arXiv:2304.00931v1 [cs.DB] 3 Apr 2023

数据图修复：优选方法

Abriola Sergio[1,2], Cifuentes Santiago[1,2], Pardal Nina[2,3], 和 Pin Edwin[2]

[1] 计算系´算机´，精确科学学院，布宜诺斯艾利斯大学，阿根廷[2] ICC CONICET，布宜诺斯艾利斯，阿根廷[3] 计算机科学系，谢菲尔德大学，英国

摘要。一致性知识库的修复问题已在知识表示与推理和数据库理论领域得到研究，并在数十年间取得了长足进展。随着信息变得更加复杂、相互关联，为了能够查询和推理这些信息，人们开发了新型存储库、表示语言和语义。图数据库为表示数据之间的关系提供了有效方式，并能高效地处理和查询这些连接。本文聚焦于为包含数据值的图数据库计算优选（子集和超集）修复的问题，并采用基于一组 Reg-GXPath 表达式（作为完整性约束）的一致性概念。具体而言，我们研究基于两种不同偏好准则计算优选修复的问题：一种基于权重，另一种基于多重集。结果表明，在大多数情况下，即使没有可供利用的偏好准则，也能够保持相同的计算复杂度。

关键词：数据图 · 修复 · 偏好

1 引言

图数据库在许多现代应用中都很有用。在这些应用中，数据的拓扑结构与数据本身同样重要，例如社交网络分析、 [13],数据溯源 [3], 和语义网 [5]。数据库的结构通常通过导航语言进行查询，例如正则路径查询（RPQ） [8] ，这类查询能够捕获由某种特定类型的路径连接的一对节点。这些查询语言可以扩展以增强表达能力，但通常也会增加求值的复杂度。例如，合取二元正则路径查询（C2RPQ）是正则路径查询的一种自然扩展，通过为语言添加反向遍历边的能力，并将表达式闭包扩展为合取（类似于关系合取查询）来定义。

正则路径查询（RPQ）及其最常见的扩展形式（合取二元正则路径查询（C2RPQ）和嵌套正则表达式（NRE） [7]）只能作用于图中的边，无法与节点中的数据值进行任何交互。因此，人们设计了用于数据图（即数据既存在于路径中，也存在于节点本身的图数据库）的查询语言，例如正则表达式映射（REM）和 Reg-GXPath [16]。

与关系型数据库的情况一样，人们通常期望数据能够保留与其所表示的现实世界相关的某种语义结构。在图数据库中，可以通过路径约束 [1,11]来表达这些完整性约束。

当数据库不满足其完整性约束时，一种可行的方法是寻找一个满足这些约束的“相似”数据库。在文献中，这个新数据库称为修复 [4], ；要对其作出恰当定义，就必须精确界定“相似”的含义。

文献中有多种修复概念，其中包括基于集合的修复 [20], 、基于属性的修复 [21], 和基于基数的修复 [17]。考虑在一组 Reg-GXPath 表达式 R 下对图数据库 G 进行基于集合的修复 G' 时，问题有两种自然的限制情形： G'是 G 的子图，以及 G' 是 G的超图。这类修复通常分别称为子集修复和超集修复 [20,6]。由于修复可能并不唯一，因此可以在修复集合上施加一种序，并根据该序寻找“最优”修复。关于关系数据库 [14,19] 及其他类型的基于逻辑的形式体系 [10,9] 中的优选修复，已有大量研究。然而，据我们所知，尚无研究聚焦于图数据库或数据图。在本文中，我们研究如何根据我们提出的两种偏好准则寻找优选修复。

本文结构如下。第 2 节介绍数据图模型的语法和语义所需的预备知识与符号表示，以及一致性和不同类型修复的定义。第 3 节提出两种为修复赋予偏好的方案。第一种基于权重赋值，第二种则将边和数据上的序提升为多重集序。对于这两种方案，我们都研究计算优选修复问题的计算复杂度。第 4 节讨论结论和未来的研究方向。

2 定义

固定一个有限的边标签集合 Σ_e ，以及一个可数（有限或无限可枚举）的数据值集合 Σ_n （有时称为数据标签），并假设这两个集合均非空，且 $\Sigma_e \cap \Sigma_n = \emptyset$。数据图 G 是一个三元组 (V, L_e, D)，其中 V 是节点集合， L_e 是从 $V \times V$ 到 $\mathcal{P}$（Σ_e）的映射，用于定义图的边； D 是从 V 到数据值集合 Σ_n 的映射。

Reg-GXPath 表达式通过以下相互递归给出：

$$\varphi, \psi := \mathrm{c}^{=} \mid \mathrm{c}^{\neq} \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \langle\alpha\rangle \mid \langle\alpha = \beta\rangle \mid \langle\alpha \neq \beta\rangle$$
$$\alpha, \beta := \epsilon \mid _ \mid \mathrm{a} \mid \mathrm{a}^{-} \mid \alpha \circ \beta \mid \alpha \cup \beta \mid \alpha \cap \beta \mid \alpha^{*} \mid \overline{\alpha} \mid [\varphi] \mid \alpha^{n,m}$$

其中 c 遍历 Σ_n ，而 a 遍历 Σ_e。像 φ 这样的公式称为节点表达式，像 α 这样的公式称为路径表达式。Reg-GXPath 的子集 Core-GXPath 规定，克莱尼星号只能应用于标签及其逆向标签（即 a^{-}）。这些语言的语义在 [16] 中定义，其方式与用于导航图的常规正则语言类似

用于导航图 [8],，此外还添加了一些额外功能，例如路径表达式补集 $\overline{\alpha}$ 和数据测试。$\langle\alpha\rangle$ 运算符是嵌套正则表达式（或 NREs）中常用的运算符，见于 [7]。给定一个数据图 $G = (V、 L、 D)$，其语义如下：

$$[\![\epsilon]\!]_G = \{(v,v) \mid v \in V\} \qquad [\![_]\!]_G = \{(v,w) \mid L(v,w) \neq \emptyset\} [\![\mathrm{A}]\!]_G = \{(v,w) \mid \mathrm{A} \in L(v,w)\} \quad [\![\mathrm{A}^-]\!]_G = \{(w,v) \mid \mathrm{A} \in L(v,w)\} [\![\alpha^*]\!]_G = [\![\alpha]\!]_G$的自反传递闭包$[\![\alpha \star \beta]\!]_G = [\![\alpha]\!]_G \star [\![\beta]\!]_G$ 对于 $\star \in \{\circ, \cup, \cap\}$

$$[\![\overline{\alpha}]\!]_G = V \times V \setminus [\![\alpha]\!]_G \qquad [\![[\varphi]]\!]_G = \{(v,v) \mid v \in [\![\varphi]\!]_G\}$$
$$[\![\mathrm{c}^=]\!]_G = \{v \in V \mid D(v) = \mathrm{c}\} \quad [\![\mathrm{c}^{\neq}]\!]_G = \{v \in V \mid D(v) \neq \mathrm{c}\}$$
$$[\![\varphi \wedge \psi]\!]_G = [\![\varphi]\!]_G \cap [\![\psi]\!]_G \qquad [\![\varphi \vee \psi]\!]_G = [\![\varphi]\!]_G \cup [\![\psi]\!]_G$$
$$[\![\neg\varphi]\!]_G = V \setminus [\![\varphi]\!]_G \qquad [\![\langle\alpha\rangle]\!]_G = \{v \mid \exists w \in V, (v,w) \in [\![\alpha]\!]_G\}$$
$$[\![\langle\alpha = \beta\rangle]\!]_G = \{v \mid \exists u,w, (v,u) \in [\![\alpha]\!]_G, (v,w) \in [\![\beta]\!]_G, D(u) = D(w)\}$$
$$[\![\langle\alpha \neq \beta\rangle]\!]_G = \{v \mid \exists u,w, (v,u) \in [\![\alpha]\!]_G, (v,w) \in [\![\beta]\!]_G, D(u) \neq D(w)\}$$

我们用 $\alpha \Rightarrow \beta$ 表示路径表达式 $\beta \cup \overline{\alpha}$，用 $\varphi \Rightarrow \psi$ 表示节点表达式 $\psi \vee \neg\varphi$。我们还将一个标签 A 记为 $\downarrow_{\mathrm{A}}$，以便轻松区分表达式中的“路径”片段。例如，表达式 $\mathrm{A}[\mathrm{c}^=]\mathrm{A}$ 将记为 $\downarrow_{\mathrm{A}} [\mathrm{c}^=] \downarrow_{\mathrm{A}}$。显然，表达式 $\alpha \cap \beta$可以改写为 $\overline{\alpha} \cup \overline{\beta}$，同时保持语义不变。对于使用运算符 $\neg$ 的节点表达式，也会出现类似情况，即运算符 $\wedge$和 $\vee$。我们在此语法中定义了所有这些运算符，因为后文将关注 Reg-GXPath 的一个片段，称为 Reg-GXPath^{pos}；除了$\overline{\alpha}$ 和 $\neg\varphi$ 产生式外，它与原语法相同。因此，在 Reg-GXPath^{pos} 中，除非原始 Reg-GXPath 语法中包含 $\cap$ 运算符，否则我们将无法“模拟”该运算符。

我们还用 Reg-GXPath$^{pos}_{node}$ 表示 Reg-GXPath^{pos} 的一个子集，该子集仅包含节点表达式。

一致性 给定一个特定的数据库，我们希望节点表达式或路径表达式能够表示我们期望在数据中发现的某种结构属性。这类 Core-GXPath 或 Reg-GXPath 表达式通过定义数据之间的语义关系，充当完整性约束。形式上，我们将一致性定义如下：

定义 1（一致性）。设 G 是一个数据图， $R = P \cup N$ 是一组约束，其中 P 由路径表达式组成， N 由节点表达式组成。如果满足以下条件，我们称 G 关于 R是一致的，记作 $G \models R$：(a) 对所有 $\varphi \in N$，都有 $[\![\varphi]\!] = V_G$；(b) 对所有 $\alpha \in P$，都有$[\![\alpha]\!] = V_G \times V_G$。否则，我们称 G 关于 R是不一致的。

在本文其余部分中，只要上下文足以明确限制条件集 R，我们就简称 G 为（一致）或（不一致）。

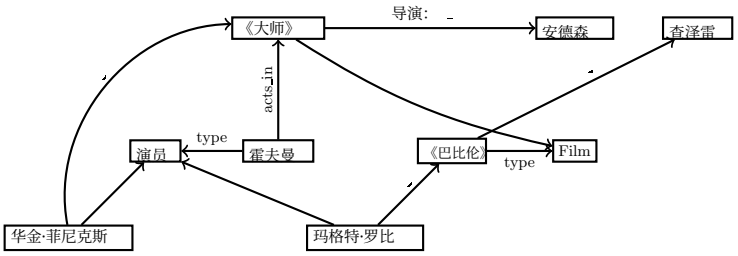

图 1：电影数据图。

示例 1。考虑图 1 中的电影数据库，其中一些节点表示电影产业中的人物（如演员或导演），另一些则表示电影或纪录片。

如果我们想从该图中筛选出只保留通过保罗·托马斯·安德森执导的电影与菲利普·西摩·霍夫曼合作过的演员，那么就需要满足以下公式：

$$\varphi = \langle\downarrow_{\text{TYPE}}[actor^{=}]\rangle \Rightarrow \langle\downarrow_{\text{ACTS_IN}}\langle\downarrow_{\text{DIRECTED_BY}}[Anderson^{=}]\rangle\downarrow^{-}_{\text{ACTS_IN}}[Hoffman^{=}]\rangle.$$

请注意，φ 在所示的数据图中并未得到满足，因为罗比没有与霍夫曼合作出演安德森执导的电影，因此就 $\{\varphi\}$ 而言，我们无法保持一致性。还要注意，该限制条件也适用于霍夫曼，因此他必须至少参演一部安德森执导的电影，才能满足这一约束。

修复 如果图数据库 G 相对于一组约束R 不一致（即 R 中存在未得到满足的路径表达式或节点表达式），我们希望计算出一个新的图数据库 G'，使其相对于 R 一致，且与 G 的差异最小。根据对“最小差异”语义的某种形式化定义，这个新数据库 G' 通常称为 G 相对于 R 的修复。

这里我们考虑集合修复，其中最小差异的概念基于节点集和边集。虽然我们可以通过恰当定义对称差来描述任意数据图之间的距离，但寻找此类修复的复杂度相当高，因此通常考虑集合修复，即一个图仅通过添加或删除信息 [6,18,20]而从另一个图得到。由此产生了子集修复和超集修复，这也是本文的重点。

我们称数据图 $G = (V, L_e, D)$ 是数据图 $G' = (V', L'_e, D')$ 的子集（记作 $G \subseteq G'$），当且仅当 $V \subseteq V'$ 并且对于所有 $v, v' \in V$，都有 $L_e(v, v') \subseteq L'_e(v, v')$，且 $D(v) = D'(v)$。在这种情况下，我们也称 G' 是 G的超集。

定义 2（子集修复和超集修复）。设 R 为一组限制条件，G 为一个数据图。若满足：(a) $G' \models R$，(b) $G' \subseteq G$（分别为 $G' \supseteq G$），则称 G' 为 G 的子集修复（分别为超集修复），或 $\subseteq$-修复（分别为 $\supseteq$-修复），

并且 (c) 不存在数据图 G'' 使得 $G'' \models R$ 且 $G' \subset G'' \subseteq G$ （分别为$G' \supset G'' \supseteq G$）。我们将相对于 R的 G 的子集修复（分别为超集修复）集合记为 $\subseteq$-Rep（G， R）（分别为 $\supseteq$-Rep（G， R））。

示例 2。在示例 1 中，删除值为 Margot Robbie 的节点后，我们得到该图数据库的一个 $\subseteq$-修复。

偏好 现在，我们介绍两种偏好准则，用于为修复集合确定序。按照 [9] 中处理描述逻辑知识库的方式，我们给出图数据库上的权重概念，并可通过由此诱导的序将其转化为偏好。

定义 3（权重函数）。给定函数 w：$\Sigma_e \sqcup \Sigma_n \to \mathbb{N}$（其中 $\sqcup$ 表示不相交并），我们可以将 w 扩展到 Σ_e 和 Σ_n 上的任意有限数据图 $G =$（V， L_e， D），如下所示：

$$w(G) = \sum_{x,y \in V} \left(\sum_{z \in L_e(x,y)} w(z) \right) + \sum_{x \in V} w(D(x)).$$

在考虑如何从多个可能的子集修复或超集修复中选出一个时，一种方法是为不同的边标签和数据值赋予不同的权重，以体现它们的优先级。可以将这些权重汇总，得到整个数据图的权重度量；然后依据非负整数的自然序比较所有可能的修复，从而确定一个优选修复。

定义 4（基于权重的偏好）。给定权重函数 w，我们定义 $G_1 <_w G_2$ iff $w(G_1) < w(G_2)$。

如果 $G_1 <_w G_2$，我们称在超集修复的上下文中， G_1 相较于 G_2 更受偏好（$\boldsymbol{w}$）；而在子集修复的情况下，称 G_2 相较于 G_1 更受偏好（$\boldsymbol{w}$）。

示例 3。考虑这样一种情境：数据图表示物理网络，边表示两种不同的连接质量等级（例如，不同程度的鲁棒性、抵御物理攻击的能力），我们将其称为 $\downarrow_{\text{LOW}}$ 和 $\downarrow_{\text{HIGH}}$。设$R = \{\alpha_{connected_dir}, \alpha_{2l \to good}\}$ 为一组约束，其中

$$\alpha_{connected_dir} = _^*$$
$$\alpha_{2l \to good} = \downarrow_{\text{LOW}}\downarrow_{\text{LOW}} \Rightarrow \downarrow_{\text{HIGH}}\downarrow_{\text{LOW}} \cup \downarrow_{\text{LOW}}\downarrow_{\text{HIGH}} \cup \downarrow_{\text{HIGH}}\downarrow_{\text{HIGH}} \cup \downarrow_{\text{HIGH}} \cup \downarrow_{\text{LOW}} .$$

$\alpha_{connected_dir}$ 表达了有向连通性的概念，而 $\alpha_{2l \to good}$ es- tablishes 表明，如果一个节点可以通过两条低质量边到达，那么也可以通过一条“良好”路径到达。也就是说，要么一步即可到达，要么分两步到达，且至少使用一条高质量边。

我们可以考虑一个权重函数，用来表示构建此网络中的节点和连接的代价。例如，它可以为所有数据值 $w(x) = 20$ 赋予相同的权重，为低质量连接 $w(\downarrow_{\text{LOW}}) = 1$ 赋予较低的代价，并为高质量连接 $w(\downarrow_{\text{HIGH}}) = 5$ 赋予较高的代价。

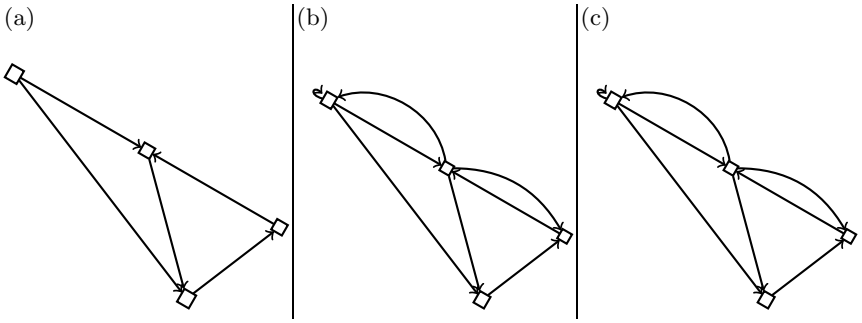

图 2：（a）一个不满足 $\alpha_{connected_dir}$ 也不满足 $\alpha_{2l\to good}$ 的来自例 3 的数据图：它作为有向图并不连通，而且节点对 (c, e) 通过两条 $\downarrow_{\text{LOW}}$ 相连，但无法通过一条「良好」路径相连。（b）针对图（a）示例的一种可能的 w 超集修复；请注意，删除任何一条新增边最终都会违反 R。此修复对应的权重为原始权重加上 $1 + 1 + 5$（来自新增的 LOW 和 HIGH边）。（c）针对图（a）示例的一种 w优选超集修复。此修复对应的额外权重为 3，并且可以证明不存在权重更低的其他超集修复。

现在，给定一个不满足约束的数据图 G，可以将 w-首选超集修复解读为：在最小化 w 给出的成本的同时，构建一个满足约束的网络超集的最具成本效益的方式。完整示例请参见图 2。

通常认为，权重函数 w（定义在 Σ_n 和 Σ_e上）是固定的，并且应当是易于计算的函数。也就是说，给定 $\Sigma_e \sqcup \Sigma_n$的合理编码，我们预期对于每个 $x \in \Sigma_e \sqcup \Sigma_n$ 和某个多项式 $p(n)$，$w(x) \leq 2^{p(|x|)}$，并且还假设 $w(x)$ 的结果应能在关于 $|x|$（即 x的大小）的多项式时间内计算。根据希望建模的权重函数类型，也可以对 w 施加其他限制条件（例如，对于某个多项式 $q(n)$，$w(x) \leq q(|x|)$）；但我们指出，如果不加任何限制条件，w 甚至可能是不可计算函数。

我们研究的第二类偏好准则基于多重集。

定义 5（多重集）。给定集合 A，其有限多重集的集合定义为 $\mathcal{M}_{<\infty}(A) = \{M : A \to \mathbb{N} \mid M(x) \neq 0$ 仅对有限个 $x\}$. 成立。给定严格偏序 $(A, <)$，多重集序 $(\mathcal{M}_{<\infty}(A), <_{mset})$ 定义如下 [12,15]: $M_1 <_{mset} M_2$ 当且仅当 $M_1 \neq M_2$，并且对于所有 $x \in A$，若 $M_1(x) > M_2(x)$，则存在某个 $y \in A$，使得 $x < y$ 且 $M_1(y) < M_2(y)$。

如果（A, $<$）是严格偏序（分别为全序），那么（$\mathcal{M}_{<\infty}$（A） $<_{mset}$）是偏序（分别为全序）。如果（A, $<$）是良基序[4]，，那么（$\mathcal{M}_{<\infty}$（A） $<_{mset}$）也有良基序 [12]。

[4] 也就是说，对于所有 $S \subseteq A$，若 $S \neq \emptyset$，则存在 $m \in S$，使得对于每个 $s \in S$都有 $s \not< m$。

定义 6. 给定一个关于 Σ_e 和 Σ_n 的有限数据图 G，我们将其边和数据值的多重集定义为关于 Σ_e 和 Σ_n 的多重集 $G^{\mathcal{M}}$，使得：

$$G^{\mathcal{M}}(x) = \begin{cases} |[\![x]\!]_G| & x \in \Sigma_e \\ |[\![x^{=}]\!]_G| & x \in \Sigma_n. \end{cases}$$

请注意，所有这些边和数据值的多重集都属于 $\mathcal{M}_{<\infty}(A)$，且满足 $A = \Sigma_e \sqcup \Sigma_n$。

定义 7（基于多重集的偏好）。设 G_1、G_2 是 Σ_e 和 Σ_n 上的两个有限数据图，并设 $<$ 是定义在 $A = \Sigma_e \sqcup \Sigma_n$ 上的偏序。若 $G_1^{\mathcal{M}} <_{mset} G_2^{\mathcal{M}}$，则称 $G_1 <_{Gmset} G_2$。

若 $G_1 <_{Gmset} G_2$，在超集修复的情境下，我们称 G_1 在 M 偏好上优于 G_2；而在子集修复的情境下，则称 G_2 在 M 偏好上优于 G_1。

例 4。考虑图 2 中的数据图。忽略任何可能的数据值，可以看到图 b) 和 c) 对应的多重集分别为（采用非正式的多重集符号表示）：$\{\text{LOW}, \text{LOW}, \text{LOW}, \text{LOW}, \text{LOW}, \text{HIGH}, \text{HIGH}, \text{HIGH}\}$ 和 $\{\text{LOW}, \text{LOW}, \text{LOW}, \text{LOW}, \text{LOW}, \text{LOW}, \text{HIGH}, \text{HIGH}\}$。假设 $\text{LOW} > \text{HIGH}$，那么在此情况下，数据图 b) 在 $\mathcal{M}$ 偏好上优于 c)。

3 优选修复

本节讨论子集修复和超集修复，以及 Ω-优选准则，其中 $\Omega \in \{w, \mathcal{M}\}$，并使用 Reg-GXPath 的不同子集来处理 $\mathcal{L}$。

我们始终假设权重函数 w：$\Sigma_e \sqcup \Sigma_n \to \mathbb{N}$ 是固定且可高效计算的：给定大小为 n 的 $x \in \Sigma_e \sqcup \Sigma_n$ 的编码，可以在 $poly(n)$ 时间内计算其值 $w(x)$。这意味着，对于任意数据图 G，$w(G)$ 也可以在 $poly(|G|)$ 时间内计算。

同样，在考虑多重集优选修复时，我们假设定义在 $\Sigma_e \sqcup \Sigma_n$ 上的序关系 $<$ 可以高效计算：给定 x，$y \in \Sigma_e \sqcup \Sigma_n$，可以在 x 和 y 的表示上于多项式时间内判定 $x < y$。因此，我们可以在 $poly(|G_1| + |G_2|)$ 中判定 $G_1 <_{Gmset} G_2$。此外，我们假设序关系 $<$ 是良基的。这意味着不存在无限的下降链 $\text{c}_1 > \text{c}_2 > \ldots$

请注意，基于权重和基于多重集的偏好会在有限数据图上诱导出一种序，且该序不存在无限下降链。因此，如果存在超集修复，那么也必然存在一个（基于权重或基于多重集的）优选修复。另一方面，给定数据图的子集数量有限，因此如果存在修复，就必然存在一个优选修复。

当表达式集合 R 固定时，这些问题的复杂度通常称为数据复杂度。我们推导出的多数下界都适用于这种情况。

3.1 首选子集修复

对于 $\subseteq$-修复， [2] 证明：对于固定的 Reg-GXPath^{pos} 表达式集合 R，判定是否存在非平凡修复（即不同于 $\emptyset$ 数据图）是一个 NP-COMPLETE 问题；如果只允许将 Reg-GXPath^{pos} 中的节点表达式用作 $\mathcal{L}$，则该问题是可处理的。请注意：

命题 1. 判断 G 是否存在相对于 R 的非平凡 $\subseteq$-修复，可归约为判断 G 是否存在相对于 R 的非平凡 Ω-首选 $\subseteq$-修复。

给定一个固定的权重函数 w （分别地，一个序 $<$），如果 G 相对于 R 存在一个 $\subseteq$- 修复 G'，则 G 存在一个首选 $\subseteq$- 修复（反之亦然）。因此，直接由命题 1 和 [2] 中上述结果可得：

定理 1. 对于给定的数据图 G 和一组 Reg-GXPath^{pos} 表达式 R，判断是否存在非平凡的 Ω-首选$\subseteq$-修复的问题，对于固定的一组 Reg-GXPath^{pos} 路径表达式而言，属于 NP-COMPLETE 。

当 $\mathcal{L} \subseteq$ Reg-GXPath$^{pos}_{node}$ 时，可以在多项式时间内计算出一个子集修复，而且该修复是唯一的 [2]。因此，它必定是首选修复：

定理 2。给定数据图 G、一组 Reg-GXPath^{pos} 表达式 R和偏好准则 Ω，存在一种算法，可以计算出相对于 R 的 G 的 Ω优选$\subseteq$修复。

3.2 优选超集修复

存在一个限制条件集 R，其中仅包含 Reg-GXPath$^{pos}_{node}$ 表达式，使得计算 w 优选 $\supseteq$修复已是难解问题：

定理 3。给定数据图 G、一组 Reg-GXPath$^{pos}_{node}$ 表达式 R和一个自然数 K，令 Π_w 表示判定是否存在一个相对于 R 、权重不超过 K 的 G 的 w- 优选 $\supseteq$ 修复的问题。那么，存在一组肯定节点表达式 $\mathcal{R}$ 和一个权重函数 w，使得该问题是 NP-COMPLETE。

证明。一般而言，该问题属于 NP：如果存在一个相对于 R 的 G 修复，根据 [2, 定理 24]，那么就存在一个“标准形式”的修复。该修复的大小受 $poly(|G| + |R|)$ 限制；通过检查定理的证明，并考虑到 w 是一个非负函数，可以证明优选修复始终具有这种“标准形式”。于是，肯定证书由一个数据图 G' 构成，满足 $|G'| \leq poly(|G| + |R|)$、 $G' \models R$、 $G \subseteq G'$和 $w(G') \leq K$。

对于困难性，我们将 3-SAT 降低到 Π_w，并固定 R 。对于这种归约，我们考虑 $\Sigma_e = \{\text{VALUE_OF}, \text{APPEARS_IN}, \text{APPEARS_NEGATED_IN}\}$ 和 $\Sigma_n =$

$\{clause, var, \top, \bot\}$。我们将权重函数 w 定义为 $w(c) = 2$，适用于 $c \in \Sigma_n \cup \{\text{APPEARS_IN}, \text{APPEARS_NEGATED_IN}\}$ 和 $w(\text{VALUE_OF}) = 1$。

给定一个包含 n 个变量和 m 个子句的 3-CNF 公式 ϕ，我们构造一个数据图 G、一组 Reg-GXPath$^{pos}_{node}$ 节点表达式 R，并定义一个数 K，使得 ϕ 可满足，当且仅当 G 相对于 R 存在一个超集修复 G'，使得 $w(G') \leq K$。

我们将图定义为 $G = (V_G, L_G, D_G)$，其中：

$$V_G = \{x_i \mid 1 \leq i \leq n\} \cup \{c_j \mid 1 \leq j \leq m\} \cup \{\bot, \top\}$$

$$L_G(x_i, c_j) = \begin{cases} \{\text{APPEARS_IN}, \text{APPEARS_NEGATED_IN}\} \text{ if } x_i \text{ and } \neg x_i \text{ appear in } c_j \\ \{\text{APPEARS_IN}\} \text{ if only } x_i \text{ appears in } c_j \\ \{\text{APPEARS_NEGATED_IN}\} \text{ if only } \neg x_i \text{ appears in } c_j \\ \emptyset \text{ otherwise} \end{cases}$$

对于其他每一对 $v, w \in V_G$，都有$L_G(v, w) = \emptyset$
对于 $x \in \{\bot, \top\}$，有$D_G(x) = x$；对
于 $1 \leq i \leq n$，有$D_G(x_i) = var$；对
于 $1 \leq j \leq m.$，有$D_G(c_j) = clause$

ϕ的结构由 APPEARS条边编码。我们定义 $K = w(G) + n$，并希望 G 相对于 R 的任意权重为 K 的超集修复，都能利用边 VALUE_OF对“节点”变量的赋值进行编码。为此，我们定义 Reg-GXPath$^{pos}_{node}$ 表达式

$$\psi_1 = \langle [var^{\neq}] \cup \downarrow_{\text{VALUE_OF}} [\top] \cup \downarrow_{\text{VALUE_OF}} [\bot] \rangle \quad (1)$$

$$\psi_2 = \langle [clause^{\neq}] \cup \downarrow^{-}_{\text{APPEARS_IN}} \downarrow_{\text{VALUE_OF}} [\top] \cup \downarrow^{-}_{\text{APPEARS_NEGATED_IN}} \downarrow_{\text{VALUE_OF}} [\bot] \rangle. \quad (2)$$

表达式 ψ_1 强制每个变量节点都有一条指向布尔节点的 VALUE 边，而表达式 ψ_2 则强制每个子句在 G的任何修复中都得到“满足”。因此，我们定义 $R = \{\psi_1, \psi_2\}$。现在证明，ϕ 可满足，当且仅当 G 相对于 R 存在权重至多为 $w(G) + n$的超集修复。

$\Longrightarrow$) 设 f 是 ϕ 的变量上的一个赋值，它将 ϕ 求值为真。然后，我们按以下方式向 G “添加”边：若 $f(x_i) = \top$，则添加边(x_i,VALUE_OF, $\top$)；否则添加(x_i, VALUE_OF, $\bot$)。该图满足 R 中的两个表达式，且代价为 $w(G) + n$，因为 $w(\text{VALUE_OF}) = 1$。

$\Longleftarrow$) 设 G' 是相对于 R、权重至多为$w(G) + n$的 G 超集修复。由于 G的每个超集修复都必须至少添加 n VALUE条边，且每条边的代价为1个单位，因此我们知道，G' 必定恰好是原图 G加上 n VALUE 条边（每个变量节点一条）。于是，我们可以利用这些边定义 ϕ 的变量赋值：如果边(x_i, VALUE_OF, $\top$)存在于 G'中，就定义 $f(x_i) = \top$；否则定义 $f(x_i) = \bot$。由于 ψ_2在 G'中得到满足，这个赋值必定满足 ϕ。□

Π_w 的困难性意味着，除非 P = NP，否则不存在算法能够针对固定的 Reg-GXPath$^{pos}_{node}$ 表达式计算 w优选的 $\supseteq$修复。

此外，对于简单的 Reg-GXPath$^{pos}_{node}$ 表达式，计算多重集优选修复也是一个困难问题：

定理 4。给定一个数据图 G、一组 Reg-GXPath$^{pos}_{node}$ 表达式 R、一个边标签 $\mathrm{A} \in \Sigma_e$ 和一个自然数 K，令 $\Pi_{\mathcal{M}}$ 表示判定问题：是否存在 G 的一个关于 R 的 $\mathcal{M}$-优选 $\supseteq$-修复 G'，使得 G' 中标签为 A的边至多有K 条。于是，存在一个正节点表达式集合 R 和一个良序 $<$，使得该问题是 NP-COMPLETE。

证明。考虑序$\textsc{value_of} < \textsc{appears_in} < \textsc{appears_negated_in} < clause < var < \top < \bot$ 和 $K = n$，证明与定理 3 的证明类似。□

$\Pi_{\mathcal{M}}$ 的困难性意味着，对于固定的 Reg-GXPath$^{pos}_{node}$ 表达式集合，计算 $\mathcal{M}$-优选 $\supseteq$-修复的困难性。

备注 1。对于 $\supseteq$-修复，文献 [2] 已证明：对于固定的 Reg-GXPath 表达式集合，判定是否至少存在一个修复是不可判定问题；并且， NP-HARD 如果 Reg-GXPath^{pos} $\subseteq \mathcal{L}$。与此同时，研究表明，如果 $\mathcal{L} \subseteq$ Reg-GXPath^{pos} 且限制条件集合固定，或者更确切地说，如果 $\mathcal{L} \subseteq$Reg-GXPath$^{pos}_{node}$，则存在用于计算超集修复的多项式时间算法。定理 3 和定理 4 表明，考虑优选修复时，这些可处理情形会变得棘手。

4 结论

在这项工作中，我们分析了数据图的优选修复。我们特别关注这样一个问题：在两种不同的数据感知偏好准则下，判断一个数据图 G 是否具有非平凡的优选修复；其中一种准则基于权重，另一种基于多重集序。我们证明，在某些情况下，与缺乏偏好的版本相比，这些准则不会使修复判定问题变得更加困难。

这一情境下仍有一些问题尚未解决，例如寻找可能基于现实世界应用的、更精细的可处理版本。此外，超集修复的确切复杂度类仍然未知，因为我们目前仅为该问题证明了一个下界。值得在基于偏好的情境下研究数据图 [6], 的其他修复类型定义，例如基于对称差 [20], 的定义。最后，研究更一般的准则族（例如 [19],中提出的准则），并分析判定问题的复杂度是否会随数据图的变化而改变，也将十分有意义。

参考资料

1. S. Abiteboul 和 V. Vianu。带约束的正则路径查询。 *Journal of Computer and System Sciences* 58 3 428 452, 1999 , ( ): – 。

2. S. Abriola、S. Cifuentes、M. V. Martínez、N. Pardal 和 E. Pin。关于为数据图寻找集合修复的复杂性。 *Journal of Artificial Intelligence Reasoning*, 76 721 759, 2023 : – 。 3. M. K. Anand、S. Bowers 和 B. Ludäscher。高效查询科学工作流溯源图的技术。载于 *EDBT*，第 10 卷，第 287–298 页，2010 年。 4. M. Arenas、L. Bertossi 和 J. Chomicki。不一致数据库中的一致查询答案。载于 *PODS*，第 99 卷，第 68–79 页。Citeseer，1999 年。 5. M. Arenas 和 J. Pérez。使用 SPARQL 查询语义网数据。载于 *Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* 305 316, 2011 ，第 – 页。

6. P. Barceló 和 G. Fontaine。图数据库上一致查询回答的数据复杂度。 *Journal of Computer and System Sciences*, 88:164–194, 2017。 7. P. Barceló、J. Pérez 和 J. L. Reutter。嵌套正则表达式的相对表达能力。 *AMW*, 12:180–195, 2012。 8. P. Barceló Baeza。查询图数据库。载于 *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGAI symposium on Principles of database systems*，第 175–188 页。ACM，2013 年。 9. M. Bienvenu、C. Bourgaux 和 F. Goasdoué。在优先修复语义下查询不一致的描述逻辑知识库。载于 *Proceedings of the AAAI Conference on Artificial Intelligence* 28, 2014 ，第 卷。

10. G. Brewka。优先子理论：一种扩展的默认推理逻辑框架。载于 N. S. Sridharan 编，*Proceedings of the 11th International Joint Conference on Artificial Intelligence. Detroit MI USA August 1989* 1043 ，，，，第 – 1048 页。Morgan Kaufmann，1989 年。 11. P. Buneman、W. Fan 和 S. Weinstein。半结构化数据库中的路径约束。 *Journal of Computer and System Sciences*, 61(2):146–193, 2000。 12. N. Dershowitz 和 Z. Manna。使用多重集序证明终止性。 *Communications of the ACM* 22 8 465 476, 1979 , ( ): – August 。 13. W. Fan。为社交网络分析重新审视图模式匹配。载于 *Proceedings of the 15th International Conference on Database Theory* 8 21, 2012 ，第 – 页。

14. S. Flesca、F. Furfaro 和 F. Parisi。聚合约束下的优先数据库修复。载于 *International Conference on Scalable Uncertainty Management*，第 215–229 页。施普林格，2007 年。 15. G. Huet 和 D. C. Oppen。方程与重写规则：综述。载于 *Formal Language Theory* 349 405 1980 ，第 – 页。爱思唯尔，。 16. L. Libkin、W. Martens 和 D. Vrgoč。带数据图查询。*Journal of the ACM (JACM)* 63 2 1 53, 2016 , ( ): – 。 17. A. Lopatenko 和 L. Bertossi。基于基数和增量修复语义的数据库中一致查询回答的复杂性。载于 *In ICDT*，第 179–193 页。施普林格，2007 年。 18. T. Lukasiewicz、M. V. Martinez 和 G. I. Simari。Datalog + *OTM Confederated International Conferences On the Move to Meaningful Internet Systems* 488 500 /– 中容忍不一致的查询回答的复杂性。载于 ，第 – 页。施普林格，2013 年。 19. S. Staworko、J. Chomicki 和 J. Marcinkowski。关系数据库中的优先修复与一致查询回答。*Annals of Mathematics and Artificial Intelligence* 64 2 209 246, 2012 , ( ): – 。 20. B. ten Cate、G. Fontaine 和 P. G. Kolaitis。一致查询回答的数据复杂度。载于 *Proceedings of the 15th International Conference on Database Theory* 12, 22 33, 2012 ，ICDT ’ 第 – 页。

21. J. Wijsen. 用于一致性查询回答的数据库修复的精简表示。载于
Proceedings of the 9th International Conference on Database Theory, ICDT '03, 第 378–393 页。施普林格出版社，2002。