公式。给定一个 Υ-结构 $\mathcal{A}$，其语义是先用 Γ 的符号扩充 $\mathcal{A}$ （在所有位置将它们初始化为空或未定义）。然后，重复应用 τ，直到 φ 成立。

即使在宽松语义下，这与 IFP(SUM) 程序的区别也在于，转导可以扩展论域，并且可以任意更改（也可以缩小）关系和函数。这种情况可以在迭代转导的每一步中发生。

我们可以证明，迭代转导在顺序复合下封闭。要表达一个查询，我们可以指定一个答案符号，就像为 IFP(SUM) 程序所做的那样。由此得出：

命题 7.1。 设 m 为一个自然数。每个在$\mathbf{K}(m, 1)$ 上的布尔FO($\mathbf{R}_{\text{lin}}, f$) 查询，都可以由一个迭代 FO(SUM) 转导表达；该转导在任意 $\mathcal{N} \in \mathbf{K}(m,1)$上只迭代 $O(\ell)$ 次，其中 ℓ 是 $\mathcal{N}$ 的深度。

注 *7.2*。 迭代转导可谓一种“核选项”，因为它让人联想到一阶逻辑扩展 while 循环和对象创建的情形；这一扩展曾在 20 世纪 90 年代受到研究 [3, 2, 14, 15]。这类扩展通常会使查询语言在有限非加权结构上达到图灵完备。我们可以证明（证明从略），类似地，迭代 FO(SUM) 转导（与上述命题一样，不对迭代次数设深度界）在有理数加权结构上也是图灵完备的。

8. 结论

我们探讨了对深度神经网络进行模型无关查询的方法及其障碍。在固定深度情形下，FO(SUM) 已具有相当强的表达能力，而当前的主要挑战在于寻找良好的实现策略。若深度不受限制，在表达能力、理论复杂度和查询语言设计方面也会面临挑战。我们引入了一种语言 IFP(SUM)，使我们能够评估无界深度的神经网络，并且更一般地，能够表达关于此类网络的丰富查询集合。

一些有趣的问题仍有待解决。一个非常具体的问题，即使独立于神经网络应用，也是刻画 IFP(SUM)（不带标量限制）的数据复杂度，即使是在无权结构上。我们还遇到了这样一个问题：与函数不动点语义相比，宽松不动点语义是否能够保持更低的元数。另一个有趣的问题是，如何调整尺度逻辑FO($\mathbf{R}$ f)，使其能够在任意函数 f：$\mathbb{R}^m \to \mathbb{R}^p$ 其中 m 和 p 并非预先固定。在布尔模型上，已经存在针对此问题的非常优雅的语言 [8, 7]。

最后，当然，我们还应研究对其他机器学习（ML）模型的查询，例如数值决策树、Transformer 模型和图神经网络。机器学习、逻辑和数据库理论领域已经积累了大量工作，用于理解 ML 模型的逻辑表达能力。然而，通过逻辑方法查询这些模型是一个截然不同的主题 [39]。

参考文献

[1] S. Abiteboul、R. Hull 和 V. Vianu。《数据库基础》。Addison-Wesley，1995。[2] S. Abiteboul 和 P.C. Kanellakis。“对象标识作为一种查询语言原语”。载于：*ACM*会刊 45.5（1998），第 798–842 页。[3] S. Abiteboul 和 V. Vianu。“用于数据库查询与更新的 Datalog 扩展”。载于：《计算机与系统科学杂志》43.1（1991），第 62–124 页。[4] M. Abo Khamis、H.Q. Ngo 和 A. Rudra。“数据库内部的函数调度”。载于：《*SIGMOD Record*》46.1（2017），第 6–13 页。[5] Abo Khamis 等人。“（预）半环上的 Datalog 收敛性”。载于：*ACM*会刊71.2（2024），8:1–8:55。[6] A. Albarghouthi。“神经网络验证导论”。载于：《编程语言基础与趋势》7.1–2（2021），第 1–157 页。 URL: https://verifieddeeplearning.com。

[7] M. Arenas 等人。“一种用于解释决策树的统一语言”。载于：《知识表示与推理原理》第 21 届国际会议论文集。P. Marquis、M. Ortiz 和 M. Pagnucco 编。IJCAI组织，2024 年，第 60–70 页。[8] M. Arenas 等人。“模型可解释性符号语言基础”。载于：第 35 届神经信息处理系统年会论文集。M. Ranzato 等人编。2021 年，第 11690–11701 页。[9] Steffen van Bergerem 和 Nicole Schweikardt。“学习由权重聚合逻辑描述的概念”。载于：第 29 届 EACSL 计算机科学逻辑年会（CSL 2021），卢布尔雅那，斯洛文尼亚（线上会议），2021 年 1 月 25–28 日。Christel Baier 和 Jean Goubault-Larrecq 编。第 183 卷。LIPIcs。Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik，2021 年，10:1–10:18。 DOI: 10.4230/LIPICS.CSL.2021.10。[10] Steffen van Bergerem 和 Nicole Schweikardt。“带计数和权重聚合的一阶逻辑的 VC 维数”。载于：第 33 届 EACSL 计算机科学逻辑年会（CSL 2025），阿姆斯特丹，荷兰，2025 年 2 月 10–14 日。J¨org Endrullis 和 Sylvain Schmitz 编。第 326 卷。LIPIcs。Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik，2025 年，15:1–15:17。 DOI：10.4230/LIPICS.CSL.2025.15。[11] J. Bochnak、M. Coste 和 M.-F. Roy。《实代数几何》。Springer-Verlag，1998。[12] Ch. Brix 等人。第五届神经网络验证国际竞赛（VNN-COMP）：摘要与结果。arXiv:2412.19985。2024。[13] P. Buneman、J. Cheney 和 S. Vansummeren。“查询与更新语言中隐式溯源的表达能力”。载于：ACM数据库系统汇刊 33.4（2008），28:1–28:47。[14] J. Van den Bussche 和 J. Paredaens。“基于对象的数据模型中复值的表达能力”。载于：《信息与计算》120（1995），第 220–236 页。[15] J. Van den Bussche 等人。“对象创建型数据库转换语言的完备性”。载于：ACM会刊 44.2（1997），第 272–319 页。[16] J.-Y. Cai、M. F¨urer 和 N. Immerman。“图识别所需变量数量的最优下界”。载于：组合数学 12.4（1992），第 389–410 页。[17] B. Courcelle 和 J. Engelfriet。《图结构与单子二阶逻辑》。剑桥大学出版社，2012。[18] H.-D. Ebbinghaus 和 J. Flum。《有限模型论》。第二版。Springer，1999。[19] J¨org Flum 和 Martin Grohe。《参数化复杂性理论》。Springer，2006。

[20] Vincent Froese、Moritz Grillo 和 Martin Skutella。“ReLU神经网络的单射性与验证的复杂性（扩展摘要）”。载于：第三十八届年度学习理论会议，2025年7月30日至4日，法国里昂。Nika Haghtalab 和 Ankur Moitra 编。第291卷。《机器学习研究论文集》。PMLR，2025年，第2188–2189页。 URL: https://proceedings.mlr.press/ v291/froese25a.html。[21] F. Geerts。“图学习的查询语言视角”。载于：第42届ACM数据库原理研讨会论文集。ACM，2023年，第373–379页。[22] F. Geerts 和 J. Van den Bussche。“带注释数据库查询语言的关系完备性”。载于：《计算机与系统科学杂志》77.3（2011），第491–504页。[23] F. Geerts 和 J.L. Reutter。“图神经网络的表达能力与近似性质”。载于：第10届国际学习表征会议。OpenReview.net，2022年。[24] M. Gerarts、J. Steegmans 和 J. Van den Bussche。“SQL4NN：将模型作为数据进行验证和表达性查询”。载于：第9届端到端机器学习数据管理研讨会论文集。ACM，2025年，10:1–10:5。[25] E. Gr¨adel 和 Y. Gurevich。“元有限模型论”。载于：《信息与计算》140.1（1998），第26–81页。[26] T.J. Green、G. Karvounarakis 和 V. Tannen。“溯源半环”。载于：第26届ACM数据库系统原理研讨会论文集。ACM，2007年，第31–40页。[27] M. Grohe。《描述复杂性、规范化与可定义图结构理论》。第47卷。《逻辑讲义》。剑桥大学出版社，2017年。[28] M. Grohe 等人。“神经网络查询语言”。载于：第28届国际数据库理论会议论文集。S. Roy 和 A. Kara 编。第328卷。《莱布尼茨国际信息学论文集》。Dagstuhl城堡—莱布尼茨信息学中心，2025年，9:1–9:18。[29] L. Hella 等人。“带聚合算子的逻辑”。载于：《ACM会刊》48.4（2001），第880–907页。[30] Lauri Hella。“多项式时间中的逻辑层级”。载于：《信息与计算》129（1996），第1–19页。[31] W. Hodges。《模型论》。剑桥大学出版社，1993年。[32] N. Immerman。“可在多项式时间内计算的关系查询”。载于：《信息与控制》68（1986），第86–104页。[33] P.C. Kanellakis、G.M. Kuper 和 P.Z. Revesz。“约束查询语言”。载于：《计算机与系统科学杂志》51.1（1995年8月），第26–52页。[34] Ph.G. Kolaitis 和 Ch.H. Papadimitriou。“为什么不采用不动点否定？”载于：《计算机与系统科学杂志》43.1（1991），第125–144页。

[35] G. Kuper、L. Libkin 和 J. Paredaens 编。《约束数据库》。Springer，2000年。[36] L. Libkin。《有限模型论基础》。Springer，2004年。[37] L. Libkin。“嵌入式有限模型与约束数据库”。载于：《有限模型论及其应用》。Springer，2007年。第5章。[38] L. Libkin。“SQL的表达能力”。载于：《理论计算机科学》296（2003），第379–404页。[39] J. Marques-Silva。“机器学习中基于逻辑的可解释性”。载于：《推理网：因果关系、解释与声明式知识》。L.E. Bertossi 和 G. Xiao 编。第13759卷。《计算机科学讲义》。Springer，2023年，第24–104页。[40] Ch. Molnar。《可解释机器学习：使黑箱模型可解释的指南》。第二版。2022年。 URL：https : // christophm . github. io / interpretable-ml-book。[41] Y.N. Moschovakis。《抽象结构上的初等归纳法》。North-Holland，1974年。[42] N. Pelov、M. Denecker 和 M. Bruynooghe。“带聚合的逻辑程序的良基语义与稳定语义”。载于：《逻辑编程理论与实践》7.3（2007），第301–353页。[43] M.L. Ribeiro、S. Singh 和 C. Guestrin。“‘我为什么应该信任你？’：解释任意分类器的预测”。载于：第22届ACM SIGKDD知识发现与数据挖掘国际会议论文集。B. Krishnapuram、M. Shah、A.J. Smola 等编。ACM，2016年，第1135–1144页。[44] C. Rudin。“停止解释用于高风险决策的黑箱机器学习模型，改用可解释模型”。载于：《自然·机器智能》1（2019），第206–215页。[45] S. Shalev-Shwartz 和 S. Ben-David。《理解机器学习：从理论到算法》。剑桥大学出版社，2014年。[46] S. Torunczyk。“稀疏数据库上的聚合查询”。载于：第39届ACM数据库原理研讨会论文集。ACM，2020年，第427–443页。[47] M. Vardi。“关系查询语言的复杂性”。载于：第14届ACM计算理论研讨会论文集。1982年，第137–146页。[48] V. Vianu。“未链接的Datalog”。载于：第40届ACM数据库原理研讨会论文集。ACM，2021年，第57–69页。[49] Adrian Wurm。“神经网络的鲁棒性验证”。载于：《约束编程、人工智能与运筹学的集成——第21届国际会议，CPAIOR 2024，瑞典乌普萨拉，2024年5月28日至31日，论文集，第二部分》。Bistra Dilkina 编。第14743卷。《计算机科学讲义》。Springer，2024年，第263–278页。 DOI：10.1007/978-3-031-60599-4_18。

A. 证明

在证明中，我们使用记号 $[n]$，其中 n 为自然数，表示 $\{1, \ldots, n\}$。

A.1. 定理 4.6 的证明

我们将分三个部分证明，从宽松语义到函数式语义存在一种翻译。首先，在假设结构始终至少包含两个元素的情况下，我们证明这种翻译存在。然后，在假设结构少于两个元素的情况下，我们证明翻译存在；最后，我们展示如何将二者结合起来，得到一种适用于所有结构的翻译。

引理 A.1. 对于每个类型为 $\Upsilon \to \Gamma$ 的层 Σ，都存在一个类型为$\Upsilon \to \Gamma'$的层 Σ'，且满足 $\Gamma \subset \Gamma'$，使得对于任意论域为 A 且$|A| \geq 2$的 Υ-加权结构 $\mathcal{A}$，将 $\Sigma'(\mathcal{A})$限制在 $\Upsilon \cup \Gamma$ 的符号上后，所得结果等于 $\Sigma^L(\mathcal{A})$。

证明。 该证明的思路是：使用函数式语义中的两个阶段，模拟宽松语义下单个阶段的求值。我们首先更新所有内涵关系，然后使用内涵关系的旧值更新内涵权重函数。此外，由于在宽松语义下，内涵权重函数可以在每个阶段被完全重写，因此我们按照之前的解释，用时间戳对权重函数进行参数化。对于时间戳，我们将使用在宽松语义下相应阶段中加入内涵关系的元组集合。正是为了构造这个时间戳，我们才需要用函数式语义下的两个阶段来模拟宽松膨胀语义中的一个阶段。

令 $R_1, \ldots, R_n$ 为 Γ 中的所有关系名称。首先，我们将构造一个公式 φ_{all}，它接受由 $R_1, \ldots, R_n$ 中每个关系恰好一个元组串接而成的所有字符串。为允许某些关系为空，我们将通过在元组前添加一对额外元素来编码元组，从而加入“空元组”。如果这对元素的值相同，则表示“没有元组”；如果前缀对中的元素值不同，则表示一个实际元组。我们可以将 φ_{all} 写成如下形式

$$\varphi_{\text{all}}(x_1, \ldots, x_{s_{n+1}}) := \bigwedge_{i=1}^{n} (x_{1+s_i} = x_{2+s_i} \vee (x_{1+s_i} \neq x_{2+s_i} \wedge R_i(x_{3+s_i}, \ldots, x_{s_{i+1}})))$$

其中 $s_i := \sum_{j=1}^{i-1} (\text{ar}(R_i) + 2)$ 对于 $i \in \{1, \ldots, n+1\}$ 均成立。该公式接受的元组将构成我们的时间戳。

接下来，我们将在 Σ' 中添加以下关系规则，以跟踪 φ_{all} 的历史以及所有 $R_1, \ldots, R_n$：

$R_{\text{all}}(x_1, \ldots, x_{s_{n+1}}) \leftarrow \varphi_{\text{all}}(x_1, \ldots, x_{s_{n+1}}) R_i^{\text{old}}(x_1, \ldots, x_{\text{ar}(R_i)}) \leftarrow R_i($
$x_1, \ldots, x_{\text{ar}(R_i)})$ 对每个 $i \in \{1, \ldots, n\} R_{\text{all}}^{\text{old}}(x_1, \ldots, x_{s_{n+1}}) \leftarrow R_{\text{all}}(x_1,$
$\ldots, x_{s_{n+1}}) R_{\text{all}}^{\text{old-old}}(x_1, \ldots, x_{s_{n+1}}) \leftarrow R_{\text{all}}^{\text{old}}(x_1, \ldots, x_{s_{n+1}})$

现在，我们定义公式 $\varphi_{\text{all}}^{\text{new}}$、 $\varphi_{\text{all}}^{\text{old-new}}$ 和 $\varphi_{\text{all}}^{\text{old-old-new}}$，其中包含将添加到 R_{all} 此阶段的元组，以及分别在一阶段和两阶段之前添加到其中的元组。定义如下

$$\varphi_{\text{all}}^{\text{new}}(x_1,\dots,x_{s_{n+1}}) := \varphi_{\text{all}}(x_1,\dots,x_{s_{n+1}}) \wedge \neg R_{\text{all}}(x_1,\dots,x_{s_{n+1}})$$
$$\varphi_{\text{all}}^{\text{old-new}}(x_1,\dots,x_{s_{n+1}}) := R_{\text{all}}(x_1,\dots,x_{s_{n+1}}) \wedge \neg R_{\text{all}}^{\text{old}}(x_1,\dots,x_{s_{n+1}})$$
$$\varphi_{\text{all}}^{\text{old-old-new}}(x_1,\dots,x_{s_{n+1}}) := R_{\text{all}}^{\text{old}}(x_1,\dots,x_{s_{n+1}}) \wedge \neg R_{\text{all}}^{\text{old-old}}(x_1,\dots,x_{s_{n+1}})$$

w_1、.....、w_m 是 Γ 中所有的权重函数名称。对于每个 $i \in \{1,\dots,m\}$，我们将定义一个新的、元数为 $s_{n+1}+\mathsf{ar}(w_i)$ 的内涵权重函数名称 w_i'，它表示在将前 s_{n+1} 个变量添加到 R_{all} 的阶段中的权重函数 w_i 。对于每个 $i \in \{1,\dots,m\}$，令 $w_i(y_1、.....、y_{\text{ar}(w_i)}) \leftarrow t_i$ 为 Σ 中的权重函数规则。我们将 t_i' 定义为权重项 t_i ，其中每次出现的 R_j 都替换为对应于每个 $j \in \{1,\dots,n\}$ 的 R_j^{old}，而每次出现的 $w_k(z_1、.....、z_{\text{ar}(w_k)})$ 都替换为

$$\operatorname*{avg}_{p_1,\dots,p_{s_{n+1}}:\varphi_{\text{all}}^{\text{old-old-new}}(p_1,\dots,p_{s_{n+1}})} w_k'(p_1,\dots,p_{s_{n+1}},z_1,\dots,z_{\text{ar}(w_k)})$$

对于每个 $k \in \{1,\dots,m\}$。现在，我们将以下权重函数规则添加到 Σ'

$$w_i'(x_1,\dots,x_{s_{n+1}},y_1,\dots,y_{\text{ar}(w_i)}) \leftarrow \mathsf{if}\ \varphi_{\text{all}}^{\text{new}}(x_1,\dots,x_{s_{n+1}})\ \mathsf{then}\ t_i'\ \mathsf{else}\ \bot$$

这确保了 w_i' 只能在更新内涵权重函数的阶段中更新。这是因为 $\varphi_{\text{all}}^{\text{new}}$ 只有在前一阶段有元组被添加到某个内涵关系时才包含元组，因为更新内涵关系的阶段之前，会有一个只更新内涵权重函数的阶段。

接下来，我们将为 $R_1,\dots,R_n$ 添加关系规则。为此，我们需要确保在更新内涵权重函数时，不更新内涵关系。为此，我们将公式 $\varphi_{\text{weight-update}}$ 定义如下:

$$\varphi_{\text{weight-update}} := \exists x_1,\dots,x_{s_{n+1}}\ \varphi_{\text{all}}^{\text{new}}(x_1,\dots,x_{s_{n+1}})$$

设 $R_i(x_1,\dots,x_{\text{ar}(R_i)}) \leftarrow \varphi_i(x_1,\dots,x_{\text{ar}(R_i)})$ 是 Σ 中针对每个 $i \in \{1,\dots,n\}$ 的 R_i 关系规则。我们将 $\varphi_i'(x_1,\dots,x_{\text{ar}(R_i)})$ 定义为公式 φ_i，其中每次出现的 $w_j(y_1,\dots,y_{\text{ar}(w_j)})$ 都替换为

$$\operatorname*{avg}_{p_1,\dots,p_{s_{n+1}}:\varphi_{\text{all}}^{\text{old-new}}(p_1,\dots,p_{s_{n+1}})} w_j'(p_1,\dots,p_{s_{n+1}},y_1,\dots,y_{\text{ar}(w_j)})$$

对每个 $k \in \{1,\dots,m\}$。现在我们将以下关系规则添加到 Σ'

$$R_i(x_1,\dots,x_{\text{ar}(R_i)}) \leftarrow \neg\varphi_{\text{weight-update}} \wedge \varphi_i'(x_1,\dots,x_{\text{ar}(R_i)})$$

对每个 $i \in \{1,\dots,n\}$。

现在，我们终于要为 w_1、.....、w_m 添加权重函数规则。在此之前，我们需要定义公式 φ_{stop}，该公式仅在 R_1、.....、R_n 已停止更新时为真

。公式 $\varphi_{\text{all}}^{\text{new}}$ 和 $\varphi_{\text{all}}^{\text{old-new}}$ 分别跟踪将添加到 R_{all} 中的新元组，以及在前 1 个阶段添加到其中的元组。如果在关系更新阶段没有更新任何关系，那么在下一个阶段（即权重更新阶段）中，公式 $\varphi_{\text{all}}^{\text{new}}$ 将为空。仅凭这一点不足以检查关系是否已停止增长，因为在每个关系更新阶段中，$\varphi_{\text{all}}^{\text{new}}$ 都为空，这是由于关系不会在前一个权重更新阶段中更新。因此，我们可以将 φ_{stop} 定义为

$$\varphi_{\text{stop}} := \neg\exists x_1, \ldots, x_{s_{n+1}}(\varphi_{\text{all}}^{\text{new}}(x_1, \ldots, x_{s_{n+1}}) \vee \varphi_{\text{all}}^{\text{old-new}}(x_1, \ldots, x_{s_{n+1}}))$$

接下来，由于 φ_{stop} 会在某个权重更新阶段变为真，我们将使用 $\varphi_{\text{all}}^{\text{old-old-new}}$来确定最终阶段的标识符。因此，我们会向 Σ'添加以下规则

$$w_i(x_1, \ldots, x_{\text{ar}(w_i)}) \leftarrow$$
$$\text{if } \varphi_{\text{stop}} \text{ then } \operatorname*{avg}_{p_1,\ldots,p_{s_{n+1}}:\varphi_{\text{all}}^{\text{old-old-new}}(p_1,\ldots,p_{s_{n+1}})} w_i'(p_1, \ldots, p_{s_{n+1}}, x_1, \ldots, x_{\text{ar}(w_i)}) \text{ else } \bot$$

对于每个 $i \in \{1, \ldots m\}$，其中 b_i 是 w_i的元数。 □

为了完成从宽松语义到函数式语义的翻译证明，我们需要给出一种适用于论域大小小于 2 的加权结构的翻译。不过，在此之前，我们先简要定义有界宽松终止指标。

定义 A.2. 设 Σ 是类型为 $\Upsilon \to \Gamma$ 的一个 IFP(SUM) 层，且 $\mathcal{C}$ 是 Υ-加权结构的一个类。如果存在某个 n，使得对于每个 $\mathcal{A} \in \mathcal{C}$，$\Sigma(\mathcal{A})$的宽松终止指标小于或等于$n$，则称宽松终止指标是 $\mathcal{C}$-有界的 。

引理 A.3. 对于任意类型为 $\Upsilon \to \Gamma$ 的 IFP(SUM) 层 Σ 以及由 Υ-加权结构组成的类 $\mathcal{C}$，若宽松终止指标是 $\mathcal{C}$-有界的，则存在一组 FO(SUM)公式和权重项，使得对每个 $\mathcal{A} \in \mathcal{C}$，以下结论成立：

- 对于每个关系名称 $R \in \Gamma$，都存在一个与 R 元数相同的公式 φ_R，使得 $\varphi_R^{\mathcal{A}}$ 与 $R^{\Sigma^L(\mathcal{A})}$一致。
- 对于每个权重函数名称 $w \in \Gamma$，都存在一个自由变量数量与 w 的元数相同的权重项 t_w，使得 $t_w^{\mathcal{A}}$ 与 $w^{\Sigma^L(\mathcal{A})}$一致。

证明。设 n 是一个自然数，小于或等于 Σ的宽松终止指标上界加一。首先，我们将通过归纳证明：对于每个关系名称 $R \in \Gamma$，都可以写出一个公式 φ_R^n；对于每个权重函数名称 $w \in \Gamma$，都可以写出一个权重项 t_w^n，使得 φ_R^n 与 $R^{\mathcal{B}_n}$ 一致，并且对于每个 Υ-加权结构 $\mathcal{A}$，t_w^n 与 $w^{\mathcal{B}_n}$ 一致，其中 $\mathcal{B}_n := L_\Sigma^n(\mathcal{A})$。对于基例，即 $n = 0$ 的情形，这显然成立，因为每个项都是 $\bot$，每个公式都只是 false。设 $R \in \Gamma$ 是一个关系名称，且 $R(x_1, \ldots, x_m) \leftarrow \varphi(x_1, \ldots, x_m)$ 是它在 Σ 中的规则。那么对于任意 $n > 0$，φ_R^n 等于 φ，其中每个关系 $R' \in \Gamma$ 的每次出现都替换为 $\varphi_{R'}^{n-1}$，每个权重函数 $w' \in \Gamma$ 都替换为 $t_{w'}^{n-1}$；这些都是通过归纳构造出来的。类似地，我们可以为每个权重函数名称 $w \in \Gamma$ 构造 t_w^n。

接下来，我们将证明可以构造一个公式 $\varphi^{(n)}$ 宽松终止指标，使得对于每个 Υ-加权结构，$\mathcal{A} \models \varphi^{(n)}$ 宽松终止指标 当且仅当 n 是在 $\mathcal{A}$ 上求值的 Σ 的宽松终止指标。我们仍将通过归纳来证明。设 Γ 中所有关系名称组成的集合为 $\{R_1, \ldots, R_m\}$，它们各自的元数为 $a_1, \ldots, a_m$。对于基例，即 $n = 0$ 的情形，我们可以将公式简单地定义为

$$\varphi^{(0)}_{\text{lti}} := \bigwedge_{R_i \in \{R_1, \ldots, R_m\}} \neg \exists x_1, \ldots, x_{\text{ar}(R_i)}\, \varphi^1_R(x_1, \ldots, x_{\text{ar}(R_i)})$$

对于 $n > 0$ 的情况，我们可以将公式定义为

$$\begin{aligned}\varphi^{(n)}_{\text{lti}} :=& \Big(\bigwedge_{R_i \in \{R_1, \ldots, R_m\}} \neg \exists x_1, \ldots, x_{\text{ar}(R_i)} (\varphi^n_R(x_1, \ldots, x_{\text{ar}(R_i)}) \wedge \neg \varphi^{n-1}_R(x_1, \ldots, x_{\text{ar}(R_i)})) \Big) \\ & \wedge \Big(\bigwedge_{i=0}^{n-1} \neg \varphi^{(i)}_{\text{lti}} \Big)\end{aligned}$$

其中，所有 $\varphi^{(i)}$ lti 均针对所有 $i < n$ 归纳定义。

最后，我们可以结合上述两个证明，构造所需的公式和权重项，如下所示

$$\begin{aligned}\varphi_{R_i}(x_1, \ldots, x_{\text{ar}(R_i)}) &:= \bigvee_{k=0}^{T} (\varphi^{(k)}_{\text{lti}} \wedge \varphi^k_{R_i}(x_1, \ldots, x_{\text{ar}(R_i)})) \\ t_{w_j}(x_1, \ldots, x_{\text{ar}(w_j)}) &:= \mathsf{if}\ \varphi^{(0)}_{\text{lti}}\ \mathsf{then}\ t^0_{w_j}(x_1, \ldots, x_{\text{ar}(w_j)}) \\ &\quad\ \mathsf{else\ if}\ \varphi^{(1)}_{\text{lti}}\ \mathsf{then}\ t^1_{w_j}(x_1, \ldots, x_{\text{ar}(w_j)}) \\ &\quad\ \vdots \\ &\quad\ \mathsf{else\ if}\ \varphi^{(T)}_{\text{lti}}\ \mathsf{then}\ t^T_{w_j}(x_1, \ldots, x_{\text{ar}(w_j)}) \\ &\quad\ \mathsf{else}\ \bot\end{aligned}$$

对于每个元数为 $\text{ar}(R_i)$ 的关系名称 $R_i \in \Gamma$，以及每个权重函数名称 $w_j \in \Gamma$，其中 T 是 Σ 的宽松终止指标的上界。 □

命题 A.4。对于任意类型为 $\Upsilon \to \Gamma$ 的 IFP(SUM) 层 Σ，存在一个类型为 $\Upsilon \to \Gamma'$ 的 IFP(SUM) 层 Σ'，满足 $\Gamma \subset \Gamma'$，使得对于任意 Υ-加权结构 $\mathcal{A}$，都有 $\Sigma'(\mathcal{A})$限制在 $\Upsilon \cup \Gamma$ 上后，等于 $\Sigma^L(\mathcal{A})$。

证明。引理 A.1 和引理 A.3 可以合并为同一个层。设 R_i 是 Γ 中的一个关系，并设 $R_i(x_1, \ldots, x_{\text{ar}(R_i)}) \leftarrow \varphi^{\text{big}}_{R_i}(x_1, \ldots, x_{\text{ar}(R_i)})$ 是其在引理 A.1 所对应层中的关系规则。同样，设 $t^{\text{big}}_{w_j}\ x_1\ \ldots\ x_{\text{ar}(w_j} \leftarrow \Gamma$ 是 Γ 中的一个权重函数，并设 (, ,)) (, ,)). 我们首先将 Σ' 中不属于引理 A.1 所对应层的符号的所有规则添加进去。然后将以下规则添加到 Σ' 中

$R_i(x_1,\dots,x_{\mathrm{ar}(R_i)}) \leftarrow$
$\quad((\exists y_1,y_2\, y_1 \neq y_2) \wedge \varphi^{\mathrm{big}}_{R_i}(x_1,\dots,x_{\mathrm{ar}(R_i)}))$
$\quad\vee(\neg(\exists y_1,y_2\, y_1 \neq y_2) \wedge \varphi^{\mathrm{small}}_{R_i}(x_1,\dots,x_{\mathrm{ar}(R_i)}))$
$w_j(x_1,\dots,x_{\mathrm{ar}(w_j)}) \leftarrow$
$\quad$ if $\exists y_1,y_2\, y_1 \neq y_2$ then $t^{\mathrm{big}}_{w_i}(x_1,\dots,x_{\mathrm{ar}(w_j)})$ else $t^{\mathrm{small}}_{w_i}(x_1,\dots,x_{\mathrm{ar}(w_j)})$

对于每个这样的 R_i 和 w_j 在 Γ中，其中 $\varphi^{\mathrm{small}}_{R_i}$和 $t^{\mathrm{small}}_{w_j}$分别是 FO(SUM) 公式和权重项，分别对应于引理 R_i 和 w_i 中的 A.3。由于只有在论域的大小小于或等于 1 时才会考虑这些公式和权重项，我们知道，在这种情况下，宽松终止指标受关系数量的界定，因此可以构造这些公式和权重项。 □

A.2. 引理4.8的证明

显然，只需证明对于仅含一个层的程序，该结论成立。此外，由于在仅含一个元素的结构中，所有 IFP(SUM) 程序都很容易看出等价于 FO(SUM) 表达式，因此不失一般性，我们只考虑至少含有 2 个元素的结构$\mathcal{A}$。

设 Σ 是类型为 $\Upsilon \to \Gamma$ 的一个层，它由规则 $R_i(\boldsymbol{x}_i) \leftarrow \psi_i$ （其中 $i \in [k]$ ）和 $F_j(\boldsymbol{x}_j) \leftarrow \eta_j$ （其中 $j \in [\ell]$）组成。不失一般性，我们假设 $k,\ \ell \geq 1$；否则，我们引入虚设规则。对于 $i \in [k]$，令 r_i 为 R_i的元数；对于 $j \in [\ell]$，令s_j 为 F_j的元数。此外，令 $r := \max\{r_1,\dots,r_k,s_1,\dots,s_\ell\}$。

对于所有 $i \in [k+\ell]$，令

$$\chi_i(z_1,\dots,z_{k+\ell}) := z_i \neq z_{i'} \wedge \bigwedge_{j\in[k+\ell]\setminus\{i\}} z_j = z_{i'},$$

其中，$i' = 1$ 当 $i \neq 1$ 时，$i' = 2$ 当 $i = 1$时。在下文中，$\boldsymbol{z}$ 始终遍历（$k+\ell$）元组（z_1、……、$z_{k+\ell}$）。我们将使用（$k+\ell$）元组 $\boldsymbol{c}$表示 $[k+\ell]$中的索引，其中$\boldsymbol{c}$ 表示 i ，当且仅当它满足 χ_i（$\boldsymbol{c}$），也就是说，第 i个元素与其他所有元素都不同，且除第 i个元素外的所有元素都相等。于是，通过一个（$r+k+\ell$）元函数 G，我们可以表示（$k+\ell$）个 r元函数 G_i，其中

$$G_i(\boldsymbol{a}) = b \iff G(\boldsymbol{a},\boldsymbol{c}) = b \quad \text{for all } \boldsymbol{c} \text{ satisfying } \chi_i(\boldsymbol{c}).$$

如果某些函数 G_i 的元数较小，我们也可以表示它们，只需忽略不需要的参数即可。此外，我们还可以通过关系的特征函数来表示关系。

设 $F \notin \Upsilon$ 是一个元数为 $(k+\ell+r)$ 的新函数符号。我们将按照刚才所述的方式使用 F 表示关系 R_i 和函数 F_i。对于每个 $i \in [k]$，令 ψ'_i 为从 ψ_i 得到的公式：将每个子公式 $R_p(x'_1,\dots,x'_{r_p})$ 替换为公式

$$\rho_p(x'_1,\dots,x'_{r_p}) := \exists x_{r_p+1}\dots\exists x'_r\exists \boldsymbol{z}\big(\chi_p(\boldsymbol{z}) \wedge F(x'_1,\dots,x'_r,\boldsymbol{z}) = 1\big)$$

并将每个子项 $F_q(x'_1, \ldots, x'_{s_q})$ 替换为项

$$\tau_q(x'_1, \ldots, x'_{s_q}) \coloneqq \operatorname*{avg}_{(x'_{s_q+1}, \ldots, x'_r, \boldsymbol{z}):\chi_q(\boldsymbol{z})} F(x'_1, \ldots, x'_r, \boldsymbol{z}).$$

同样，对于每个 $j \in [\ell]$，我们令 η'_j 为由 η_j 得到的项：将每个子公式 $R_p(x'_1, \ldots, x'_{r_p})$ 替换为公式 $\rho_p(x'_1, \ldots, x'_{r_p})$，并将每个子项 $F_q(x'_1, \ldots, x'_{s_q})$ 替换为项 $\tau_q(x'_1, \ldots, x'_{s_q})$。

现在，对于每个 $i \in [k+\ell]$，我们如下定义项 $\theta_i(\boldsymbol{x}, \boldsymbol{z})$：

- 我们令

$$\theta_{k+\ell}(\boldsymbol{x}, \boldsymbol{z}) \coloneqq \mathsf{if}\ \chi_{k+\ell}(\boldsymbol{z})\ \mathsf{then}\ \eta'_{k+\ell}\ \mathsf{else}\ \bot;$$

对于 $j \in [\ell - 1]$，- 我们令

$$\theta_{k+j}(\boldsymbol{x}, \boldsymbol{z}) \coloneqq \mathsf{if}\ \chi_{k+j}(\boldsymbol{z})\ \mathsf{then}\ \eta'_{k+j}\ \mathsf{else}\ \theta_{k+j+1};$$

对于 $i \in [k]$，- 我们令

$$\theta_i(\boldsymbol{x}, \boldsymbol{z}) \coloneqq \mathsf{if}\ \chi_i(\boldsymbol{z}) \wedge \varphi'_i\ \mathsf{then}\ 1\ \mathsf{else}\ \theta_{i+1}.$$

令 $\theta \coloneqq \theta_1$，并考虑 IFP(SUM) 项

$$\zeta(\boldsymbol{x}, \boldsymbol{z}) \coloneqq \mathsf{ifp}\big(F(\boldsymbol{x}, \boldsymbol{z}) \leftarrow \theta\big)(\boldsymbol{x}, \boldsymbol{z}).$$

设 $\mathcal{A}$是一个结构（至少含有 2 个元素）。由于项 θ 的所有自由变量都出现在 $\boldsymbol{x}$ 或 $\boldsymbol{z}$中，因此无需赋值即可定义函数 $F^{(n)}$： $A^{k+\ell+r} \to \mathbb{R}_\bot$，其中$n \in \mathbb{N}$。由归纳法 n，很容易证明，对于所有 $i \in [k]$ 和 $\boldsymbol{a}^=$（a_1，……，a_r），若 $\in A^r$且 $\boldsymbol{c} \in A^{k+\ell}$ 满足 $\mathcal{A} \models \chi_i(\boldsymbol{c})$，则有

$$F^{(n)}(\boldsymbol{a}, \boldsymbol{c}) = 1 \iff (a_1, \ldots, a_{r_i}) \in R_i^{T_\Sigma^n(\mathcal{A})}$$

并且，对于所有 $j \in [\ell]$ 和 $\boldsymbol{a} =$（a_1，……，a_r），若 $\in A^r$ 且 $\boldsymbol{c} \in A^{k+\ell}$满足 $\mathcal{A} \models \chi_{k+j}(\boldsymbol{c})$，则有

$$F^{(n)}(\boldsymbol{a}, \boldsymbol{c}) = F_j^{T_\Sigma^n(\mathcal{A})}(a_1, \ldots, a_{s_j}).$$

此外，对于所有 $\boldsymbol{a} \in A^r$ 以及所有满足 $\mathcal{A} \not\models \chi_i(\boldsymbol{c})$的 $\boldsymbol{c} \in A^{k+\ell}$，对任意 $i \in [k+\ell]$，都有 $F^{(n)}(\boldsymbol{a}, \boldsymbol{c}) = \bot$。

为了完成证明，我们需要根据 Σ的答案符号 S 进行分类讨论。若对某个 $i \in [k]$有 $S = R_i$，则令

$$\xi(x_1, \ldots, x_{r_i}) \coloneqq \exists x_{r_i+1} \ldots \exists x_r \exists \boldsymbol{z}\Big(\chi_i(\boldsymbol{z}) \wedge \zeta(x_1, \ldots, x_r, \boldsymbol{z}) = 1\Big).$$

如果对于某个 $j \in [\ell]$，$S = F_j$，我们令

$$\xi(x_1, \ldots, x_{s_j}) \coloneqq \operatorname*{avg}_{(x_{s_j+1}, \ldots, x_r, \boldsymbol{z}):\chi_j(\boldsymbol{z})} \zeta(x_1, \ldots, x_r, \boldsymbol{z}).$$

A.3. 定理 5.3的证明

作为起点，我们指出，很容易证明 FO(SUM) 具有多项式时间数据复杂度。

引理 A.5. 存在一种算法，给定一个FO(SUM)项 $\xi(\boldsymbol{x})$、一个有理数结构 $\mathcal{A}$ 和一个元组 $\boldsymbol{a} \in A^{|\boldsymbol{x}|}$，可在关于 $\|\mathcal{A}\|$ 的多项式时间内计算 $[\![\xi]\!]^{\mathcal{A}}(\boldsymbol{a})$。

引理的证明。所有对有理数进行的运算，都可以在关于输入数字位数的多项式时间内完成。因此，我们可以在多项式时间内计算项。然后，我们可以用通常的方法计算一阶公式。 □

现在证明定理 5.3。考察引理 4.9 的证明可知，将表达式转换为范式的过程保持标量性。因此，只需考虑范式中的表达式，这基本上意味着我们必须计算项

$$\eta = \mathsf{ifp}\left(F(\boldsymbol{x}) \leftarrow \theta\right)(\boldsymbol{x}'), \tag{A.A}$$

其中 $\theta(\boldsymbol{x})$ 是一个次数为 ≤ 1 的 FO(SUM) 项。令 $\Upsilon := \mathsf{ext}(\eta) = \mathsf{ext}(\theta) \setminus \{F\}$，且 $k := \mathsf{ar}(F)$。

函数 f 的公分母 是指满足以下条件的 $q \in \mathbb{N}_{>0}$：$D \to \mathbb{Q}$，且对所有 d，$f(d) \cdot q \in \mathbb{N}$。注意，如果 D 是有限的，那么 f 有唯一的最小公分母。如果$D = \emptyset$，默认令 1 为 f 的最小公分母。对于 $f : D \to \mathbb{Q}_\perp$，我们将（最小）公分母 定义为 f 在满足 $f(d) \neq \perp$ 的所有 $d \in D$ 构成的集合上的限制的（最小）公分母。

Claim 1. 设 $\mathcal{A}$ 是一个有理数 Υ-结构，且 $\zeta(\boldsymbol{y})$ 是词汇表为 $\Upsilon \cup \{F\}$ 的一个 FO(SUM) 项，其中 $|\boldsymbol{y}| = \ell$，并且满足$\deg_{\{F\}}(\zeta) \leq 1$。

那么存在 c、$d \in \mathbb{N}$，使得 $\|c\|$、$\|d\| \in \|\mathcal{A}\|^{O(1)}$，并且对每个函数 $F : A^k \to \mathbb{Q}_\perp$，以下结论成立。令 q 为 $\overline{F}$的公分母，并令 $p \in \mathbb{N}_{>0}$ 满足：对于所有满足 $F(\boldsymbol{a}) \neq \perp$ 的 $\boldsymbol{a} \in A^k$，都有 $p \geq |F(\boldsymbol{a})| \cdot q$。

那么存在一个 $j \in [d]$，使得 $j \cdot q$ 是 $[\![\zeta]\!]^{(\mathcal{A},F)}$ 的公分母（将其视为从 $A^{\overline{\ell}}$ 到 $\mathbb{Q}_\perp$ 的函数），并且对于所有满足 $[\![\zeta]\!]^{(\mathcal{A},F)}(\boldsymbol{b}) \neq \perp$ 的 $\boldsymbol{b} \in A^\ell$，都有 $\left|[\![\zeta]\!]^{(\mathcal{A},F)}(\boldsymbol{b})\right| \cdot j \cdot q \leq c \cdot p$。

Proof of Claim 1: 我们对 ζ 使用归纳法证明该结论。基例很直接：

- 若 $\zeta = r$ 为某个常数 $r = \frac{p'}{q'} \in \mathbb{Q}$，则令 $c := p'$ 和 $d := q'$；若 $\zeta = \perp$，则令$c := 0$ 和 $d := 1$。；
- 若 $\zeta = F(\boldsymbol{y})$，则令 $c := d := 1$；
- 若对某个权重函数符号 $G \in \Upsilon$，有 $\zeta = G(\boldsymbol{y})$，则令 d 为 $G^{\mathcal{A}}$的最小公倍数，并令 $c := d \cdot \max\{|G^{\mathcal{A}}(\boldsymbol{b})| : \boldsymbol{b} \in A_{\mathsf{ar}(G)}\}$。

接下来假设 $\zeta = \zeta_1 \circ \zeta_2$，并令 c_i，d_i 这些常数对应于 ζ_i，它们是我们从归纳假设中得到的。

• 如果 $\circ \in \{+,-\}$，我们令 $d := d_1 d_2$ 和 $c := c_1 d_2 + c_2 d_1$。

• 若 $\circ = \cdot$，则至多有一个 ζ_i 包含 F，因为 $\deg_{\{F\}}(\zeta) \leq 1$。设 ζ_2 不包含 F。那么不失一般性，我们可以假设 d_2 是 $[\![\zeta_2]\!]^{\mathcal{A}}$ 和 c_2 的公分母：$= d_2 \cdot \max\left\{ \left| [\![\zeta_2]\!]^{\mathcal{A}}(\boldsymbol{a}) \right| \, \middle| \, \boldsymbol{a} \in A^k, \text{ 且 } [\![\zeta_2]\!]^{\mathcal{A}}(\boldsymbol{a}) \neq \bot \right\}$。令 $c := c_1 c_2$，并令 $d := d_1 d_2$。

• 若 $\circ = /$，则 ζ_2 不包含 F，同样可以假设 d_2 是 $[\![\zeta_2]\!]^{\mathcal{A}}$ 和 c_2 的公分母：$= d_2 \cdot \max\left\{ \left| [\![\zeta_2]\!]^{\mathcal{A}}(\boldsymbol{a}) \right| : \boldsymbol{a} \in A^k, \text{ 且 } [\![\zeta_2]\!]^{\mathcal{A}}(\boldsymbol{a}) \neq \bot \right\}$。令 $c := c_1 d_2$，并令 $d := d_1 c_2$。

接下来，假设 $\zeta = \text{if } \varphi \text{ then } \zeta_1 \text{ else } \zeta_2$，并令 c_i, d_i 为我们根据归纳假设得到的 ζ_i 的常数。令 $d := d_1 d_2$，并令 $c := \max\{c_1 d_2, c_2 d_1\}$。

最后，假设 $\zeta(\boldsymbol{y}) = \sum_{\boldsymbol{z}:\varphi(\boldsymbol{y},\boldsymbol{z})} \zeta'(\boldsymbol{y},\boldsymbol{z})$，其中 $|\boldsymbol{z}| = m$。并令 c', d' 为我们根据归纳假设得到的 ζ' 的常数。令 $j \leq d'$，使得 $j \cdot q$ 是 $[\![\zeta']\!]^{(\mathcal{A},F)}$ 的公分母，将其视为从 $A^{\ell+m}$ 到 $\mathbb{Q}_\bot$ 的函数。于是 $j \cdot q$ 也是 $[\![\zeta]\!]^{(\mathcal{A},F)}$ 的公分母，将其视为从 A^ℓ 到 $\mathbb{Q}_\bot$ 的函数。因此，我们可以令 $d := d'$。

对于 $\boldsymbol{b} \in A^\ell$，令 $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n$ 为所有满足 $(\mathcal{A}, F) \models \varphi(\boldsymbol{b}, \boldsymbol{c})$ 的 $\overline{\boldsymbol{c}} \in A^m$ 的列表。根据归纳假设，对于所有 $i \in [n]$，我们都有 $|[\![\zeta']\!]^{(\mathcal{A},\overline{F})}(\boldsymbol{b}, \boldsymbol{c}_i)| \cdot j \cdot q \leq c' \cdot p$。因此

$$|[\![\zeta]\!]^{(\mathcal{A},\overline{F})}(\boldsymbol{b})| \cdot j \cdot q \leq \sum_{i=1}^{n} \left| [\![\zeta']\!]^{(\mathcal{A},\overline{F})}(\boldsymbol{b}, \boldsymbol{c}_i) \right| \cdot j \cdot q \leq n \cdot c' \cdot p \leq |A|^m \cdot c' \cdot p.$$

令 $c := |A|^m \cdot c'$。

命题的证明至此完成。

设 $\mathcal{A}$ 是一个有理数 Υ-结构。为计算 (A.A) 中的项 η，我们计算函数序列 $F^{(t)} : A^k \to \mathbb{Q}_\bot$，其中 $t \in \{0, \ldots, |A|^k\}$。回顾 $F^{(0)}(\boldsymbol{a}) = \bot$ 对所有 $\boldsymbol{a}$ 都成立，并且 $F^{(t+1)} = [\![\theta]\!]^{(\mathcal{A},F^{(t)})}$。根据将命题 1 应用于 $\theta(\boldsymbol{x})$ 的结果，选择 c、d。于是，对所有 $t \in \mathbb{N}$，如果 q 是 $F^{(t)}$ 的公分母，那么对于某个 $j \in [d]$，jq 是 $F^{(t+1)}$ 的公分母。此外，如果 $p \in \mathbb{N}_{>0}$ 使得对于所有满足 $F^{(t)}(\boldsymbol{a}) \neq \bot$ 的 $\boldsymbol{a} \in A^k$，都有 $p \geq \left|F^{(t)}(\boldsymbol{a})\right| \cdot q$，那么对于所有满足 $F^{(t+1)}(\boldsymbol{a}) \neq \bot$ 的 $\boldsymbol{a} \in A^k$，都有 $c \cdot p \geq \left|F^{(t+1)}(\boldsymbol{a})\right| \cdot j \cdot q$。

注意到，对于所有满足 $F^{(0)}(\boldsymbol{a}) \neq \bot$ 的 $\boldsymbol{a} \in A^k$，1 都是 $F^{(0)}$ 和 $1 \geq |F^{(0)}(\boldsymbol{a})|$ 的公分母。通过简单的归纳法可知，对于每个 $t \geq 1$，都存在一个 $q_t \leq d^t$，使得 q_t 是 $F^{(t)}$ 的公分母，并且对于所有满足 $F^{(t)}(\boldsymbol{a}) \neq \bot$ 的 $\boldsymbol{a} \in A^k$，$c^t \geq |F^{(t)}(\boldsymbol{a})| \cdot q_t$。

由于 $\|c\|, \|d\| = \|A\|^{O(1)}$，因此对于 $t \leq |A|^k$，有 $\sum_{\boldsymbol{a} \in A^k} \|F^{(t)}(\boldsymbol{a})\| = \|A\|^{O(1)}$。因此，我们可以利用引理 A.5，在多项式时间内计算 $F^{(t+1)} = [\![\theta]\!]^{(\mathcal{A},F^{(t)})}$。

A.4. 定理的证明 5.5

令 $\mathcal{A}$，$\mathcal{B}$ 为 Υ-结构。从 $\mathcal{A}$ 到 $\mathcal{B}$ 的一个同构是一个双射 $\pi : A \to B$，满足以下两个条件。

（i）对于所有 k元关系符号 $R \in \Upsilon$ 以及所有元组 $\boldsymbol{a} \in A^k$，都有 $\boldsymbol{a} \in R^{\mathcal{A}} \iff \pi(\boldsymbol{a}) \in R^{\mathcal{B}}$。

(ii) 对于所有 k 元权重函数符号 $F \in \Upsilon$ 以及所有元组 $\boldsymbol{a} \in A^k$，都有 $F^{\mathcal{A}}(\boldsymbol{a}) = F^{\mathcal{B}}(\pi(\boldsymbol{a}))$。

从 $\mathcal{A}$ 到 $\mathcal{B}$ 的局部同构 是一个双射 λ，它将集合 dom(λ) $\subseteq A$ 映射到集合 rg(λ) $\subseteq B$，并且对所有元组 $\boldsymbol{a} \in$dom(λ)^k 满足条件 (i) 和 (ii)。将局部同构描述为一组 $\lambda \subseteq A \times B$ 有序对通常会比较方便。

设 $k, \ell \in \mathbb{N}$，使得 $\ell \leq k$ 且 $\boldsymbol{a} = (a_1, \ldots, a_\ell) \in A^\ell, \boldsymbol{b} = (b_1, \ldots, b_\ell) \in B^\ell$。在 $\mathcal{A}$, $\mathcal{B}$ 上进行的带初始位置 $\boldsymbol{a}, \boldsymbol{b}$ 的双射 k-卵石博弈由两名玩家进行，分别称为 Spoiler 和 Duplicator。博弈的一个位置是大小为 $|p| \leq k$ 的集合 $p \subseteq A \times B$；初始位置为 $p_0 := \{(a_i, b_i) \mid i \in [\ell]\}$。如果 $|A| \neq |B|$，或者如果 p_0 不是局部同构，博弈立即结束，Spoiler 获胜。否则，博弈将按可能无限的回合序列进行。每一回合由以下步骤 (a)–(c) 组成。假设回合开始前的位置为 p。

(a) Spoiler 选择一个大小为 $|p'| < k$ 的子集 $p' \subseteq p$。

(b) Duplicator 选择一个双射 $\beta \colon A \to B$。

(c) Spoiler 选择一个 $a \in A$，新局面为 $p \cup \{(a, \beta(a))\}$。

如果在博弈过程中到达的某个局面 p 不是局部同构，则博弈结束，Spoiler 获胜。否则，博弈继续。如果博弈永不结束，也就是说，每个局面都是局部同构，则 Duplicator 获胜。

我们将此博弈记为 BP$_k(\mathcal{A}, \boldsymbol{a}, \mathcal{B}, \boldsymbol{b})$；如果 $\ell = 0$，则简记为 BP$_k(\mathcal{A}, \mathcal{B})$。

注意，不失一般性，我们可以假设在每一轮的步骤 (a) 中，如果当前局面 p 的大小为 $|p| < k$，那么 Spoiler 选择 $p' = p$；而如果 $|p| = k$，则 Spoiler 选择一个大小为 $|p'| = k - 1$ 的 $p' \subset p$。

引理 A.6. 设 $k, \ell \in \mathbb{N}$ 满足 $\ell \leq k$。此外，设 $\mathcal{A}, \mathcal{B}$ 为 Υ-结构，且$\boldsymbol{a} = (a_1, \ldots, a_\ell) \in A^\ell$, $\boldsymbol{b} = (b_1, \ldots, b_\ell) \in B^\ell$，使得 *Duplicator* 在博弈 $\mathrm{BP}_k(\mathcal{A}, \boldsymbol{a}, \mathcal{B}, \boldsymbol{b})$ 中有获胜策略。那么，对于所有至多含有k 个变量的 FO(SUM) 公式 $\varphi(x_1, \ldots, x_\ell)$，都有

$$\mathcal{A} \models \varphi(a_1, \ldots, a_\ell) \iff \mathcal{B} \models \varphi(b_1, \ldots, b_\ell), \tag{A.B}$$

并且，对于所有至多含有 k 个变量的 FO(SUM) 权重项 $\theta(x_1, \ldots, x_\ell)$，都有

$$\theta^{\mathcal{A}}(a_1, \ldots, a_\ell) = \theta^{\mathcal{B}}(b_1, \ldots, b_\ell). \tag{A.C}$$

证明。 采用同时归纳法，对 φ 和 θ进行证明。不等式、布尔联结词、算术运算符以及 if-then-else 的基例和归纳步骤都很直接。唯一有趣的情况是量化和求和。

我们先考虑量化。假设 $\varphi(x_1、\ldots\ldots、x_\ell) = \exists x \psi(x_1、\ldots\ldots、x_\ell, x)$，并且$\mathcal{A} \models \varphi(a_1、\ldots\ldots、a_k)$。令 $a \in A$ 满足 $\mathcal{A} \models \psi(a_1、\ldots\ldots、a_k, a)$。考虑第一轮中的

博弈 $\mathrm{BP}_k(\mathcal{A}, \boldsymbol{a}, \mathcal{B}, \boldsymbol{b})$。初始位置为 $p_0 = \{(a_i, b_i) \mid i \in [\ell]\}$。假设在步骤 (a) 中，Spoiler 选择位置 $p' := p_0$。这是可能的，因为 $|p_0| \le \ell < k$。令 β 为 Duplicator 根据其获胜策略在 (b) 中选择的双射。假设在步骤 (c) 中，Spoiler 选择 a，并令 $b := \beta(a)$。于是新位置为 $p_0 \cup \{(a, b)\}$，且 Duplicator 赢得博弈 $\mathrm{BP}_k(\mathcal{A}, \boldsymbol{a}a, \mathcal{B}, \boldsymbol{b}b)$。根据归纳假设，$\mathcal{A} \models \psi(a_1, \ldots, a_k, a) \iff \mathcal{B} \models \psi(b_1, \ldots, b_k, b)$，因此由 a 的选择可得 $\mathcal{B} \models \psi(b_1, \ldots, b_k, b)$。于是 $\mathcal{B} \models \varphi(b_1, \ldots, b_k)$。

同样，如果 $\mathcal{B} \models \varphi(b_1, \ldots, b_k)$，那么 $\mathcal{A} \models \varphi(a_1, \ldots, a_k)$。这就证明了（A.B），对于 $\varphi(x_1, \ldots, x_\ell) = \exists x \psi(x_1, \ldots, x_\ell, x)$。

公式 $\varphi(x_1, \ldots, x_\ell) = \forall x \psi(x_1, \ldots, x_\ell, x)$ 也可以用类似的方法处理。

最有趣的情形是求和项。考虑这样一个项

$$\theta(x_1, \ldots, x_\ell) = \sum_{(x_{\ell+1}, \ldots, x_m) : \varphi(x_1, \ldots, x_m)} \eta(x_1, \ldots, x_m),$$

对于某个 $m > \ell$、公式 $\varphi(x_1, \ldots, x_m)$ 和项 $\eta(x_1, \ldots, x_m)$。

我们考虑博弈 $\mathrm{BP}_k(\mathcal{A}, \boldsymbol{a}, \mathcal{B}, \boldsymbol{b})$ 的前 $m - \ell$ 回合，其中 Duplicator 按照其获胜策略进行游戏。令 $p_0 = \{(a_i, b_i) \mid i \in [\ell]\}$ 为初始位置。由于 $m \le k$，我们可以假设，在前 $(m - \ell)$ 回合的每一回合的步骤 (a) 中，Spoiler 只选择当前大小为 $< k$ 的位置。对于每个元组 $\boldsymbol{a}' = a'_{\ell+1} \ldots a'_m \in A^{m-\ell}$，$\beta_1^{\boldsymbol{a}'} \ldots \beta_m^{\boldsymbol{a}'}$，$\boldsymbol{b}^{\boldsymbol{a}'} = b_{\ell+1}^{\boldsymbol{a}'} \ldots b_m^{\boldsymbol{a}'} \in B^{m-\ell}$，, ,，我们归纳地定义一个序列 , , 以及一个元组 (, ,)，如下：$\beta_1^{\boldsymbol{a}'}$ 是 Duplicator 在博弈第一回合步骤 (b) 中选出的双射，且 $b_1^{\boldsymbol{a}'} := \beta_1^{\boldsymbol{a}'}(a_{\ell+1})$。假设 Spoiler 在步骤 (c) 中选择了 $a_{\ell+1}$，那么新位置 $p_1 := p_0 \cup \{(a_{\ell+1}, b_1^{\boldsymbol{a}'})\}$ 是 Duplicator 的一个获胜位置。对于归纳步骤，考虑某个 $i < m - \ell$，并假设位置 $p_i = p_0 \cup \{(a_{\ell+1}, b_1^{\boldsymbol{a}'}) \ldots, (a_{\ell+i}, b_i^{\boldsymbol{a}'})\}$ 是 Duplicator 的一个获胜位置。令 $\beta_{i+1}^{\boldsymbol{a}'}$ 为 Duplicator 在博弈第 $(i+1)$ 回合步骤 (b) 中选出的双射，且 $b_{i+1}^{\boldsymbol{a}'} := \beta_{i+1}^{\boldsymbol{a}'}(a_{\ell+i+1})$。假设 Spoiler 在步骤 (c) 中选择了 $a_{\ell+i+1}$，那么新位置 $p_{i+1} := p_{i+1} = p_0 \cup \{(a_{\ell+1}, b_1^{\boldsymbol{a}'}) \ldots, (a_{\ell+i+1}, b_{i+1}^{\boldsymbol{a}'})\}$ 仍是 Duplicator 的一个获胜位置。

因此，根据归纳假设，我们有

$$\mathcal{A} \models \varphi(a_1, \ldots, a_m) \iff \mathcal{B} \models \varphi(b_1, \ldots, b_\ell, b_1^{\boldsymbol{a}'}, \ldots, b_{m-\ell}^{\boldsymbol{a}'}), \tag{A.D}$$

$$\eta^{\mathcal{A}}(a_1, \ldots, a_m) = \eta^{\mathcal{B}}(b_1, \ldots, b_\ell, b_1^{\boldsymbol{a}'}, \ldots, b_{m-\ell}^{\boldsymbol{a}'}). \tag{A.E}$$

令 $\beta\colon\ A^{m-\ell} \to B^{m-\ell}$ 为由 $\beta(\boldsymbol{a}')$ 定义的映射：$= (b_1^{\boldsymbol{a}'}, \ldots, b_{m-\ell}^{\boldsymbol{a}'})$。我们将证明 β 是双射。一旦证明了这一点，(A.D) 和 (A.E) 就意味着 $\theta^{\mathcal{A}}(\boldsymbol{a}) = \theta^{\mathcal{B}}(\boldsymbol{b})$。

为证明 β 是单射，考虑不同的元组 $\boldsymbol{a}' = (a'_{\ell+1}, \ldots, a'_m)$ $\boldsymbol{a}'' = (a''_{\ell+1}, \ldots, a''_m) \in A^{m-\ell}$。令 $i \in \{0, \ldots, m - \ell - 1\}$ 满足：对于所有 $j \le i$ 和 $a'_{\ell+i+1} \neq a''_{\ell+i+1}$，都有 $a'_{\ell+j} = a''_{\ell+j}$。由于 $b_1^{\boldsymbol{a}'}, \ldots, b_i^{\boldsymbol{a}'}$ 和 $\beta_{i+1}^{\boldsymbol{a}'}$ 仅取决于 $a'_{\ell+1}, \ldots, a'_{\ell+i}$，因此有 $\beta_{i+1}^{\boldsymbol{a}'} = \beta_{i+1}^{\boldsymbol{a}''}$。由于 $\beta_{i+1}^{\boldsymbol{a}'}\colon\ A \to B$ 是双射，且 $a'_{\ell+i+1} \neq a''_{\ell+i+1}$，因此有

$$b_{i+1}^{\boldsymbol{a}'} = \beta_{i+1}^{\boldsymbol{a}'}(a'_{\ell+i+1}) \neq \beta_{i+1}^{\boldsymbol{a}'}(a''_{\ell+i+1}) = \beta_{i+1}^{\boldsymbol{a}''}(a''_{\ell+i+1}) = b_{i+1}^{\boldsymbol{a}''}.$$

因此 $\beta(\boldsymbol{a}') \neq \beta(\boldsymbol{a}'')$，这证明了 β 是单射。由于 β 的论域 $A^{m-\ell}$ 和陪域 $B^{m-\ell}$ 是大小相同的有限集，因此可得 β 是双射。 □

为证明定理 5.5，我们应用以下众所周知的结果。

定理 A.7（Cai、F¨urer 和 Immerman [16]）。对于每个 $k \in \mathbb{N}$，都存在图 $\mathcal{A}_k$，$\mathcal{B}_k$ 使得 $\mathcal{A}_k \not\cong B_k$ 并且 *Duplicator* 在博弈$\mathrm{BP}_k(\mathcal{A}_k, \mathcal{B}_k)$中拥有获胜策略。

此外，$\mathcal{A}_k$ 和 $\mathcal{B}_k$ 可在多项式时间内区分。也就是说，存在一个多项式时间算法，接受所有 $\mathcal{A}_k$ 并拒绝所有 $\mathcal{B}_k$。

定理 5.5 的证明。设 Q 是布尔查询：当多项式时间算法定理 A.7 接受 $\mathcal{A}$ 时，它在结构 $\mathcal{A}$ 上为真，否则为假。反设存在一个 IFP(SUM) 句子 φ 表示 Q。根据引理 4.9，我们可以假设 $\varphi = \exists \boldsymbol{x}\Big(\chi(\boldsymbol{x}) \wedge \mathsf{ifp}\ \big(F(\boldsymbol{x}) \leftarrow \theta\big)(\boldsymbol{x})\Big)$，其中 χ是某个选择条件，$\theta(\boldsymbol{x})$是某个 FO(SUM) 项。令 $\ell := |\boldsymbol{x}|$，并设 m 为 θ中出现的变量（自由或约束）的数量。

令 $k := 2\ell + m + 1$，并考虑定理 $\mathcal{A} := \mathcal{A}_k$和 $\mathcal{B} := \mathcal{B}_k$中的结构 A.7。于是

$$\mathcal{A} \models \varphi \quad \text{and} \quad \mathcal{B} \not\models \varphi. \tag{A.F}$$

此外，Duplicator 在博弈 $\mathrm{BP}_k(\mathcal{A}_k, \mathcal{B}_k)$ 中拥有获胜策略，这意味着 $\mathcal{A}$ 和 $\mathcal{B}$ 具有相同的序，记为 n。此外，根据引理 A.6，$\mathcal{A}$ 和 $\mathcal{B}$ 满足相同的 FO(SUM) 句子，且变量数至多为 $k-1$。

考虑序列 $F^{(t)}{}_{\mathcal{A}}$和 $F^{(t)}_{\mathcal{B}}$。对于 $t \in \mathbb{N}$，我们计算在 $\mathcal{A}$和 $\mathcal{B}$中分别求值项 $\mathsf{ifp}\ \big(F(\boldsymbol{x}) \leftarrow \theta\big)(\boldsymbol{x})$ 时所得的结果。我们声称，对于每个 $t \geq 1$，都存在一个 FO(SUM) 项 $\theta^{(t)}(\boldsymbol{x})$，其变量数至多为 $\ell + m$，使得对所有 $\boldsymbol{a} \in A^{\ell}$，$F^{(t)}_{\mathcal{A}}(\boldsymbol{a}) = [\![\theta^{(t)}]\!]^{\mathcal{A}}(\boldsymbol{a})$，并且对所有 $\boldsymbol{b} \in B^{\ell}$，$F^{(t)}_{\mathcal{B}}(\boldsymbol{b}) = [\![\theta^{(t)}]\!]^{\mathcal{B}}(\boldsymbol{b})$。令 $\theta^{(1)}$ 为由 θ 得到的项：将 θ 中的每个子项 $F(\boldsymbol{y})$ 替换为常数 $\bot$。此外，对于每个$t \geq 1$，令 $\theta^{(t+1)}$ 为由 θ 得到的项：将每个子项 $F(\boldsymbol{y})$ 替换为

$$\zeta(\boldsymbol{y}) := \sum_{\boldsymbol{z}:\boldsymbol{z}=\boldsymbol{y}} \sum_{\boldsymbol{x}:\boldsymbol{x}=\boldsymbol{z}} \theta^{(t)}(\boldsymbol{x}).$$

此处 $\boldsymbol{z}$ 是一个与 $\boldsymbol{y}$ 和 $\boldsymbol{x}$均不相交的 ℓ元变量元组，而 $\boldsymbol{z} = \boldsymbol{y}$ 是$\bigwedge_{i=1}^{\ell} z_i = y_i$的缩写。两个求和算子的作用仅仅是将正确的变量放入项 $\theta^{(t)}$中；对于所有结构 $\mathcal{C}$ 和元组 $\boldsymbol{c} \in C^{\ell}$，均有 $[\![\zeta(\boldsymbol{y})]\!]^{\mathcal{C}}(\boldsymbol{c}) = [\![\theta^{(t)}(\boldsymbol{x})]\!]^{\mathcal{C}}(\boldsymbol{c})$。因此，在语义上，$\theta^{(t+1)}$ 是通过将 F 替换为定义 $F^{(t)}$的项而从 θ 得到的。此外，请注意，项 $\theta^{(t+1)}$ 除 θ 中的变量外，仅包含 $\boldsymbol{z}$ 和 $\boldsymbol{x}$ 中的变量，因此至多有 $2\ell + m$ 个变量。

由于不动点过程至多经过 $t := n^{\ell}$ 步即可收敛，因此对于所有 $\boldsymbol{a} \in A^{\ell}$，我们有 $[\![\ \mathsf{ifp}\ \big(F(\boldsymbol{x}) \leftarrow \theta\big)(\boldsymbol{x})]\!]^{\mathcal{A}}(\boldsymbol{a}) = [\![\theta^{(t)}]\!]^{\mathcal{A}}(\boldsymbol{a})$，类似地，对于所有 $\boldsymbol{b} \in B^{\ell}$，有 $[\![\ \mathsf{ifp}\ \big(F(\boldsymbol{x}) \leftarrow \theta\big)(\boldsymbol{x})]\!]^{\mathcal{B}}(\boldsymbol{b}) = [\![\theta^{(t)}]\!]^{\mathcal{B}}(\boldsymbol{b})$。因此

$$\mathcal{A} \models \varphi \iff \mathcal{A} \models \exists \boldsymbol{x}\big(\chi(\boldsymbol{x}) \wedge \theta^{(t)}(\boldsymbol{x})\big) \iff \mathcal{B} \models \exists \boldsymbol{x}\big(\chi(\boldsymbol{x}) \wedge \theta^{(t)}(\boldsymbol{x})\big) \iff \mathcal{B} \models \varphi,$$

因为公式 $\exists \boldsymbol{x}\big(\chi(\boldsymbol{x}) \wedge \theta^{(t)}(\boldsymbol{x})\big)$ 至多含有 $2\ell + m$ 个变量。这与（A.F）相矛盾。

□

A.5. 定理 5.6 的证明

我们将单变量 X多项式不等式的布尔组合简称为条件。描述是如下形式的表达式

若 γ_0 则 $\perp$否则若 γ_1
则 r_1否则若 γ_2 则 r_2

. . . 否
则 r_n

其中， γ_i 是条件， r_i 是单变量 X 的有理函数。我们称这种描述 δ 是良定义的，如果对于每个不满足 γ_0 的实数 a，所有 r_i 在 a 上都有定义（即不会发生除以零的情况）。对于任意 $a \in \mathbb{R}$，值 $\delta(a) \in \mathbb{R}_\perp$ 现在都以显而易见的方式定义。

令 $\mathbf{K}_1 \subseteq \mathbf{K}(1,1)$ 表示图 1(c) 中所示的前馈神经网络类，其中所有偏置均设为 0。每个网络 $\mathcal{N} \in \mathbf{K}_1$ 都有一个输入节点和一个输出节点，我们始终将它们分别记为 in 和 out。网络可以有任意数量的隐藏节点；我们用 $H(\mathcal{N})$ 表示这个数量。注意，任意两个隐藏节点都是对称的。

一个符号赋值 是从有限变量集 Y到 $\{\text{in},\text{out } h\}$ 的映射 σ，其中 h 是一个表示“隐藏”的符号。重要的是，在任何有限的 Y 上，符号赋值的数量都只有有限个。现在，如果某个 Y 上的实际赋值 ν 属于某个$\mathbf{K}_1$，且对于每个变量 $y \in Y$，都有 $\nu(y) = \text{in}$ 当且仅当$\sigma(y) = \text{in}$； $\nu(y) = \text{out}$ 当且仅当 $\sigma(y) = \text{out}$；并且 $\nu(y)$ 是隐藏节点当且仅当 $\sigma(y) = h$，则称其属于类型 σ。

通过精细但繁琐的归纳法，我们可以验证以下结论。

引理 A.8. 对于每个FO(SUM) 公式 φ，以及 φ的自由变量上的每个符号赋值 σ，都存在一个条件 γ，使得对于每个 $\mathcal{N} \in \mathbf{K}_1$ 以及 $\mathcal{N}$ 中每个属于类型 σ 的赋值 ν，都有 $\mathcal{N}, \nu \models \varphi$ 当且仅当 $\gamma(H(\mathcal{N}))$ 成立。

类似地，对于每个FO(SUM) 权重项 θ，以及 θ的自由变量上的每个符号估值 σ，都存在一个定义良好的描述 δ，使得对于每个 $\mathcal{N} \in \mathbf{K}_1$以及 $\mathcal{N}$ 中每个属于类型 σ 的赋值 ν，都有 $[\![\theta]\!]^{(\mathcal{N},\nu)} = \delta(H(\mathcal{N}))$。

现在考虑在 K(1, 1) 上的查询 Q，其中 $Q(\mathcal{N})$ 为真，当且仅当 $f^{\mathcal{N}}(1)$ 是偶自然数。对于任意 $\mathcal{N} \in \mathbf{K}_1$，，这意味着 $H(\mathcal{N})$ 是偶数。为导出矛盾，假设存在一个闭合 IFP(SUM) 公式 φ，使得对于每个 $\mathcal{N} \in \mathbf{K}_1$，$\mathcal{N} \models \varphi$ 当且仅当 $H(\mathcal{N})$ 是偶数。由于这类 $\mathcal{N}$ 中所有隐藏节点都是对称的，很容易看出，φ 中的所有不动点都能在常数次迭代后到达。因此，不失一般性，我们可以假设 φ 属于 FO(SUM)。于是，根据引理 A.8，并注意到 φ 没有自由变量，存在一个条件 γ，使得对于所有自然数 a，$\gamma(a)$ 成立当且仅当 a 是偶数。这是不可能的，因为 $\mathbb{R}$ 上多项式不等式的布尔组合只能定义有限个区间的并 [11]。

A.6. 定理 5.10 的证明

为简单起见，我们给出不带参数的布尔查询的证明。令 $\mathcal{N} \in \mathbf{K}(m,p)$。假设 $\mathcal{N}$的约化权重有某个多项式 $P(X)$界 P。

我们选择一个 $c \in \mathbb{N}$，使得对于所有 $n \geq 2$，$P(n) < n^c$。

作为证明的第一步，我们注意到，可以用一种规范的方式对 $\mathcal{N}$ 的节点进行线性排序。令 $i_1, \ldots, i_m$ 和 $o_1, \ldots, o_p$ 分别为 $\mathcal{N}$ 的输入节点和输出节点。我们令 $i_1 < \ldots < i_m < \tilde{o}_1 < \ldots < \tilde{o}_m$（回想一下，我们始终假设输入节点与输出节点互不相同）。对于隐藏节点 $\tilde{u}$，令 $i_j < \tilde{u} < \tilde{o}_k$。对于不同的隐藏节点 $\tilde{u}, \tilde{u}'$，若 u 的深度小于 u' 的深度，则令 $\tilde{u} < \tilde{u}'$。若 u 和 u' 的深度相同，但 $b(u) \neq b(u')$，则当且仅当 $b(u) < b(u')$ 时，令 $\tilde{u} < \tilde{u}'$。若 $b(u) = b(u')$，我们考虑所有深度更小的节点 $\tilde{v}_1 < \ldots < \tilde{v}_m$。此时存在某个 $i \in [m]$，使得 $w(\tilde{v}_i, u) \neq w(\tilde{v}_i, u')$，否则便会有 $u \sim u'$，从而得到 $\tilde{u} = \tilde{u}'$。我们选取满足 $w(\tilde{v}_i, \tilde{u}) \neq w(\tilde{v}_i, \tilde{u}')$ 的最小 i，并当且仅当 $w(\tilde{v}_i, \tilde{u}) < w(\tilde{v}_i, \tilde{u}')$ 时令 $\tilde{u} < \tilde{u}'$。

线性序 $\prec$ 上的导出一个准序 $\preceq$：我们令 $u \preceq v$，当且仅当 $\tilde{u} \leq \tilde{v}$。注意，$u \sim v$ 当且仅当 $u \preceq v$ 且 $v \preceq u$。

这个定义在 $\mathcal{N}$ 上的准序 $\preceq$ 可以由 sIFP(SUM) 定义。我们首先构造一个项 $\theta_{\mathrm{dep}}(x)$，使得 $[\![\theta_{\mathrm{dep}}]\!]^{\mathcal{N}}(u)$ 是 $\mathcal{N}$ 中 u 的深度。然后我们可以轻松构造一个 sIFP(SUM) 公式 $\varphi_{\preceq}(x, y)$，使得 $\mathcal{N} \models \varphi_{\preceq}(u, v) \iff u \preceq v$。此外，我们构造一个项 $\theta_w(x, y)$，使得 $[\![\theta_{\mathrm{wt}}]\!]^{\mathcal{N}}(u, v)$ 是 $\widetilde{\mathcal{N}}$ 中边 $(\tilde{u}, \tilde{v})$ 的权重。为统一记法，我们还令 $\theta_{\mathrm{bias}}(x) := b(x)$ 表示定义节点偏置的项。这样一来，我们就基本上在 $\mathcal{N}$ 中定义了 $\widetilde{\mathcal{N}}$。

令 $n := |\mathcal{N}|$。那么，n 是准序 $\preceq$ 的长度。注意，$n \geq 2$，因为 $\mathcal{N}$ 至少有一个输入节点和一个输出节点。回忆一下，我们选择了 c 使得 $P(n) < n^c$，其中多项式 $P(X)$ 对权重进行了界定。令 $\preceq_c$ 为与 $\preceq$ 关联的 c 元组上的字典序。也就是说，对于元组 $\boldsymbol{u} = (u_1, \ldots, u_c)$ 和 $\boldsymbol{v} = (v_1, \ldots, v_c)$，当且仅当 $\boldsymbol{u} \preceq_c \boldsymbol{v}$ 在以下情形之一成立时成立：对于所有 $i \in [c]$ 都有 $u_i \sim v_i$，或者存在满足 $u_i \not\sim v_i$ 的最小 i，使得 $u_i \prec v_i$。我们用数字 $i \in \{0, \ldots, n^c - 1\}$ 为准序 $\preceq_c$ 中的位置编号。对于每个这样的 i，令 $\mathsf{eqcl}(i)$ 表示关于 $\preceq_c$ 的第 i 个等价类；对于 $\mathcal{N}$ 的每个 c 元组 $\boldsymbol{u}$ 节点，令 $\mathsf{ind}(\boldsymbol{u})$ 为满足 $\boldsymbol{u} \in \mathsf{eqcl}(i)$ 的唯一 $i \in \{0, 1 \ldots, n^c - 1\}$。我们构造一个 sIFP(SUM) 公式 $\varphi_{\mathrm{lex}}(\boldsymbol{x}, \boldsymbol{y})$，该公式定义 $\preceq_c$，并构造一个项 $\theta_{\mathrm{ind}}(\boldsymbol{x})$，使得对于每个 c 元组 $\boldsymbol{u}$，都有 $[\![\theta_{\mathrm{ind}}]\!]^{\mathcal{N}}(\boldsymbol{u}) = \mathsf{ind}(\boldsymbol{u})$。此外，令 $\varphi_{\mathrm{slex}}(\boldsymbol{x}, \boldsymbol{y}) := \varphi_{\mathrm{lex}}(\boldsymbol{x}, \boldsymbol{y}) \wedge \neg\varphi_{\mathrm{lex}}(\boldsymbol{y}, \boldsymbol{x})$ 为定义严格字典序 $\prec_c$ 的公式，这也将很方便。

As $\widetilde{\mathcal{N}}$ 可写成分数界权重 $\frac{p}{q}$，其中 p 和 q 是偏置和权重都...假设对于 $\mathcal{N}$ 的节点 u，我们有既约形式的 $b(\tilde{u}) = b(u) = {}^{r(u)}\!/_{q(u)}$；对于 $\mathcal{N}$ 的每条边 (u, v)，我们有既约形式的 $w(\tilde{u}, \tilde{v}) = {}^{r(u,v)}\!/_{q(u,v)}$。接下来的一步可能是证明中的关键步骤。我们希望用 sIFP(SUM) 定义这些数 $r(u)$ $q(u)$ $r(u, v)$ $q(u, v)$。然而，如何直接通过项 $\theta(x)$ 或 $\theta(x, y)$ 来定义并不明显。我们通过定义准序 $\preceq_c$ 中这些数的指标来绕开这个问题。我们构造 sIFP(SUM) 公式 $\varphi_{\mathrm{bias}}(x, \boldsymbol{z}, \boldsymbol{z}')$ 和 $\varphi_{\mathrm{wt}}(x, y, \boldsymbol{z}, \boldsymbol{z}')$，使得对于 $\mathcal{N}$ 的所有节点 u、v 以及节点的 c-元组 $\boldsymbol{t}$、$\boldsymbol{t}'$，都有

$$\mathcal{N} \models \varphi_{\mathrm{bias}}(u, \boldsymbol{t}, \boldsymbol{t}') \iff \mathrm{ind}(\boldsymbol{t}) = |r(u)| \text{ and } \mathrm{ind}(\boldsymbol{t}') = q(u),$$
$$\mathcal{N} \models \varphi_{\mathrm{wt}}(u, v, \boldsymbol{t}, \boldsymbol{t}') \iff \mathrm{ind}(\boldsymbol{t}) = |r(u, v)| \text{ and } \mathrm{ind}(\boldsymbol{t}') = q(u, v).$$

为了构造 $\varphi_{\text{bias}}(x, \boldsymbol{z}, \boldsymbol{z}')$，我们先处理符号，令

$$\varphi_{\text{bias}}(x, \boldsymbol{z}, \boldsymbol{z}') \coloneqq \big(\theta_{\text{bias}}(x) < 0 \wedge \varphi_{\text{neg}}(x, \boldsymbol{z}, \boldsymbol{z}')\big) \vee \big(\theta_{\text{bias}}(x) \geq 0 \wedge \varphi_{\text{pos}}(x, \boldsymbol{z}, \boldsymbol{z}')\big).$$

现在令

$$\varphi_{\text{pos}}(x, \boldsymbol{z}, \boldsymbol{z}') \coloneqq \big(\theta_{\text{bias}}(x) \cdot \theta_{\text{ind}}(\boldsymbol{z}') = \theta_{\text{ind}}(\boldsymbol{z})\big) \wedge \forall \boldsymbol{y}' \Big(\varphi_{\text{slex}}(\boldsymbol{y}', \boldsymbol{z}') \rightarrow \neg \exists \boldsymbol{y}\, \theta_{\text{bias}}(x) \cdot \theta_{\text{ind}}(\boldsymbol{y}') = \theta_{\text{ind}}(\boldsymbol{y})\Big).$$

要定义 φ_{neg}，我们只需将 φ_{pos} 中两处 $\theta_{\text{bias}}(x)$ 都替换为 $(-1) \cdot \theta_{\text{bias}}(x)$。公式 $\varphi_{\text{wt}}(x, y, \boldsymbol{z}, \boldsymbol{z}')$ 也可以用类似方式定义。

~利用所有这些 sIFP(SUM) 表达式，我们可以在 $\mathcal{N}$中定义 $\mathcal{N}$ 的一个有序副本；形式上，这可以通过转导 [27]来实现。根据 Immerman-Vardi 定理 [32,47], , 有序结构上的每个多项式时间可计算查询都可以用 IFP表示，因此也可以用 sIFP(SUM)表示。由此，我们得到一个 IFP(SUM) 公式，它在 $\mathcal{N}$中定义了将查询 Q 应用于 ~所得的 $\mathcal{N}$答案。由于该查询是模型无关的，这也给出了将查询 Q 应用于 $\mathcal{N}$所得的答案。

A.7. 定理 6.2 的证明

设 $\varphi \coloneqq \bigwedge_{i=1}^{m}(\lambda_{i1} \wedge \lambda_{i2} \wedge \lambda_{i3})$，其中 $\lambda_{ij} \in \{X_k, \neg X_k\}$ 对某个 $k \in [n]$ 成立，这是一个关于布尔变量 $X_1, \ldots, X_n$ 的 3-CNF 公式。接下来，我们将构造一个前馈神经网络 $\mathcal{N}$，使得当且仅当 φ 不可满足时， $f^{\mathcal{N}}$ 是零函数。该构造基于这样一个想法：将二进制表示为 $(0.a_1a_2\ldots a_n)_2$ 的数视为变量赋值，并让 $\mathcal{N}$ 在这些数上模拟 φ。为此，我们首先使用一个具有以下性质的辅助前馈神经网络 $\mathcal{N}^{\text{split}} \in \mathbf{K}(1, n)$。对于所有 $i \in [n]$，都有

1. $\forall x\, f^{\mathcal{N}}_{\text{拆分输出}_i}(x) \in [0,1]$和 2. 如果 $x \in [0,1)$ 是 2^{-n}的倍数，且 $x = (0.a_1a_2\ldots a_n)_2$,，那么 $f^{\mathcal{N}}_{\text{拆分输出}_i}(x) = a_i$。

如何构造这样的网络 $\mathcal{N}^{\text{split}}$ 是众所周知的。为方便读者，我们在证明末尾给出该构造。接下来，我们旨在用前馈神经网络模拟 φ。为此，令 $f_\varphi : \mathbb{R}^n \to [0,1]$ 定义为

$$f_\varphi(\boldsymbol{x}) \coloneqq \min_{i \in [m]} \max_{j \in [3]} \ell_{ij}(\boldsymbol{x})$$

其中，对于 $\boldsymbol{x} = (x_1, \ldots, x_n)$，令 $\ell_{ij}(\boldsymbol{x}) = x_k$当 $\lambda_{ij} = X_k$ 时，以及 $\ell_{ij}(\boldsymbol{x}) = 1 - x_k$当 $\lambda_{ij} = \neg X_k$时。于是，对于所有 $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$，我们都有 $f_\varphi(\boldsymbol{x}) \in \{0,1\}$，其中 $f_\varphi(\boldsymbol{x}) = 1$当且仅当赋值 $X_k \mapsto x_k$ 满足 φ。注意，利用 $\max(x_i, x_j) = x_i + \text{ReLU}(x_j - x_i)$，可以轻松构造一个 FNN$\mathcal{N}_\varphi \in \mathbf{K}(n, 1)$，使其满足 $f^{\mathcal{N}_\varphi} = f_\varphi$。将计算 $f^{\mathcal{N}_\varphi} \circ f^{\mathcal{N}}$ split 的 FNN 作为 $\mathcal{N}$的候选似乎很有吸引力。事实上，如果 φ可满足，那么满足赋值 $X_i \mapsto a_i$会产生一个输入 $x = (0.a_1a_2\ldots a_n)_2$，使得 $f^{\mathcal{N}_\varphi}\big(f^{\mathcal{N}}\text{split}(x)\big) = 1$。然而，如果 φ 不可满足，我们希望 $f^{\mathcal{N}}$对所有 $x \in \mathbb{R}$均为 0，而不仅仅是对 2^{-n}的倍数为 0。幸运的是，我们可以利用以下洞见来确保这一要求。

Claim 2. 假设 φ 不可满足。那么，对于所有 $\boldsymbol{x} \in [0,1]^n$，都有 $0 \le f_\varphi(\boldsymbol{x}) \le 1/2$。

Proof of Claim 2: 设 $c \coloneqq \max_{\boldsymbol{x}\in[0,1]^n} f_\varphi(\boldsymbol{x})$。我们需要证明 $c \le 1/2$。假设 $c > 0$。选择 $\boldsymbol{x} = (x_1, \ldots, x_n) \in [0,1]^n$，使得 $f_\varphi(\boldsymbol{x}) = c$，且 $k \in [n]$ 的数量最少，并满足 $x_k \notin \{0,1\}$。那么存在一个 $i \in [m]$，使得 $X_k \in \{\lambda_{i1}, \lambda_{i2}, \lambda_{i3}\}$ 且 $x_k = \max_{j\in[3]} \ell_{ij}(\boldsymbol{x})$；否则，我们可以将 x_k 设为 0，而不降低 $f_\varphi(\boldsymbol{x})$。同理，存在一个 $i' \in [m]$，使得 $\neg X_k \in \{\lambda_{i'1}, \lambda_{i'2}, \lambda_{i'3}\}$且 $1 - x_k = \max_{j\in[3]} \ell_{i'j}(\boldsymbol{x})$；否则，我们可以将 x_k 设为 1，而不降低$f_\varphi(\boldsymbol{x})$。

由于 $x_k \le 1/2$ 或 $1 - x_k \le 1/2$，所以 $\max_{j\in[3]} \ell_{ij}(\boldsymbol{x}) \le 1/2$ 或 $\max_{j\in[3]} \ell_{i'j}(\boldsymbol{x}) \le 1/2$，因此 $c = f_\varphi(\boldsymbol{x}) \le 1/2$。这就证明了该结论。

由于 $\{f^{\mathcal{N}}_{\frac{1}{2}}(x) \mid x \in \mathbb{R}\} \subseteq [0,1]^n$ φ $\max_{x\in\mathbb{R}} f^{\mathcal{N}_\varphi}(f^{\mathcal{N}}(x)) \le$分割$(\,)$，如果不可满足，则得到 max分割$(\,)$。因此，我们令其由分割 与 的串接组成，并用权重为 2 的边将其连接到一个偏置为 -1 的神经元。然后 $\mathcal{N}$ 计算函数

$$f^{\mathcal{N}}(x) = \mathrm{ReLU}\Big(-1 + 2 \cdot f^{\mathcal{N}_\varphi}\big(f^{\mathcal{N}^{\mathsf{split}}}(x)\big)\Big)$$

对于不可满足的 φ，该函数恒为零；否则达到 1。

接下来给出 $\mathcal{N}^{\mathsf{split}}$ 的构造。为方便起见，我们将使用线性化*Sigmoid*函数（也称为 ReLU1），其定义为

$$\mathrm{lsig}(x) = \mathrm{ReLU}(x) - \mathrm{ReLU}(x-1) = \begin{cases} 1 & \text{if } x > 1, \\ x & \text{if } 0 \le x \le 1, \\ 0 & \text{if } x < 0. \end{cases}$$

我们首先旨在构造一个值域为 $[0,1]$ 且 $f^{\mathcal{N}_1}((0.a_1a_2\ldots a_n)_2) = a_1$ 的 $\mathcal{N}_1$ 前馈神经网络，属于 $\mathbf{K}(1,1)$。为此，我们注意到

$$2 \cdot (0.a_1a_2\ldots a_n)_2 - 1 \begin{cases} \ge 0 & \text{if } a_1 = 1, \\ \le -2^{-(n-1)} & \text{if } a_1 = 0. \end{cases}$$

现在，我们可以放大这一差距，并通过以下方式提取 a_1

$$a_1 = \mathrm{lsig}((2 \cdot (0.a_1a_2\ldots a_n)_2 - 1) \cdot 2^{n-1} + 1).$$

根据前一个等式，我们选择 $\mathcal{N}_1$ 来计算 $f^{\mathcal{N}_1}(x) = \mathrm{lsig}((2x-1)\cdot 2^{n-1}+1)$，从而获得所需性质。为了将其推广为构造 $\mathcal{N}^{\mathsf{split}}$，我们注意到 $(2 \cdot (0.a_1a_2\ldots a_n)_2 - a_1) = (0.a_2a_3\ldots a_n)$。因此，我们可以迭代地应用这一构造，并通过计算得到满足这些性质的前馈神经网络 $\mathcal{N}^{\mathsf{split}}$

$$f^{\mathcal{N}^{\mathsf{split}}}_{\mathrm{out}_i}(x) = \mathrm{lsig}\Big(\big(2^i x - 1 - \sum_{j=1}^{i-1} 2^{i-j} f^{\mathcal{N}^{\mathsf{split}}}_{\mathrm{out}_j}(x)\big) \cdot 2^{n-i} + 1\Big).$$