arXiv:2601.06764v1 [cs.DB] 11 Jan 2026

寻找缺失答案修复的复杂性

Jesse Comer 美国宾夕法尼
亚大学**Val Tannen** 美国宾
夕法尼亚大学

摘要

我们研究如何为查询答案中的缺失元组确定数据库修复这一问题。我们证明，当查询属于输入的一部分——即联合复杂性情形——对于允许使用弱形式的投影和选择的查询类，判定是否存在修复等价于可满足性问题。随后，我们根据查询中允许出现的关系代数运算，确定了带否定原子的合取查询的并的哪些子类能够在多项式时间内解决最小修复问题。与此相反，我们证明，只要输入查询同时允许投影和连接，该问题就是**NP**难的，并且通过严格归约，在近似方面也至少与集合覆盖问题一样困难。此外，我们证明，求带（允许出现否定原子的）合取查询的并的极小修复的大小是$\mathbf{OptP}[\log(n)]$-完全的，而计算一个极小修复只需进行$O(n^2)$次针对**NP** 预言机的查询。允许递归时，所有这些变体的联合复杂性都会显著增加，下界达到**EXP** 。然而，从数据复杂度角度来看，我们证明，对于所有可由半正 Datalog 程序表示的查询，都能在多项式时间内确定极小修复。

2012年 ACM 学科分类计算理论 → 复杂性理论与逻辑；计算理论 → 逻辑与数据库

关键词与短语缺失答案、数据库修复、Datalog、计算复杂性

1 引言

我们研究以下缺失答案修复问题的**计算复杂性**：给定一个查询 Q、一个数据库实例$\mathbf{I}$ （即输入或源数据库），以及一个可能不属于答案 $Q(\mathbf{I})$ 的元组 a，该问题要求找到对$\mathbf{I}$的一项更新 U（即对$\mathbf{I}$中的元组进行插入和删除所组成的集合，这些操作统称为修复），从而得到一个修复后的实例$\mathbf{I}_{\mathrm{U}}$，使得 $a \in Q(\mathbf{I}_{\mathrm{U}})$。「找到」一项修复的概念可以通过四种方式精确定义：判定是否存在修复（$\mathbf{MA}_{dec}$）、判定是否存在大小至多为 k 的修复（$\mathbf{MA}_{bound}$）、确定极小修复的大小（$\mathbf{MA}_{size}$），以及计算一个大小最小的修复（$\mathbf{MA}_{min}$）。对于问题的每个版本，我们都考察联合复杂性（此时问题以查询类为参数）和数据复杂度（此时问题以一个固定查询为参数）。在这两种情形下，我们都假设数据库模式固定。缺失答案问题与文献中已有深入研究的两个概念相关：视图更新和数据库修复。我们概述这两个概念，并说明我们的结果与以往结果有何不同。我们还讨论否定如何将缺失答案修复问题与文献中的其他类似问题联系起来。

视图更新。 缺失答案修复问题可应用于视图更新问题 [18]：如果 Q 定义了一个视图，那么缺失答案修复问题就是寻找对输入数据库实例的更新，使得给定元组被插入 到 Q 在更新后的实例上定义的视图中。据本文作者所知，迄今为止，大多数关于视图复杂性 的研究都聚焦于寻找实例更新，以便从视图中删除 一个元组。此前的工作仅限于单调查询，因此更新仅限于删除输入数据库中的元组（参见关于删除

传播 [8, 15, 32, 31, 43] 和韧性 [21, 22, 37]的研究）。与此不同，我们关注视图插入而非视图删除，并考虑更一般的（非单调）查询。因此，我们的更新既包括插入，也包括删除。此外，与从查询中删除元组的相关结果相比，我们的结果乐观得多。与韧性不同，后者的数据复杂度即使对于某些合取查询也是**NP**难的；我们在此证明，对于所有半正 Datalog 程序，缺失答案修复问题在数据复杂度意义下可在多项式时间内求解。

关于插入传播的研究也已有开展，该研究旨在向视图中添加元组；此外，还有关于注释传播的研究，旨在修改源数据库中某个元组的注释，从而修改视图中目标元组的注释 [8, 52, 14, 15]。以往关于这些问题的研究与本文所研究的缺失答案问题存在重要差异。为理解这些差异，请注意，一般而言，以更新视图为目标对数据库实例执行的更新并不唯一。因此，视图更新方面的研究始终强调寻找满足某些预期标准的实例更新 [18]。如前所述，我们关注的是向查询结果中插入元组，同时尽量减小应用于数据库的更新规模。在随后对相关工作的讨论中，我们将概述人们所考虑过的其他更新标准。

数据库修复。数据库修复的概念源于一致查询回答（CQA）领域的文献 [2]。在该语境下，给定一组完整性约束 IC 和一个不一致数据库实例 I（即不满足 IC 的实例），修复是一个经过极小修改的一致数据库实例 I′。一致查询回答的目标是确定给定元组是否为确定答案，即对于给定查询 Q 的所有修复 I′，该元组都出现在查询结果中。有多种方式可以定义 I′ 相对于 I “经过极小修改”的含义。与本文所采用概念最为相近的是基数修复语义 [3, 36]。给定一个不一致数据库实例 I 以及一致数据库实例 I′ 和 I″，若 $|\mathrm{I} \oplus \mathrm{I}'| \leq |\mathrm{I} \oplus \mathrm{I}''|$，则记作 I′ $\leq_C^{\mathbf{I}}$ I″（其中 $S \oplus T$表示集合 S和 T的对称差）。在采用基数修复的一致查询回答中，目标是找出对于所有 $\leq_C^{\mathbf{I}}$-极小一致数据库 I′都出现在 Q(I′) 中的元组。换言之，基数修复中的极小性由输入数据库与修改后数据库之间对称差的大小定义。

基数修复语义下的一致性查询回答（CQA）与缺失答案修复问题在若干重要方面有所不同。在 CQA 中，修复 是一个 $\leq_C^{\mathbf{I}}$-极小的数据库，并且满足输入中的完整性约束**IC** 。相比之下，对我们而言，修复是指任何一种更新，只要它能使输入元组成为更新后数据库上输入查询的答案即可。这意味着，给定一个输入数据库和输入查询，极小修复的大小还取决于输入元组。直观地说，我们不是固定修复大小，再寻找所有修复中都出现在查询答案里的元组，而是固定元组，再寻找最小修复。由于存在这些差异，这两个问题的计算复杂性如何关联，并不能立即看清。

数据库修复出现的另一个背景是数据清洗 [20]。例如，清洗从 Web 收集的数据，其中一个方面是恢复预期的完整性约束，因此需要修复不一致的数据库实例（例如 [33, 57, 5, 10, 24]）。这与一致性查询回答（CQA）类似，只不过首要目标仅仅是恢复一致性。如果我们考虑一种能够表达某些完整性约束（通常通过布尔查询为真来表达）的查询语言，那么缺失答案修复问题就是要寻找一个最小基数修复，以便将不一致的数据库更新为一致的数据库。我们认为本文的工作与这一研究方向并无特别密切的关系，因为表达常见的完整性约束形式（例如函数依赖和包含依赖，或元组-

查询类	$\mathbf{MA}^{\mathcal{L}}_{dec}$	$\mathbf{MA}^{\mathcal{L}}_{bound}$	$\mathbf{MA}^{\mathcal{L}}_{size}$	$\mathbf{MA}^{\mathcal{L}}_{min}$
合取查询的并集¬[sjf]	**P**	**P**	**P**	**P**
$\mathbf{SJU}^{\neg}$	**P**	**P**	**P**	**P**
$\mathbf{SPU}^{\neg}$	**P**	**P**	**P**	**P**
PJ	**P**	**NP**-完全	$\mathbf{OptP}[\log(n)]$-完全	$\mathbf{FP}^{\mathbf{NP}[n^2]}$
$\mathbf{UCQ}^{\neg}$	**P**	**NP**-完全	$\mathbf{OptP}[\log(n)]$-完全	$\mathbf{FP}^{\mathbf{NP}[n^2]}$
Datalog	**EXP**-完全	**EXP**-完全	**FEXP**	**FEXP**
sp-Datalog	**EXP**-完全	**2EXP**	**F2EXP**	**F2EXP**
RA	不可判定	**2NEXP**	不可计算	不可计算

表 1 联合复杂性结果。我们用**合取查询的并**¬[sjf] 表示无自连接**合取查询的并**¬ 查询类。请注意，$\mathbf{SJU}^{\neg}$（分别为$\mathbf{SPU}^{\neg}$）表示无投影（分别为无连接）**合取查询的并**¬ 查询类。类似地，**PJ** 表示无选择合取查询类。Datalog 的标签“sp”代表“半正”（即包含原子否定的程序）。我们在第2节中正式定义这些类。

生成依赖）时，若不在查询中使用某种形式的全称量化，就无法实现。相比之下，我们在此关注的是半正 Datalog 的片段，它不包含全称量化[1]。

缺失答案与否定。 由于删除传播和韧性研究的是通过修改数据库来向查询答案中添加元组，因此可将它们恰当地解释为错误答案 修复问题。缺失答案修复与错误答案修复的关系如下。给定查询 Q，令 $\neg Q$ （即 Q 的否定；参见注释 2.3）表示一个查询：对于每个数据库实例$\mathbf{I}$，它返回 Q的补集（相对于$\mathbf{I}$的活动域上所有具有适当元数的元组）。显然，从 Q 的答案中删除一个元组的问题，与向 $\neg Q$ 的答案中插入一个元组的问题相同。先前关于错误答案修复的研究主要聚焦于非递归单调查询，尤其是合取查询或合取查询的并 [14, 15, 21, 40, 41, 44, 45]。相比之下，本文聚焦于半正 Datalog 及其片段。虽然否定直接关联了缺失答案修复与错误答案修复，但对合取查询的并应用否定会得到全称 （而非存在性）查询，而这些查询无法用本文所考虑的半正 Datalog 子类表示。

贡献。 我们在联合复杂性设置下考虑该问题的四个版本：$\mathbf{MA}^{\mathcal{L}}_{dec}$、$\mathbf{MA}^{\mathcal{L}}_{bound}$、$\mathbf{MA}^{\mathcal{L}}_{size}$和$\mathbf{MA}^{\mathcal{L}}_{min}$，其中问题以查询类$\mathcal{L}$为参数。我们的主要联合复杂性结果总结如下表所示。

表 1 中的一些结果并不紧确；在这种情况下，表中仅列出了我们对该问题复杂度所能给出的最佳上界。例如，我们给出的 $\mathbf{MA}^{\mathcal{L}}_{min}$ 关于$\mathcal{L} \in \{\mathbf{PJ}, \mathbf{合取查询的并集}^{\neg}\}$ 的上界是 $\mathbf{FP}^{\mathbf{NP}[O(n^2)]}$, ，这与我们得到的 $\mathbf{FP}^{\mathbf{NP}[O(\log(n))]}$下界并不相符。不过，对于（优化问题的函数版本的）预言机复杂度，此类结果通常就是如此。例如，据作者所知，

[1] 然而，需要注意的是，某些完整性约束（例如元组生成依赖）可以在全称量化变量用常量实例化时用半正Datalog表示。从这个意义上说，我们的结果表明，元组生成依赖可以在多项式时间内按“逐元组”方式进行修复（每个元组的数据复杂度）。[2] $\mathbf{FP}^{\mathbf{NP}[O(\log(n))]}$ 和$\mathbf{FP}^{\mathbf{NP}[O(n^2)]}$ 的形式化定义见第2节。

查询类	$\mathbf{MA}^{Q}_{bound}$	$\mathbf{MA}^{Q}_{size}$	$\mathbf{MA}^{Q}_{min}$
sp-Datalog	**P**	**P**	**P**
CCQ	**P类** 或**NP**-完全	**P类** 或$\mathbf{FP}^{\mathbf{NP}[\log(n)]}$	**P类** 或$\mathbf{FP}^{\mathbf{NP}}$
RA	可判定	可计算	可计算

表 2 数据复杂度结果。我们用**CCQ** 表示形如 $\neg Q$（见注释 2.3）的查询类，其中 Q 是一个合取查询。

对于最小集合覆盖、最小顶点覆盖、最大团等优化问题，构造最优解的已知最佳下界是 $\mathbf{FP}^{\mathbf{NP}[O(\log(n))]}$，而这些问题目前已知的最紧上界是 $\mathbf{FP}^{\mathbf{NP}[O(n)]}$。另一个上下界不紧确的例子是 $\mathbf{MA}^{\mathbf{sp\text{-}Datalog}}_{bound}$ 的复杂度。我们在第 4 节证明，该问题是**EXP**难的，且属于**2EXP**，但其精确复杂度仍有待确定。

除了表 1 所示的结果外，我们还通过严格归约（ [48]中定义的L-归约的一种特殊情况；见定义 2.5）给出了$\mathbf{MA}^{\mathbf{PJ}}_{min}$ 的最小集合覆盖近似难度结果。 [34] 中引入并在 [23], 中进一步阐述的**OptP**类，用于刻画确定**NP**完全优化问题最优值的复杂度。鉴于$\mathbf{MA}^{\mathbf{PJ}}_{bound}$是**NP**完全的，**OptP**自然适合用于刻画$\mathbf{MA}^{\mathbf{PJ}}_{size}$的复杂度。

在数据复杂度方面，我们研究$\mathbf{MA}^{Q}_{dec}$、$\mathbf{MA}^{Q}_{bound}$和$\mathbf{MA}^{Q}_{min}$ 问题，其中查询 Q 是固定的；结果汇总于表 2。请注意，对于**RA** 查询 Q ，$\mathbf{MA}^{Q}_{bound}$、$\mathbf{MA}^{Q}_{size}$和$\mathbf{MA}^{Q}_{min}$ 问题的复杂度因查询而异；我们将在第 5 节末讨论这些问题的可判定性。此外，关于**CCQ** 的结果可直接由引言中涉及否定的观察以及 [21, 22] 中关于某些合取查询子类韧性二分性的结果得出。该类的$\mathbf{MA}^{Q}_{size}$ 和$\mathbf{MA}^{Q}_{min}$ 结果可由标准论证得出（[47, 第 415-422] 页）：利用**NP** 预言机查询进行二分搜索，以确定最优修复的大小；并利用**NP** 预言机查询进行多项式时间构造，以得到一个极小修复。我们列出这一行，是为了说明即使对于合理的查询类，各种缺失答案问题在数据复杂度方面也可能难以处理。

相关工作。 在联合复杂性下，缺失答案修复问题推广了查询可满足性 [1] （回顾一下，如果存在一个数据库实例**I**，使得 $Q(\mathbf{I}) \neq \emptyset$，那么查询 Q 就是可满足的）。更准确地说，布尔查询 Q 可满足，当且仅当对于缺失答案判定问题的输入$\langle Q,\mathbf{E}\ \epsilon\rangle$ 存在一个修复，其中 **E** 和 ϵ 分别为空数据库和空元组。在第 4 节中，我们将这一观察推广到非布尔查询，并利用这种关系，将已知的可满足性复杂度下界转移到许多类查询的缺失答案问题上。

正如引言中所提到的，通常有许多数据库更新都可能实现视图中期望的变更。因此，关于视图更新的研究重点在于寻找能够实现特定目标的更新：（1）尽量减小实例更新的规模；（2）在更新实例的同时保留指定的完整性约束；（3）确定是否存在一种视图更新，不会引起其他视图“副作用”；（4）尽量减少视图上的“副作用”。请注意，视图副作用 指的是除原先期望的更新之外，对视图中元组所做的额外更改。

我们关注这样的视图插入：它使对数据库所做的插入和删除数量最少（即上述目标（1））。先前工作对这个问题已有一定研究。具体而言，已有研究表明，对于无选择合取查询，在

联合复杂性意义下，计算视图插入的极小修复是**NP**难的；而对于缺少连接或投影的合取查询的并，则可以在多项式时间内完成。我们从几个方面细化了这一结果。首先，我们针对无选择合取查询给出了类似的困难性结果，即使假设模式是固定的也是如此。相比之下，[14] 允许模式中出现任意多个关系，其困难性证明依赖于这一事实。其次，我们证明，多项式时间结果可以扩展到带有原子否定的查询。最后，我们给出了这些问题的函数版本（$\mathbf{MA}_{size}$ 和$\mathbf{MA}_{min}$）在**OptP** 和$\mathbf{FP}^{\mathbf{NP}}$ 中的具体上界。其他关于插入传播的工作则着重于目标（2）和（3），尤其是计算对源数据库的更新，使其在不产生视图副作用的情况下满足函数依赖等完整性约束 [43, 44]。注释传播方面的许多工作也聚焦于此 [14, 15]。在删除传播和韧性的研究中，目标（1）至（4）都已有所研究 [8, 14, 15, 32, 31, 21, 41, 40, 45, 22, 42]，不过所用查询语言并不像本文研究的语言那样强大。

在 [41]，中，作者定义了一个有界插入传播问题，并证明该问题对于合取查询在联合复杂性意义下是Σ_2^P-完全的。该论文定义的问题类似于我们的**缺失答案**$_{bound}$ 问题，但有一些重要区别。首先，模式不是固定的。其次，假设关系具有类型化属性，并且每个属性的域都在输入中指定。相比之下，我们假设模式是固定的，而且任何数据（甚至是输入结构的活动域之外的数据）都可能出现在修复中。这两个区别在 Σ_2^P-完全性证明中都起着重要作用，见 [41, 定理 4]，，因此显然该问题在计算上大不相同。特别是，我们的结果意味着，对于合取查询，**缺失答案**$_{bound}$ 问题是**NP**-完全的。

本文的工作与先前关于缺失答案和错误答案问题的研究之间，一个显著区别在于我们考虑了功能更强大的查询语言。我们考虑合取查询、合取查询的并以及 Datalog，也考虑它们扩展为包含否定（外延）原子的情形。特别是，这类扩展要求修复同时包含插入和删除，而不只是插入；仅插入足以修复单调查询的缺失答案。就联合复杂性而言，与可满足性的关系适用于任意查询类，这些查询类仅需对与弱形式的选择或投影进行复合保持封闭。相比之下，先前关于错误答案的研究聚焦于合取查询的并（UCQ） [8, 15] ，或仅聚焦于合取查询（CQ） [32, 31, 21, 22]。另一个显著区别是，我们允许插入的数据不必属于输入数据库的活动域**I** 或查询的活动域 Q。对于不含否定的查询，这不会造成差别；然而，如上所述，我们也考虑含有否定原子的查询。相比之下，先前关于任意一阶查询的缺失答案研究 [24, 57] 将插入限制为由活动域元素构成的数据。

此外，还值得强调一种不同的视图更新方法，该方法源自 [18]，，作者在其中定义了更新翻译，并给出了正确翻译视图更新的条件。 [4], 进一步扩展了这项工作，提出了视图补集的概念，它是一种将数据库实例映射到视图对的特殊映射。请注意，翻译和补集的定义独立于特定的数据库模型。在 [16], 中，作者将这一框架专门应用于关系数据库模型，并采用一种简化设定：仅包含一个关系、函数依赖，以及通过对该关系进行投影来定义的视图。作者给出了若干与视图插入和视图补相关的判定问题的复杂度结果。一般而言，这些问题似乎处于多项式层次的较低层级；要确定这一框架与本文所研究的缺失答案修复问题之间的

关系，还需要进一步研究。

寻找缺失答案的解释 也采用了多种技术进行研究 [12, 28, 27, 54, 56, 50, 7, 6, 49], ，尤其是使用来源 [9, 25, 29, 24, 57] 和因果关系的形式化方法。后者的研究讨论了量化原因相对重要性的两种方式：责任度 [39, 38] 和韧性 [21, 22, 37]。事实上，计算韧性本质上就是计算错误答案修复。

2 预备知识

基本记号和定义。我们固定两个可数无限集合 Var 和 Dom，分别由变量和常量组成。项是 Var ∪Dom 的元素。赋值是一个部分映射 g: Var $\to$ Dom。我们用 x, y, z, u, v 表示变量，用 a, b, c, d, e 表示常量，并用 s, t 表示任意项。我们使用上划线记号表示变量或常量元组，并用 len(a) 表示元组 a 的长度。给定一个项元组 t 和一个赋值 g，我们用 $g(t)$ 表示将 t 中所有变量替换为其在 g 下的像后得到的常量元组。我们有时会滥用记号，将元组视为集合。我们固定一个有限模式 Σ，其中包含外延数据库谓词（EDB），并固定一个无限模式 S，其中包含内涵数据库谓词（IDB）。每个关系符号 $F \in \Sigma \cup$ S 都关联一个自然数元数（记作 arity(F)），并且我们假设 S 对每个元数都包含无限多个符号。

原子要么是形如 $s = t$ 的等式，其中 s 和 t 是项；要么是形如 $F(t_1, \ldots, t_k)$ 的表达式，其中 $F \in \Sigma \cup \in$S，且 arity(F)= $= k$， $t_1, \ldots, t_k$ 是项。外延原子（分别地，内涵原子）是由 Σ （分别地，S）中的关系符号构成的原子。正文字就是原子，负文字是形如 $\neg\alpha$ 的表达式，其中 α 是正文字。我们用 α, β, γ 表示（正或负）文字。我们用 rel(α) 表示文字 α 中出现的关系符号，用 var(α) 表示 α 中出现的变量。事实是不含变量的外延原子。给定子集 $D \subseteq \subseteq$Dom，我们用 Facts(D) 表示仅包含 D 中常量的所有事实。实例是事实的有限集合；我们用 I、J 表示实例。

一个*Datalog* 规则是形如 r 的表达式

$$S(\bar{x}) \text{ :- } \alpha_1, \ldots, \alpha_n, \tag{1}$$

其中， $S(\bar{x})$ 是一个内涵原子， $\bar{x}$ 是 Var 上的一个元组，且 len($\bar{x}$) = arity(S)；每个 α_i 都是一个正文字，满足 rel(α_i) $\in \Sigma \cup$S，且 $\bar{x} \subseteq \bar{y} = \bigcup_{i \leq n}$ var(α_i)。我们将 $S(\bar{x})$ 称为 r 的头部，并将 α_1、.....、 α_n 称为 r 的体，记作 Body$_r(\bar{y})$。虽然我们不允许常量出现在规则的头部中，但可以通过在规则体中使用等式原子来模拟常量。我们用 free(r) 表示出现在 r 头部中的变量，并用 bound(r) 表示 r 体中未出现在 free(r) 中的变量。此外，令 var(r) = free(r) ∪ bound(r)。如果 t 是一个项组成的元组，且 len(y)−= len(t)，那么我们用 Body$_r(t)$ 表示将 y 中的变量替换为 t 中对应变量后得到的表达式。

Datalog 程序 P是由有限个规则 $r_1, \ldots, r_m$ 组成的序列，其中每个出现在某个规则 r_i 的体中的内涵关系符号，也必须出现在某个规则 r_j的头部中；每个 Datalog 程序 P 还指定了一个内涵关系符号 $\mathbf{ans}_P$，表示 P的答案谓词。程序 P的元数（记作 **元数**(P)）是其答案谓词的元数。我们还假设所有 Datalog 程序都是安全的，即每条规则中的每个变量都出现在该规则体的某个正文字中。

合取查询 是只包含一条规则且其体仅含外延原子的 Datalog 程序，而合取查询的并是由合取查询组成的 Datalog 程序

，其中每个查询的头部都相同。我们用 CQ、UCQ 和 Datalog 分别表示合取查询、合取查询的并以及 Datalog 程序的类。半正 Datalog 规则的形式与 Datalog 规则相同（见公式 1），但允许每个 α_i 是满足 rel(α_i) $\in \Sigma \cup$ S 的正文字、满足 rel(α_i) $\in \Sigma$ 的负文字，或形如 $s \neq t$ 的不等式原子，其中 $s, t \in$ Dom $\cup$ Var。我们用 CQ$\neg$、UCQ$\neg$ 和 sp-Datalog 表示在 CQ、UCQ 和 Datalog 中加入取反的外延原子与不等式原子后得到的扩展。通常，我们用 P 表示 Datalog 或 sp-Datalog 程序，用 Q 表示 UCQ$\neg$ 或其子类中的查询。我们用 adom(Q) 表示查询中出现的常量集合；对于数据库实例 I，我们还定义 adom(Q, I) $=$ adom(Q) $\cup$ adom(I)。

在第 4 节中，我们刻画了由关系代数运算定义的哪些 **合取查询的并集**$\neg$（即 **SPJU**$\neg$）子类能够在多项式时间内求解缺失答案问题。我们不引入关系代数的记号，而是考察基于规则的框架中的等价概念。如果对于半正Datalog 程序 P 中的每条规则 r，其体 r 中给定关系符号的出现次数不超过一次 Σ，则称该半正Datalog 程序为无自连接。如果对于半正 Datalog 程序 P 中的每条规则 r，都有无投影，则称该程序为无投影。如果半正 Datalog 程序不包含常量或等式原子的出现，且其原子均不包含重复变量，则称其为无选择。如果 r 的规则体仅包含一个原子，则称其为无连接。我们将**PJ** 记作**CQ**的无选择子类，将**SPU**$\neg$ 记作**合取查询的并集**$\neg$的无连接子类，并将**SJU**$\neg$ 记作**合取查询的并集**$\neg$的无投影子类。我们在第 4 节中研究这些类。

如引言中所述，对于查询类 $\mathcal{L}$ 而言，缺失答案修复问题可被视为可满足性问题的一般化，形式化定义如下。

▶**定义 2.1**。 设 L 表示任意查询类。***dbSAT***$^{\mathcal{L}}$ 问题是：给定一个查询 $Q \in L$，确定是否存在一个实例 **I** 使得 $Q(\mathbf{I}) \neq \emptyset$。

众所周知，对于关系演算而言，该问题是不可判定的 [53], ，因此对于关系代数和分层 Datalog 也同样如此。除可满足性之外，我们还将比较求值问题与缺失答案修复问题。

▶**定义 2.2**。 设 L 表示任意查询类。给定形如 $\langle Q, \mathbf{I}\ a \rangle$ 的输入，其中 $Q \in L$，**I** 是一个实例，且 $a \in \mathbf{Dom}$，L 的求值问题 记为 $\overline{\textbf{\textit{Eval}}^{\mathcal{L}}}$，其任务是确定 $a \in Q(\mathbf{I})$。

我们用$\mathbf{Eval}^{Q}$ 表示该问题的数据复杂度版本，其中查询Q 是固定的，输入的形式为$\langle \mathbf{I}\ a \rangle$。最后，为便于参考，我们定义如下查询的否定概念。

▶ 注释 2.3。在引言中，我们将查询的否定定义如下：给定查询 Q，令 $\neg Q$ （ Q 的否定 ）表示一个查询，它对每个数据库实例 **I** 返回 Q**(I)** 的补集（相对于**I**的活动域上所有具有适当元数的元组）。请注意，我们没有指定查询的否定属于哪个查询类；我们仅根据它所表示的数据库实例之间的映射来定义它。不过需要指出的是，如果 Q 可以用关系代数（**关系代数**）表示，那么$\neg Q$ 也可以用**关系代数**表示为 $D^{\mathbf{arity}}$(Q) 与 Q 的差集，其中 $D^{\mathbf{arity}}$(Q) 是**合取查询的并集**查询，它返回某个实例的活动域上所有长度为**元数**(Q)的元组组成的集合。

复杂度定义 我们假设读者熟悉常见的复杂度类**P类**、**NP**、**EXP**、**NEXP**、**2EXP**和**2NEXP**。在某个类前加上"F"，表示相应的函数问题类（例如**FP**、**FEXP**）。

▶ 定义 *2.4*。近似问题是一个三元组 $P = (I, S, cost)$，其中 I 是实例的集合，S 是将实例 x 映射x到可行解集合 $S(x)$ 的映射，而 $cost$ 是将数对 $(x, S(x))$ 映射到自然数的映射。实例 $x \in I$ 的最优代价为 $cost^*(x) := min^{y \in S(x)}\ cost(x, y)$。我们用 $opt(x)$ 表示满足 $cost(x) = cost^*(x)$ 的解的集合。

▶ 定义 *2.5*（严格归约，[46]）。考虑近似问题 $P_1 = (I_1, S_1, cost_1)$ 和 $P_2 = (I_2, S_2, cost_2)$。从 P_1 到 P_2 的严格归约是一对多项式时间可计算函数 (f, h)，其中 $f: I_1 \to I_2$，且 h 是将数对 (x, y) 映射到 $S_1(x)$ 中元素的映射；其中 $x \in I$ 且 $y \in S_2(f(x))$，满足

1. 若 $x \in I1$, $y \in S_2(f(x))$, 且 $y \in opt(f(x))$, 则 $h(x, y) \in opt(x)$；*2.* 对所有 $x \in I_1$ 和 $y \in S(f(x))$，都有 $cost(h(x, y)) \leq cost(y)$。

▶**定义 2.6.** 类 $\mathbf{FP}^{\mathbf{NP}}$ 是指可通过访问**NP** 预言机在多项式时间内计算的函数问题类。对于时间可构造函数 $l(n)$，我们将 $\mathbf{FP}^{\mathbf{NP}}[l(n)]$记作可通过向**NP** 预言机进行 $O(l(n))$ 次查询在多项式时间内计算的函数问题类，其中 n 是输入的大小。

▶ **定义 2.7** (**NP** 度量图灵机，[34])。**NP** 度量图灵机 M 是指一种非确定性多项式时间有界图灵机，其每个分支都会写出一个二进制数并接受或拒绝。这种机器对输入 $x \in \Sigma^*$ 的输出，是 M 在输入 x上所有接受分支中的最小值，记作 $opt^M(x)$。

▶ **定义 2.8** (优化多项式时间，[34])。函数 $f : \Sigma^* \to \mathbb{N}$ 属于 **OptP**，如果存在一台 **NP** 度量图灵机 M，使得对所有 $x \in \Sigma^*$，$f(x) = opt^M(x)$。此外，如果 $f \in \mathbf{OptP}$，且对所有 $x \in \Sigma^*$，$f(x)$ 的二进制编码有界于$l(|x|)$，则 f 属于 $\mathbf{OptP}[l(n)]$。

▶ 定义 *2.9*（度量归约，[34]）。设 $P_1, P_2 : \Sigma^* \to \mathbb{N}$。从$P_1$ 到 P_2 的度量归约是一对多项式时间可计算函数 (f, g)，其中 $f : \Sigma^* \to \Sigma^*$ 且$g : \Sigma^* \times \mathbb{N} \to \mathbb{N}$，满足对所有 $x \in \Sigma^*$，$P_1(x) = g(x, P_2(f(x)))$。若映射 $k \mapsto g(x, k)$ 是线性的，则度量归约 (f, g) 是线性的；若 $g(x, k) = k$，则该归约是精确的。

3 将缺失答案问题形式化

为了解决寻找缺失答案的极小修复这一问题，我们还研究以下问题：确定是否存在修复、确定是否存在大小不超过给定大小界的修复，以及确定极小修复的大小。因此，我们将修复定义为使所需元组出现在查询答案中的更新，而不考虑更新的大小。

▶ 定义 *3.1*（更新）。数据库实例 I 的更新是一个二元组 $U = (Ins, Del)$，其中 $Ins \subseteq Facts(Dom) \setminus I$，$Del \subseteq I$，且二者均为有限事实集合，分别表示要插入到 I 中和从 I 中删除的事实。给定 I 的更新 U，我们定义 $I_U = (I \cup Ins) \setminus Del$。更新 U 的大小为$|U| = |Ins \cup Del|$。如果 $Ins \cup Del \subseteq Facts(D)$，则称 U 是关于 $D \subseteq Dom$ 的更新。如果更新 U 满足 $a \in Q(I_U)$，则称 U 是关于三元组 $\langle Q, I, a \rangle$ 的修复。

换言之，更新的大小是其中包含的插入和删除的数量。注意，$|\mathbf{U}| = |\mathbf{I} \oplus \mathbf{I}_\mathbf{U}|$；也就是说，极小修复是指使

输入实例$\mathbf{I}$ 与更新后的实例$\mathbf{I}_\mathrm{U}$之间的对称差最小的修复。这种极小性度量类似于基于基数的修复语义 [3, 36]下的一致查询回答中的修复概念。

▶定义 *3.2*（缺失答案问题）。设 L 表示任意一个查询类。我们定义四个计算问题。*1.* $MA^{\mathcal{L}}_{dec}$（缺失答案判定问题）：给定 $\langle Q, I\ a\rangle$，判断是否存在一个相对于 $\langle Q, I\ a\rangle$ 的修复 U；*2.* $MA^{\mathcal{L}}_{size}$（有界缺失答案修复问题）：给定 $\langle Q, I\ a, k\rangle$，判断是否存在一个相对于 $\langle Q, I\ a\rangle$ 的修复 U，使得 $|Rep| \leq k$；*3.* $MA^{\mathcal{L}}_{size}$（缺失答案修复规模问题）：给定输入 $\langle Q, I\ a\rangle$，计算相对于 $\langle Q, I\ a\rangle$ 的修复 U 的最小规模；若修复不存在，则予以说明；*4.* $MA^{\mathcal{L}}_{min}$（缺失答案修复问题）：给定 $\langle Q, I\ a\rangle$，计算相对于 $\langle Q, I\ a\rangle$ 的最小基数修复 U；若修复不存在，则予以说明；其中 $Q \in L$，I 是一个实例，a 是 Dom 上的一个元组，其 $a =$ 元数为(Q)，且 $k \in \mathbb{N}$。

注意，上述定义允许输入中出现 L 类的任意查询——这属于联合复杂性设置，将在第 4 节中讨论。我们还将研究 MA^{Q}_{bound}，即有界缺失答案修复问题的一个变体：查询 Q 是固定的，输入形式为 $\mathrm{I}\ a$、k，目标是确定是否存在一个相对于 Q、$\mathrm{I}\ a$ 的修复 U，使得 $|\mathrm{U}| \leq k$。类似地，我们研究 MA^{Q}_{size} 和 MA^{Q}_{min}，即缺失答案修复大小问题和缺失答案修复问题；在这两个问题中，查询 Q 是固定的，输入形式为 $\mathrm{I}\ a$，目标是计算一个相对于 Q、$\mathrm{I}\ a$ 且具有最小基数的修复 U。MA^{Q}_{bound}、MA^{Q}_{size} 和 MA^{Q}_{min} 问题是数据复杂度版本，也是第 5 节的重点。在数据复杂度和联合复杂性设置中，我们都将外延数据库谓词的模式 Σ 视为固定的。

初步复杂度观察。在转向我们的主要结果之前，我们先进行一些初步观察。

▶**命题 3** *.3.* 设 L 是一个查询类 *. Then*
1. $\boldsymbol{Eval^{\mathcal{L}}} \leq_{\mathbf{P}} \boldsymbol{MA^{\mathcal{L}}_{bound}}$；并且*2.* $\boldsymbol{Eval^{\mathcal{L}}}$ 可在一次调用 $\boldsymbol{MA^{\mathcal{L}}_{size}}$（或 $\boldsymbol{MA^{\mathcal{L}}_{min}}$）预言机的情况下在线性时间内判定。

证明。第一项结论由归约 $Q, \mathrm{I}\ a \mapsto Q, \mathrm{I}\ a, 0$ 得出。第二项结论源于以下观察：$\mathrm{MA}^{\mathcal{L}}_{size}(Q, \mathrm{I}\ a) = 0$ 当且仅当 $\mathrm{MA}^{\mathcal{L}}_{min}(Q, \mathrm{I}\ a) = (\emptyset, \emptyset)$ 当且仅当 $\mathrm{Eval}^{\mathcal{L}}(Q, \mathrm{I}\ a)$ 返回真。这里只要求线性运行时间，是为了将输入复制到预言机磁带、调用预言机并返回答案。◀

请注意，命题 3.3 中的关系对于这些问题的数据复杂度版本也成立。由于 $\mathbf{Eval^{CQ}}$ 是 **NP**-完全的，[11]，根据命题 3.3 可立即得出，$\mathbf{MA^{CQ}_{bound}}$ 是 **NP**-困难的，并且$\mathbf{MA^{CQ}_{min}}$ 无法在多项式时间内求解（除非$\mathbf{P} = \mathbf{NP}$）。我们在第 4 节中显著地细化了这些结论。命题 3.3 还意味着 $\mathbf{MA^{Datalog}_{bound}}$ 是 **EXP**-困难的，因为 $\mathbf{Eval^{Datalog}}$ 是 **EXP**-完全的 [17]（此结论隐含于 [55, 30]）。

现在我们来考察缺失答案问题不同版本之间的一些基本关系。显然，$\mathbf{MA}^{\mathcal{L}}_{min}$ 问题的难度至少与$\mathbf{MA}^{\mathcal{L}}_{dec}$、$\mathbf{MA}^{\mathcal{L}}_{bound}$和$\mathbf{MA}^{\mathcal{L}}_{size}$相同；类似地，$\mathbf{MA}^{Q}_{min}$ 的难度至少与$\mathbf{MA}^{Q}_{size}$ 和 $\mathbf{MA}^{Q}_{bound}$相同。此外，$\mathbf{MA}^{\mathcal{L}}_{bound}$ 是$\mathbf{MA}^{\mathcal{L}}_{min}$ 优化问题的自然“判定版本”。$\mathbf{MA}^{\mathcal{L}}_{bound}$与 $\mathbf{MA}^{\mathcal{L}}_{dec}$之间没有直接关系，因为对于给定的三元组$\langle Q, \mathbf{I}\ a\rangle$，

可能存在修复，但不存在大小为 k 的修复。不过，对于具有以下性质的查询类，可以确定一种直接的关系。

▶**定义 3.4**（可计算有界修复）。 如果存在一个可计算函数 $\beta_{\mathcal{L}}$，它以形如 $\langle Q\mathbf{I}\ a\rangle$ 的元组作为输入，并返回一个有限集合 $S \subseteq \mathbf{Dom}$，使得只要相对于 $\langle Q、\mathbf{I}\ a\rangle$ 存在一个修复 U，则相对于 $\langle Q、\mathbf{I}_a\rangle$ 在 $\bar{S}$ 上也存在一个修复 U′，我们就称查询类 L 具有可计算有界修复。

定义 3.4 的直观含义体现在其名称中：对于具有可计算有界修复的查询类 $\mathcal{L}$，存在一个可计算函数，能够针对该查询类的任意缺失答案输入，输出一个有限的修复搜索空间。类似于定义 3.4 的概念将在第 5 节中出现，用于分析**sp-Datalog** 查询的$\mathbf{MA}^{Q}_{min}$ 问题的数据复杂度 Q （参见定义 5.1）。根据定义 3.4，可以直接验证以下命题。

▶**命题 3.5**。 如果 L 是一个允许可计算有界修复的查询类，且$\boldsymbol{Eval}^{\mathcal{L}}$ 是可判定的，那么映射 $\langle Q,\mathbf{I}\ a\rangle \to \langle Q,\mathbf{I}\ a, p_{\Sigma}(|\beta_{\mathcal{L}}(Q,\mathbf{I}\ a)|)\rangle$ 是从$\boldsymbol{MA}^{\mathcal{L}}_{dec}$ 到$\boldsymbol{MA}^{\mathcal{L}}_{bound}$的可计算归约，其中 $p_{\Sigma}(n) = \sum_{R\in\Sigma} n^{\mathbf{arity}(R)}$。

重要的是，尽管对于查询类$\mathcal{L}$ 而言，$\mathbf{MA}^{\mathcal{L}}_{bound}$ 问题似乎具有**NP** 问题的特征，但一般而言并非如此。例如，**Datalog** 求值的联合复杂性是**EXP**完全的 [17, 55, 30]，，因此 $\mathbf{MA}^{\mathcal{L}}_{bound}$ 不属于**NP** （除非$\mathbf{NP} = \mathbf{EXP}$），因为根据命题 3.3，$\mathbf{Eval}^{\mathbf{Datalog}} \leq_P \mathbf{MA}^{\mathbf{Datalog}}_{bound}$。然而，由于$\mathbf{MA}^{\mathcal{L}}_{bound}$ 的输入包括整数 k，该整数界定了见证修复中插入和删除的总次数，因此只需考虑相对于输入规模至多为指数级 的修复即可。这使我们能够给出一个一般性的上界；该上界只要求所考虑的查询类的求值问题属于**NEXP**，并且该类仅包含如下文所定义的泛型 查询。

▶ 定义 *3.6*（泛型查询）。令 $\rho : Dom \to Dom$ 为任意双射。给定一个实例 *I*，我们用 $\rho(I)$ 表示将 *I* 中每个事实里的每个常量替换为其在 ρ 下的像后得到的实例。我们对更新 *U* 类似地定义 $\rho(U)$。若对于任意数据库 *I* 以及任意逐点固定 D 的双射映射$\rho : Dom \to Dom$（即，对于所有 $d \in D$，均有 $\rho(d) = d$），都有 $\rho(Q(I)) = Q(\rho(I))$，则称查询 Q 对子集 $D \subseteq$*Dom* 是 *D*-泛型的。若查询 Q 是 *adom(Q)*-泛型的，则称其为泛型查询。

换句话说，如果查询的答案不受查询活动域中未明确出现的常量重命名的影响，那么该查询就是泛型的。现在我们可以陈述并证明 $\mathbf{MA}^{\mathcal{L}}_{bound}$ 的一般上界。

▶**命题 3.7**。 **</style}设 L 为一类泛型查询。若 Eval**$^{\mathcal{L}}$ 属于 **NEXP**，则 $\boldsymbol{MA}^{\mathcal{L}}_{bound}$ 属于**2NEXP**。

证明（概要）。根据泛型性，对于大小为 N 的输入 $Q, \mathrm{I}\ a, k$，只需考虑在 $a \cup \mathrm{adom}(Q, \mathrm{I}) \cup \bar{C}$ 上的修复，其中 C 是大小为 k 的新常量域。算法如下进行：非确定性地生成一个基数至多为 k 的更新 U，然后非确定性地为实例 $\mathrm{I_U}$ 生成 $\mathrm{Eval}^{\mathcal{L}}$ 的一个可能见证。算法最后计算 $\mathrm{Eval}^{\mathcal{L}}(\mathrm{I_U})$ 并返回答案。该算法显然是正确的。此外，注意到，由于 k 采用二进制表示，实例 $\mathrm{I_U}$ 的规模可能比输入的编码大一个指数级数量级。又因为 $\mathrm{Eval}^{\mathcal{L}}$ 的见证可能比 $\mathrm{I_U}$ 大一个指数级数量级，所以该算法属于 2NEXP。◀

由于 $\mathbf{Eval}^{\mathbf{sp\text{-}Datalog}}$ 属于 **EXP**，且 $\mathbf{Eval}^{\mathbf{RA}}$ 属于 **PSPACE** ([1])，上述结果表明，$\mathbf{MA}^{\mathbf{Datalog}}_{bound}$、$\mathbf{MA}^{\mathbf{sp\text{-}Datalog}}_{bound}$ 和 $\mathbf{MA}^{\mathbf{RA}}_{bound}$ 均属于 **2NEXP**。注意，即使假设 $\mathbf{Eval}^{\mathcal{L}}$ 属于 **EXP**，命题 3.7 的论证也能为 $\mathbf{MA}^{\mathcal{L}}_{bound}$ 的复杂性给出 **2NEXP** 上界；我们针对 $\mathbf{Eval}^{\mathcal{L}}$ 属于 **NEXP** 这一假设陈述该命题，是为了使其尽可能具有一般性。在第 4 节中，我们通过为 $\mathbf{MA}^{\mathbf{Datalog}}_{min}$（分别为 $\mathbf{MA}^{\mathbf{sp\text{-}Datalog}}_{min}$）给出一个 **FEXP**（分别为 **F2EXP**）算法，将 $\mathbf{MA}^{\mathbf{Datalog}}_{bound}$（分别为 $\mathbf{MA}^{\mathbf{sp\text{-}Datalog}}_{bound}$）的上界进一步降低至 **EXP**（分别为 **2EXP**）。

虽然$\mathbf{MA}^{\mathcal{L}}_{bound}$并不总是一个**NP**问题，但我们可以给出一个非常简单的通用判据，用于判断 $\mathbf{MA}^{\mathcal{L}}_{bound}$ 是否属于**NP**。如果一个查询类$\mathcal{L}$ 具有多项式有界修复，即$\mathcal{L}$ 具有可计算有界修复，并且 $\beta_{\mathcal{L}}$ 可在多项式时间内计算，则称其满足该条件。于是，采用与命题 3.7 的证明类似的算法，我们得到以下命题。

▶**命题 3.8**。 </style}设 **L** 是一个泛型查询类。**如果 L 具有多项式有界修复，且$\mathbf{Eval}^{\mathcal{L}}$** 属于 **NP**，那么$MA^{\mathcal{L}}_{bound}$ 属于**NP**。

回顾一下，如果存在一个从实例的域到 Q定义中的某个合取查询的满足赋值，则查询 Q 属于**合取查询的并集**¬ 在实例中得到满足。令 m 表示 Q 定义中变量数最多的合取查询所包含的变量数。对于 Q，满足赋值的值域大小至多为m，因此只需考虑在域 $a \cup \mathbf{adom}(Q,\mathbf{I}) \cup C$上的修复，其中 C 是一个由 m 新常量组成的集合。由此我们得出，**合取查询的并集**¬ 具有多项式有界修复。此外，已知$\mathbf{Eval}^{\mathbf{UCQ}\neg}$ 属于**NP** [1]。因此**合取查询的并集**¬ 满足命题 3.8 的假设，所以$\mathbf{MA}^{\mathbf{UCQ}\neg}_{bound}$问题（以及$\mathbf{MA}^{\mathcal{L}}_{bound}$ 对于$\mathcal{L}$ 的**合取查询的并集**的所有句法片段）都属于**NP**。

4 联合复杂性

现在我们证明，对于许多查询类 L，$\mathrm{MA}^{\mathcal{L}}_{dec}$（见定义 3.2）和 $\mathrm{dbSAT}^{\mathcal{L}}$（见定义 2.1）问题可以相互多项式时间归约，并由此推导出若干复杂度结果作为推论。随后我们证明，$\mathrm{MA}^{\mathbf{UCQ}\neg[\ ]}_{min}$ sjf、$\mathrm{MA}^{\mathbf{SJU}\neg}_{min}$和 $\mathrm{MA}^{\mathbf{SPU}\neg}_{min}$可以在多项式时间内求解。由此还可得出，$\mathrm{MA}^{\mathbf{UCQ}\neg[\ ]}_{bound}$ sjf、$\mathrm{MA}^{\mathbf{SJU}\neg}_{bound}$和 $\mathrm{MA}^{\mathbf{SPU}\neg}_{bound}$ 属于 P类。回顾一下，SJU¬ 和 SPU¬分别是合取查询的并集 UCQ¬的无投影片段和无连接片段。接着，我们给出一个严格多项式时间归约，该归约同时也是一个度量归约（归约的定义见第2节），并据此证明 $\mathrm{MA}^{\mathbf{PJ}}_{bound}$ 是 NP 完全的，而 $\mathrm{MA}^{\mathbf{PJ}}_{size}$ 是 OptP[log(n)]-困难的。最后，我们证明存在一个用于 $\mathrm{MA}^{\mathbf{PJ}}_{size}$的 OptP[log($n$)] 算法、一个用于 $\mathrm{MA}^{\mathbf{UCQ}\neg}_{min}$的 $\mathrm{FP}^{\mathbf{NP}}$ 算法，以及一个用于 $\mathrm{MA}^{\mathcal{L}}_{min}$的 FEXP 算法。

缺失答案判定问题。 对于涉及$\mathbf{MA}^{\mathcal{L}}_{dec}$ 和$\mathbf{dbSAT}^{\mathcal{L}}$的结果，我们只需假设查询类$\mathcal{L}$ 在弱形式的选择和投影下封闭。若存在一个多项式时间可计算映射，对于给定的一个 n元查询 $Q \in \mathcal{L}$ 和一个长度为n的元组 a，该映射可以构造出一个查询 $Q' \in \mathcal{L}$，使得对于所有实例 $\mathbf{I}$，$Q'(\mathbf{I}) \neq \emptyset$ 当且仅当$a \in Q(\mathbf{I})$，则称查询类$\mathcal{L}$ 在（弱）选择下封闭。若存在一个多项式时间可计算映射，对于给定的 $Q \in \mathcal{L}$，该映射可以构造出一个查询 $Q' \in \mathcal{L}$，使得对于所有实例$\mathbf{I}$，$Q'(\mathbf{I}) = \top$ 当且仅当 $Q(\mathbf{I}) \neq \emptyset$，则称查询类$\mathcal{L}$ 在（弱）投影下封闭。

▶ **定理 4.1**。 如果 L 在（弱）选择下封闭，则$MA^{\mathcal{L}}_{dec} \leq_{\mathbf{P}} dbSAT^{\mathcal{L}}$。

证明。 给定输入 $\langle Q、\mathbf{I}\ a\rangle$ 到 $\mathbf{MA}^{\mathcal{L}}_{dec}$，令 Q' 为 $\mathcal{L}$ 中由关于弱选择的闭包性保证存在的查询。如果 $\langle Q、\mathbf{I}\ a\rangle \in \mathbf{MA}^{\mathcal{L}}_{dec}$，则 $Q'(\mathbf{I}_U) \neq \emptyset$，因此 $Q' \in \mathbf{dbSAT}^{\mathcal{L}}$。

反之，如果 $Q' \in \mathbf{dbSAT}^{\mathcal{L}}$，则存在一个实例 $\mathbf{J}$，使得 $Q'(\mathbf{J}) = \top$，因此$a \in Q(\mathbf{J})$。于是，对于修复 $\mathrm{U} = (\mathrm{Ins}、\mathrm{Del})$，其中 $\mathrm{Ins} = \mathbf{J} \setminus \mathbf{I}$，$\mathrm{Del} = \mathbf{I} \setminus \mathbf{J}$，我们有$\mathbf{I}_{\mathrm{U}} = \mathbf{J}$。因此 $a \in Q(\mathbf{I}_{\mathrm{U}})$，从而得出结论：$\langle Q、\ \mathbf{I}\ a\rangle \in \mathbf{MA}^{\mathcal{L}}_{dec}$。◀

下一个定理的证明与定理 4.1 的证明类似。

▶ **定理 4.2**。 如果 L 在（弱）投影下封闭，那么 $\boldsymbol{dbSAT}^{\mathcal{L}} \leq_{\mathbf{P}} \boldsymbol{MA}^{\mathcal{L}}_{dec}$。

证明。给定一个输入 $Q \in \mathrm{L}$，令 $Q', \mathrm{E}\ \epsilon$ 为 $\mathrm{MA}^{\mathcal{L}}_{dec}$ 输入，其中 Q' 是由弱投影下的封闭性所保证的 L 中的查询，E 是空实例，而 ϵ 是空元组。如果 $Q \in \mathrm{dbSAT}^{\mathcal{L}}$，那么对于某个 J，有 $Q(\mathrm{J}) \neq \emptyset$，因此 $Q'(\mathrm{J}) = \top$。于是，修复 $\mathrm{U} = (\mathrm{Ins}, \mathrm{Del})$（其中 $\mathrm{Ins} = \mathrm{J}$ 且 $\mathrm{Del} = \emptyset$）产生了修复后的实例 $\mathrm{I_U} = \mathrm{J}$。因此 $Q', \mathrm{E}\ \epsilon \in \mathrm{MA}^{\mathcal{L}}_{dec}$。反之，如果 $Q', \mathrm{E}\ \epsilon \in \mathrm{MA}^{\mathcal{L}}_{dec}$，那么存在一个修复 U，使得$Q'(\mathrm{E_U}) = \top$。因此 $Q(\mathrm{E_U}) \neq \emptyset$，从而 $Q \in \mathrm{dbSAT}^{\mathcal{L}}$。◀

第2节中提到的所有查询类都满足弱选择和投影性质，而且这些映射事实上可以在与查询大小成线性关系的时间内计算。

▶**推论 4.3**。 以下陈述成立。
1. $\boldsymbol{MA}^{\mathbf{UCQ}^{\neg}}_{dec}$可在 **P类** 中判定；2. $\boldsymbol{MA}^{\mathbf{Datalog}}_{dec}$ 和 $\boldsymbol{MA}^{\mathbf{sp\text{-}Datalog}}_{dec}$ 是 **EXP**-完全的；3. $\boldsymbol{MA}^{\mathbf{RA}}_{dec}$ 不可判定。

证明。 根据定理 4.1 和 4.2，以及关于这些类的可满足性问题复杂度的已知结果 [1, 51, 35, 53]。值得注意的是，据作者所知，**EXP**-困难性$\mathbf{dbSAT}^{\mathbf{Datalog}}$ 是一个未发表的民间结果，因此我们在附录中给出这一复杂度下界的证明。◀

由于$\mathbf{MA}^{\mathcal{L}}_{dec}$ 可以借助针对$\mathbf{MA}^{\mathcal{L}}_{size}$ 或$\mathbf{MA}^{\mathcal{L}}_{min}$的预言机求解，推论 4.3 意味着$\mathbf{MA}^{\mathbf{Datalog}}_{size}$、$\mathbf{MA}^{\mathbf{sp\text{-}Datalog}}_{size}$、$\mathbf{MA}^{\mathbf{Datalog}}_{min}$ 和$\mathbf{MA}^{\mathbf{sp\text{-}Datalog}}_{min}$ 都是**EXP**-困难的，而$\mathbf{MA}^{\mathbf{RA}}_{size}$和$\mathbf{MA}^{\mathbf{RA}}_{min}$ 则不可计算。

（有界）缺失答案修复问题。 现在我们转而讨论$\mathbf{MA}^{\mathcal{L}}_{bound}$ 和$\mathbf{MA}^{\mathcal{L}}_{min}$ 问题在合取查询的并集 $\mathbf{UCQ}^{\neg}$的子类上的复杂度。对于$\mathbf{MA}^{\mathcal{L}}_{bound}$的多项式时间情形，显然只需证明$\mathbf{MA}^{\mathcal{L}}_{min}$ 可以在多项式时间内求解。

▶**定理 4.4**。 $\boldsymbol{MA}^{\mathbf{UCQ}^{\neg}[\ ]_{\mathbf{sjf}}}_{min}$、$\boldsymbol{MA}^{\mathbf{SJU}^{\neg}}_{min}$和$\boldsymbol{MA}^{\mathbf{SPU}^{\neg}}_{min}$问题均属于**FP**。

证明。 要为查询 $Q \in \mathbf{UCQ}$ 找到一个极小修复，其中该查询是合取查询 $Q_1, \ldots, Q_m$的并，只需分别为每个 Q_i 找到极小修复 U_i，并返回其中最小的一个。因此，我们只需证明，$\mathbf{CQ}^{\neg}[\mathbf{sjf}]$, $\mathbf{SJ}^{\neg}$和$\mathbf{SP}^{\neg}$查询的极小修复可以在多项式时间内找到。对于第一种情况，注意到对于无自连接合取查询，其体至多包含 $|\Sigma|$ 个文字。由于模式固定，我们可以构造三元组$\langle Q,\mathbf{I}\ a\rangle$ 的一个修复，只需在域 $a \cup \mathbf{adom}(Q, \mathbf{I}) \cup \{c\}$上进行常数次插入或删除，其中 c 是一个新常量。此外，$\mathbf{Eval}^{\mathbf{CQ}^{\neg}[\mathbf{sjf}]}$ 属于多项式时间，因此我们可以通过对所有可能的修复进行蛮力搜索，在多项式时间内求解$\mathbf{MA}^{\mathbf{CQ}^{\neg}[\ ]_{\mathbf{sjf}}}_{min}$。

现在假设给定了一个输入 $\langle Q,\mathbf{I}\ a\rangle$，用于 $\mathbf{MA}^{\mathbf{SJ}^{\neg}}_{min}$。回忆一下，$Q$ 由形式为 $S(x) :- \alpha_1, \ldots, \alpha_n$ 的规则 r 定义，其中每个 α_i 都是正文字或负文字，并且$\{x\} = \bigcup_{i \leq n} \mathbf{var}(\alpha_i)$。因此，$\mathbf{Body}_r(a)$ 是一个不含变量的表达式，所以对于

任意实例 **J**， $a \in Q$**(J)** 成立，当且仅当对每个 $i \leq n$， $\alpha_i[g] \in$**J** 成立，其中 g 是满足 $g(x) = a$ 的赋值。因此，极小修复由 U $=$ (Ins,Del) 给出，其中

$$\mathsf{Ins} = \{R(\bar{a}) \mid \alpha_i = R(\bar{x}) \text{ and } R(\bar{a}) \notin \mathbf{I} \text{ for some } i \leq n\}$$
$$\mathsf{Del} = \{R(\bar{a}) \mid \alpha_i = \neg R(\bar{x}) \text{ and } R(\bar{a}) \in \mathbf{I} \text{ for some } i \leq n\},$$

这可以在多项式时间内轻松计算。

现在假设 $\langle Q,$**I** $a\rangle$ 是 $\mathbf{MA}^{\mathbf{SP}^{\neg}}_{min}$的一个输入。回想一下， Q 由形如 $S'(x) :- \beta$的规则定义，其中 β 是一个正文字或负文字，并且可能包含元组 x中未出现的变量。显然，要么 $a \in Q$**(I)**成立，要么极小修复恰好包含一次插入（如果 β 是正文字）或恰好包含一次删除（如果 β 是负文字）。无论哪种情况，只需遍历数据库中相应关系的所有事实，即可确定需要哪种情况并计算出一个极小修复。由于这显然是一个多项式时间过程，证明完毕。◀

▶**定理 4.5**。 $MA^{\mathbf{UCQ}^{\neg}}_{size}$属于 $\mathbf{OptP}[O(\log(n))]$ ，且 $MA^{\mathbf{UCQ}^{\neg}}_{min}$属于 $\mathbf{FP}^{\mathbf{NP}}$。

证明。 我们知道，**合取查询的并集**$^{\neg}$ 具有多项式有界修复（见第 3 节）。因此，我们可以通过以下方式使用 $\mathbf{MA}^{\mathbf{UCQ}}_{size}$ 和一台 **NP** 度量图灵机求解。给定一个输入$\langle Q,$**I** $a\rangle$，在计算的每个分支上，我们非确定性地生成一个可能的更新，再非确定性地生成一个可能的满足赋值，然后检查该满足赋值是否能证明相对于 $\langle Q,$**I** $a\rangle$的更新是一个修复。若是，则以二进制写下该修复的大小并接受；否则，在该分支上拒绝。该算法显然正确，并且每个分支都在多项式时间内运行。

为了证明 $\mathrm{MA}^{\mathbf{UCQ}^{\neg}}_{min}$属于 $\mathrm{FP}^{\mathbf{NP}[O(n^2)]}$,，令 $Q,$**I** a 为一个输入，其中 Q 属于 $\mathrm{UCQ}^{\neg}$。我们首先通过二分搜索，使用针对 $\mathrm{MA}^{\mathbf{UCQ}^{\neg}}_{bound}$的 NP 查询，在区间 $[0, m]$上确定最优修复的大小 K （如果存在），其中 m 是 Q 的规则中所含文字数最多者的文字数。然后，我们按如下方式构造一个最优修复。首先，令 C 表示一个常量集合，该集合中的常量不同于 $a \cup$adom(I)，其大小等于 Q中出现的变量数；并将部分赋值初始化为 $g = \{x_i, a_i \mid i \leq \mathsf{len}(a)\}$。对于 Q 中的每个变量 x 以及 $a \cup \mathsf{adom}(D) \cup C$ 中的每个元素 b，我们向预言机查询以下问题：“是否存在一个相对于 $Q,$**I** a 的大小为 K 的修复，使得 $a \in Q$(I) 中的事实由一个扩展 $g \cup \{x, b\}$ 的赋值见证？”不难看出，该查询属于 NP。此外，我们只需向该预言机提出 $O(m(|a| + |\mathsf{adom(I)}| + m))$ 次查询，这相对于输入大小是二次的。因此，$\mathrm{MA}^{\mathbf{UCQ}^{\neg}}_{min}$属于 $\mathrm{FP}^{\mathbf{NP}[O(n^2)]}$。◀

我们现在通过将最小集合覆盖（**MIN-SC**）问题严格归约和度量归约，分别给出 $\mathbf{OptP}[\log(n)]$ 和$\mathbf{FP}^{\mathbf{NP}[\log(n)]}$ 下界，适用于$\mathbf{MA}^{\mathbf{PJ}}_{bound}$ 和$\mathbf{MA}^{\mathbf{PJ}}_{min}$。严格归约和度量归约的定义见第2节。请注意，我们将$\mathbf{MA}^{\mathcal{L}}_{min}$ 视为一种近似算法：将关于$\langle Q,$**I** $a\rangle$的修复 U 视为实例$\langle Q,$**I** $a\rangle$的可行解，并将修复的大小 |U| 视为其代价。已知，如果存在一种**MIN-SC** 近似算法，其近似比为 $(1-\alpha)\ln(n)$，对任意 $\alpha > 0$均成立，其中 n 是实例的大小，那么$\mathbf{P} = \mathbf{NP}$ [19]。

▶**定理4.6**。 如果（*EDB*）模式 Σ 包含一个一元关系符号和一个二元关系符号，那么**1.** 该 $MA^{\mathbf{PJ}}_{bound}$ 问题是**NP**完全的，并且**2.** 该 $MA^{\mathbf{PJ}}_{size}$ 问题是$\mathbf{OptP}[\log(n)]$完全的。

[3] **MIN-SC** 问题是：给定一个集合族 $X = \{s_1, \ldots\ldots, s_m\}$，其中 $U = \bigcup_{i \leq m} s_i$，求一个集合 $C \subseteq X$，使得对于每个 $u \in U$，都存在 $s \in C$，满足 $u \in s$。

3. 该$MA^{\mathbf{PJ}}_{min}$ 问题是$\mathbf{FP}^{\mathbf{NP}[\log(n)]}$-困难的。**4.** 该$MA^{\mathbf{PJ}}_{min}$ 问题是$MIN\text{-}SC$-在严格归约下难以近似。

证明。NP困难性结果可由 $\mathrm{Eval}^{\mathbf{PJ}}$ 的NP困难性以及命题 3.3 中给出的归约得出。对于近似难度结果，我们给出从 MIN-SC 问题到 $\mathrm{MA}^{\mathbf{PJ}}_{min}$ 问题的严格归约 (f, h)。该归约也是从 MIN-SC 问题到 $\mathrm{MA}^{\mathbf{PJ}}_{min}$ 问题的度量归约，由此可得 $\mathrm{MA}^{\mathbf{PJ}}_{size}$ 问题是 OptP[log(n)]-困难的。此外，根据 [34, 定理 3.3]，$\mathrm{MA}^{\mathbf{PJ}}_{min}$的 $\mathrm{FP}^{\mathbf{NP}[\log(n)]}$-困难性可由 $\mathrm{MA}^{\mathbf{PJ}}_{size}$ 的 OptP[log(n)]-困难性得出。

令 $X = \{s_1, \ldots, s_m\}$（其中 $U = \bigcup_{i \le m} s_i = \{u_1, \ldots, u_n\}$）表示**MIN-SC**的任意输入。设 P 是 Σ 中的一个一元关系符号，F 是Σ中的一个二元关系符号，并固定互不相同的常量 $a = a_1, \ldots, a_n$ 和 $b = b_1, \ldots, b_m$。现在定义输入$f(X) = \langle Q^X, \mathbf{I}^X, a_1, \ldots, a_n\rangle$ 作为 $\mathbf{MA}^{\mathbf{PJ}}_{min}$的输入，定义如下。令 Q^X 表示由以下规则定义的$\mathbf{PJ}$查询

$$\mathbf{ans}(x_1, \ldots, x_n) \text{ :- } F(y_1, x_1), \ldots, F(y_n, x_n), P(y_1), \ldots, P(y_n),$$

其中 x_1、……、x_n、y_1、……、y_n 是两两不同的变量。令 $\mathbf{I}^X = \{F(b_j, a_i) \mid u_i \in s_j\}$。其思想是，$a_i$ 常量表示 U 的元素，b_i 常量表示 S 的元素，而事实 $F(a, b)$ 断言 $a \in b$。一元 P 谓词用于“选择”要纳入集合覆盖的集合。现在我们将这一想法形式化。

我们声称 $\mathrm{MA}^{\mathbf{PJ}}_{min}(f(X)) = k$。为证明 $\mathrm{MA}^{\mathbf{PJ}}_{min}(f(X)) \le k$，设 C 是 U 的一个大小为 k 的集合覆盖。于是，对所有 $i \le n$，根据 $\mathbf{I}^X$ 的构造，我们有存在某个 $s_j \in C$ 使得 $F(b_j, a_i) \in \mathbf{I}^X$。容易推出，对于修复 $\mathbf{U} = (\{P(b_j) \mid s_j \in C\}, \emptyset)$，有 $a \in Q(\mathbf{I}^X_{\mathbf{U}})$。因此 $\mathrm{MA}^{\mathbf{PJ}}_{min}(f(X)) \le k$。为证明 $\mathrm{MA}^{\mathbf{PJ}}_{min}(f(X)) \ge k$，我们使用以下断言。

▷ 命题 4.7。如果 $\mathbf{U} = (\mathrm{Ins}, \mathrm{Del})$ 是一个满足 $a \in Q'(\mathbf{I}^X_{\mathbf{U}})$ 且基数最小的修复，则 $\mathrm{Del} = \emptyset$。此外，如果 Ins 包含一个事实 $F(c, d)$，其中 $c, d \in \mathbf{Dom}$ 为某些常量，则存在某个 $s_i \in S$ 使得 $a \in Q(\mathbf{I}^X_{\mathbf{U}})$，其中 $\mathbf{U}' = (\mathrm{Ins}', \mathrm{Del})$ 的 $\mathrm{Ins}' = (\mathrm{Ins} \setminus \{F(c, d)\}) \cup \{P(b_i)\}$。

命题证明。设 $\mathbf{U} = (\mathrm{Ins}, \mathrm{Del})$ 如命题所述。$\mathrm{Del} = \emptyset$ 是由 Q 的单调性直接得到的。由于 $a \in Q(\mathbf{I}^X_{\mathbf{U}})$，对于每个 $i \le n$，必定存在 $e \in \mathrm{Dom}$，使得 $F(e, a_i)\ P(e) \in \mathbf{I}^X$。现在注意到，对于某个 $i \le n$，$d = a_i$，因为否则我们会得到 $a \in Q(\mathbf{I}^X_{\mathbf{U}''})$，其中 $\mathbf{U}'' = (\mathrm{Ins} \setminus \{F(c, d)\}, \emptyset)$，这与 U 的极小性矛盾。设 $s_j \in S$ 为任意集合，使得 $u_i \in s_j$（根据假设，该集合必定存在，因为$\bigcup S = U$），并令 $\mathbf{U}' = (\mathbf{U} \setminus \{F(c, a_i)\}) \cup \{P(b_j)\}$。由于 $\mathrm{Del} = \emptyset$，我们有 $F(b_j, a_i)\ P(b_j) \in \mathbf{I}^x_{\mathbf{U}'}$，因此 $a \in Q(\mathbf{I}^x_{\mathbf{U}'})$，这正是我们想要证明的结论。◀

根据命题 4.7，存在一个最小基数的修复 $\mathbf{U} = (\mathrm{Ins}, \mathrm{Del})$，使得 $a \in Q(\mathbf{I}^x_{\mathbf{U}})$，其中 $\mathrm{Del} \equiv \emptyset$ 且 Ins 仅包含形如 $P(b_i)$、对应集合 $s_i \in S$ 的事实。因此$C = \{s_i \mid P(b_i) \in \mathrm{Ins}\}$ 是 U 的一个集合覆盖，所以 $\mathbf{MA}^{\mathbf{PJ}}_{min}(f(X)) \ge k$。

对于映射 h，给定一个输入 X 到 **MIN-SC** 以及某个修复 U，使得 $a_1, \ldots, a_n \in Q^X(\mathbf{I}^X_{\mathbf{U}})$，我们仿照命题 4.7 的证明，找到一个代价更低或相等的 U 的集合覆盖。首先，我们删除所有形如 $F(c, d)$ 的事实的删除和插入，其中 $d \ne a_i$ 对某个 $i \le n$ 成立。然后，对于某个 $i \le n$，我们将所有形如 $F(c, a_i)$ 的 F事实的插入，替换为任意一个满足 $u_i \in s_j$ 的 $P(b_j)$ 事实。h 的输出是与剩余 P 事实对应的集合族，它显然是 U 的一个集合覆盖，且代价不高于 U。注意，数对 (f, h) 既是严格归约（见定义 2.5），也是度量归约（见定义 2.9）。◀

我们以 $\mathbf{MA}^{\mathbf{Datalog}}_{min}$ 的一个上界作为本节的结尾。

▶**定理 4.8.** $MA^{\mathbf{Datalog}}_{min}$ 属于 **FEXP**，而 $MA^{\mathbf{sp\text{-}Datalog}}_{min}$ 属于 **F2EXP**。

证明。为证明 $\mathrm{MA}^{\mathbf{Datalog}}_{min}$ 属于 FEXP，回想一下，任何布尔 Datalog 程序P 都是单调的。此外，早期研究（[51]）已指出，P 可以在仅包含 adom(P) 中常量的数据库实例中得到满足。给定输入 $P, \mathrm{I}\ a$，我们首先通过将a 中的常量代入 P 中答案谓词为 ans_P的规则的自由变量，构造布尔 Datalog 程序 P'。根据单调性，只需对 I 进行插入即可修复 P'。此外，我们只需在常量集合 adom(P', I) 上进行插入；该集合是 Dom 的子集，其大小与输入规模呈线性关系。考虑如下算法：按基数递增的顺序，针对 Facts(adom(P', I)) \ Facts(adom(I)) 上每一种可能的更新 U，检查 U 是否为相对于 $P, \mathrm{I}\ a$ 的修复，方法是计算 $\mathrm{Eval}^{\mathbf{Datalog}}(\mathrm{I}_\mathrm{U})$。若是，则 U 是极小修复，因此返回 U。若找不到修复，则拒绝。由于循环的迭代次数与输入规模呈线性关系，且模拟 $\mathrm{Eval}^{\mathbf{Datalog}}$ 需要指数时间，因此不难看出该算法属于 FEXP。

$\mathbf{MA}^{\mathbf{sp\text{-}Datalog}}_{min}$ 属于**F2EXP** 的论证类似。在这种情况下，给定输入$\langle P, \mathbf{I}\ a\rangle$，我们需要考虑在$a \cup \mathbf{adom}(P, \mathbf{I}) \cup C$上进行的所有修复，包括插入和删除，其中 C 是一个大小相对于 P呈指数级增长的新常量集合。该搜索空间的充分性源于**sp-Datalog** 的**EXP** 可满足性判定过程 [35], ；稍作修改即可允许构造一个大小至多为指数级的可满足数据库实例**J** 。不失一般性，我们可以假设**J** 只包含$\mathbf{adom}(P) \cup C$上的事实。因此，暴力算法必须检查 $a \cup \mathbf{adom}(P, \mathbf{I}) \cup C$上的双指数多个可能更新U，其中$\mathbf{I}_\mathrm{U}$ 保证至多比输入大指数级。由此，整个暴力算法的运行时间为双指数级。◀

5 数据复杂度

本节将证明，对于所有 **sp-Datalog** 程序 P，$\mathbf{MA}^{P}_{min}$ 都是多项式时间可计算的。这将是一个一般性结果的特例：对于一大类查询，在数据复杂度下，其最小修复是多项式时间可计算的。为此，我们首先引入下面这个必要的定义，它是导出 $\mathbf{MA}^{\mathbf{sp\text{-}Datalog}}_{min}$ 多项式时间可计算性的关键观察。

▶ **定义 5.1. </style}如果存在某个，使得对于任意实例 I** 和元组 a，只要关于 $\langle Q, \mathbf{I}\ a\rangle$ 的修复存在，就存在一个修复 U′，关于 $\langle Q, \mathbf{I}\ a\rangle$，且满足 $|\mathrm{U}'| \leq c_Q$，则称查询 Q 具有常数大小修复 *</style}*。

请注意，定义 5.1 描述的是查询的一种性质，而非查询类。它只涉及修复大小的一个有限上界，但不一定意味着能够计算出修复所用常量的有界域。不过，如果映射 $Q \mapsto c_Q$ 可计算且 Q 是泛型的（见定义 3.6），则可以计算出这样的域——定理 5.4 的证明将默认这一点。然而，在研究数据复杂度时，$Q \to c_Q$ 是否可计算并不重要，重要的只是对于所关注的查询 Q，c_Q 是否存在。下面的例子有助于理解常数大小修复这一概念。

▶ 例 5.2。设 $Q \in$ 合取查询$_\neg$ 由规则 $S(x, y, z) :- R(x, y)\ R(y, z)\ \neg R(z, x)$ 定义。给定任意实例 I 和元组 a_1, a_2, a_3，修复 U 至多需要两次插入和一次删除，才能确保 $a_1, a_2, a_3 \in \mathrm{I}_\mathrm{U}$ ——也就是说，我们必须有

$R(a_1, a_2)$ $R(a_2, a_3)$ $\in \mathrm{I_U}$，并且 $R(a_3, a_1) \notin \mathrm{I_U}$。因此， Q 具有常数大小修复，且$c_Q = 3$。注意，具有常数大小修复并不意味着修复总是存在：例如，如果输入元组是 a_1, a_1, a_1，那么不存在修复 U，使得 $a_1, a_1, a_1 \in Q(\mathrm{I_U})$，因为这将要求事实 $R(a_1, a_1)$ 同时属于 $\mathrm{I_U}$且不属于 I。

▶ **引理 5.3**。 每个查询 $Q \in$ **合取查询**$_\neg$ 都存在常数大小修复。

证明。 令 $Q \in$**合取查询**$_\neg$ 为由以下形式的规则 r 定义的任意查询

$$S(\bar{x}) \ :- \ \alpha_1, \ldots, a_{n_1}, \neg\beta_1, \ldots, \neg\beta_{n_2}, \gamma_1, \ldots, \gamma_{n_3},$$

其中 α_i 是正文字， $\neg\beta_i$ 是负文字， γ_i 是等式或不等式原子。令 **I** 表示任意实例， a 表示 Dom 上的任意元组。为确保 $a \in \mathbf{I_U}$，我们需要找到一个修复 U，使得对于某个赋值 g，若 $g(x) = a$，则对于每个 $i \leq n_1$，都有 $\alpha_i[g] \in \mathbf{I_U}$；对于每个 $i \leq n_2$，都有 $\beta_i[g] \notin \mathbf{I_U}$；并且 g满足等式/不等式。对于满足 γ_i的任意赋值 g，所需插入的最大数量为 n_1，而在 n_2 中所需删除的最大数量为。因此，任何极小修复（如果存在）的最大大小为 $n_1 + n_2$。◀

▶**定理 5.4**。 如果 Q 是一个$\boldsymbol{adom}(Q)$-泛型查询，且其允许常数大小修复，并且$\boldsymbol{Eval}^Q$ 是多项式时间可计算的，那么$\boldsymbol{MA}^Q_{min}$ 属于**FP**。

证明。 设 Q 如命题所述，并设 $\langle \mathbf{I}\ a\rangle$ 是 $\mathbf{MA}^Q_{min}$的一个输入。令$m = \max\{$**元数**$(R) \mid R \in \Sigma\}$，并令 $C = \{c_1, \ldots\ldots, c_{m \cdot c_q}\}$是一个与 $\mathbf{adom}(Q, I) \cup a$ 不相交的常量集合。我们使用以下命题。

▷ 命题 5.5。如果存在一个最小基数修复 U，使得 $a \in Q(\mathrm{I_U})$，那么存在一个最小基数修复 U′，其位于 $C \cup \mathbf{adom}(Q, I) \cup a$上。

命题证明。令 D 表示 U 中出现但不出现在 adom$(Q, \mathrm{I}) \cup a$中的常量集合。由于 U 是最小的，且 Q 允许常数大小修复，因此有 $|D| \leq m \cdot c_Q$。根据 Q的 adom(Q)-泛型性，对于任意双射映射ρ : Dom $\to$ Dom（其 $\mathrm{Im}_\rho[D] \subseteq C$，且逐点固定 adom$(Q) \cup a$），都有 $\rho(Q(\mathrm{I_U})) = Q(\rho(\mathrm{I_U}))$。特别地，将 U 中每个常量替换为其在 ρ下的像所得到的 $\mathrm{U}' = \rho(\mathrm{U})$，也是一个位于 $C \cup$ adom$(Q, I) \cup a$ 上的最小基数修复，并且满足 $a \in Q(\mathrm{I_{U'}})$。◀

要找到一个最小更新 U，使得 $a \in Q(I)$，根据命题 5.5，只需检查所有基于 $\mathbf{adom}(Q,\mathbf{I}) \cup C \cup a$（按大小顺序），使得 $|\mathrm{U}| \leq c_Q$，判断 $a \in Q(\mathbf{I_U})$ 是否成立（通过调用 $\mathbf{Eval}^Q$），并输出找到的第一个修复。由于 Q 和 c_Q 都是常量，而 $\mathbf{adom}(Q,\mathbf{I}) \cup C \cup a$ 与输入大小呈线性关系，因此该过程只检查 $O(n^{c_Q})$ 个修复。此外，由于 $\mathbf{Eval}^Q$ 属于 **FP**，该算法也属于**FP**。◀

现在回顾一下展开集 [13]的概念。直观地说，sp-Datalog 程序 P 的一个展开集是一个合取查询，其头谓词为 $\mathbf{ans}_P$，体中只包含 EDB 谓词；它通过反复将 P 的某条规则中的 IDB 原子替换为 P 中各规则的体而得到，这些规则的头部包含相关的 IDB 谓词。

▶ **定义 5.6**（展开集）。 设 P是一个**sp-Datalog** 程序，其规则为 $r_1, \ldots, r_m$，答案谓词为 $\mathbf{ans}_P$。我们将 P^∞ 定义为包含 P中每条以$\mathbf{ans}_P$ 为头部的规则，且满足以下条件的最小集合：每当 P^∞ 包含一条形式如下的规则 r 时

$$\mathbf{ans}_P(\bar{x}) \ :- \ \alpha_1, \ldots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \ldots, \alpha_n,$$

其中 $\mathbf{rel}(\alpha_i) = S \in \mathbf{S}$、$\mathbf{var}(\alpha_i) = \bar{y}$，并且 r' 是 P 中形式为 $S(u) :- \textbf{体}_{r'}(u, v)$的规则，$\bar{}$其中 $\textbf{自由}(r') = u$ $\bar{}$且 $\textbf{受限}(r') = v$，那么 P^∞ 也包含规则

$$\mathbf{ans}_P(\bar{x}) \ :- \ \alpha_1, \ldots, \alpha_{i-1}, \mathbf{Body}_{r'}(\bar{y}, \bar{z}), \alpha_{i+1}, \ldots, \alpha_n$$

其中，$\bar{}z$ 是新变量组成的元组，且这些变量未出现在 r 或 r'中。我们将**展开集**(P) 定义为 P^∞ 中所有满足以下条件的规则 r 的集合： r 的体仅包含外延关系符号。

下一个定理最初针对**Datalog** 提出，但也可推广到**sp-Datalog** 程序；它使我们能够将**sp-Datalog** 程序视为**合取查询**$^\neg$ 查询的（可能是无限的）并。

▶ **定理 5.7** ([13])。 如果 $P \in$**sp-Datalog** 的答案谓词为$\mathbf{ans}(x)$，那么 $\bar{}$

$$\mathbf{ans}(\mathbf{I}) = \{\bar{a} \in \mathbf{Dom}^{len(\bar{x})} \mid \bar{a} \in Q(\mathbf{I}) \ \textit{for some} \ Q \in \mathbf{Unfoldings}(P)\}.$$

▶ **推论 5.8**。 $\boldsymbol{MA}^P_{min}$ 可针对所有 $P \in$**sp-Datalog**在多项式时间内计算。

证明。已知半正 Datalog 查询的求值可在关于数据复杂度的多项式时间内计算 [30], ，因此，根据定理 5.4，只需证明每个 $P \in$ sp-Datalog 都允许常数大小修复。为此，固定一个 $P \in$ sp-Datalog 程序，令 adom(P) $= a_1, \ldots, a_n$，并设 $k =$ arity(ans)，其中 ans 是P的答案谓词。定义 Dom^{k}上元组之间的等价关系 $\equiv$；对于元组 $b = b_1, \ldots, b_k$ 和$c = c_1, \ldots, c_k$，当且仅当以下两个条件都成立时，令 $b \equiv c$：

1. $b_i = b_j$ 当且仅当 $c_i = c_j$，并且
2. $b_i = a_j$ 当且仅当 $c_i = a_j$。

如果 $\bar{}b \equiv c$，那么根据 P 的 adom(Q)-泛型性，对于某个实例 I， $b \in$ ans(I) 当且仅当 $c \in$ans(ρ(I))，其中 $\rho :$ Dom $\to$Dom 是通过设置 $\rho(b) = c$、 $\rho(c) = b$，并对所有其他 $e \in$ Dom 设置 $\bar{}\rho(\bar{e}) = e$ 而得到的双射映射。因此，对于 Dom^k\ $\equiv$中的每个等价类 C，要么存在某个 $Q_C \in$ Unfoldings(P),$_-$使得对于所有 $b \in$ C，都存在一个实例 I 满足 $b \in Q_C$ (I)；要么不存在这样的 Q_C。在第一种情况下，令 $N_C = c_{Q_C}$；否则，令 $N_C = 0$。由于等价类的数量是有限的，因此 $c_P = \max_{C \in \mathbf{Dom}^k \backslash \equiv} c_{Q_C}$ 是有限的。于是， P 具有由 c_P 界定的常数大小修复，这正是我们要证明的结论。◀

由于 $\mathbf{MA}^P_{min}$ 对于 **sp-Datalog** 中的程序 P是多项式时间可计算的，因此自然会想到，上述论证能否扩展到关系代数（**RA**），甚至分层 Datalog（**Datalog**$^\neg$）。下面的例子表明并非如此。

▶ **例 5.9**。 考虑以下布尔型 **Datalog**$^\neg$ 查询，它也可以用 **RA** 表示，其规则为 $R :- P(x)$ 和 **ans** $:-$ $\neg R$，其中 P 是一个 EDB 谓词，而 R、 **ans** 是 IDB 谓词。用关系演算的记法表示，这等价于公式$\forall x \neg P(x)$（其中全称量词的取值范围为活动域）。容易看出，对于任意实例 **I**，**ans(I)** $\neq \emptyset$ 当且仅当 **I** 不包含任何形如$P(a)$的事实，其中 $a \in$ **Dom**任意。因此，对于此查询，最小修复的规模可能任意大。

上面的例子揭示了一个重要主题：**sp-Datalog** 中的所有查询 Q 都是存在性的，只需修改数据库中一小块局部 区域即可满足。另一方面，例 5.9 中的查询具有全称性 ，因为它表达的是关于数据库活动域的全局条件。因此，该查询的最小修复可能任意大，具体取决于输入数据库中的 P 事实数量。对于缺失答案问题，“存在性”和“局部”规则会带来可处理的复杂度；而对于错误答案问题，情况恰恰相反，这体现了引言中讨论的对偶性。

推论 4.3 对判定是否存在修复的联合复杂性，尤其是对**关系代数** 及其扩展，给出了悲观的图景。然而，在数据复杂度的设定下，对于**关系代数** 查询，$\mathbf{MA}^{Q}_{min}$ 问题总是可判定的。为此，注意到给定一个固定的**关系代数**查询 $Q \in$ ，以下算法之一是正确的：

1. 如果 Q 可满足：算法从最小规模开始，依次枚举所有可能的修复，直到找到一个修复；以及
2. 如果 Q不可满足：该算法对所有输入均返回“无修复”。

当然，无法判定上述哪种算法是正确的；即使对于我们已知可满足的**关系代数** 查询，这种暴力搜索的运行时间也极难处理。然而，对于**关系代数** 查询，这似乎就是我们一般所能得出的最佳结论，因为通常不存在此类查询所需修复规模的可计算上界。如果确实存在这样的上界，那么只需检查输入$\langle Q,\mathbf{E}\ \epsilon\rangle$ 在该可计算上界内的所有修复，即可求解 $\mathbf{dbSAT}^{\mathbf{RA}}$ —— 但这是不可能的。

6 总结与未来工作

本文的主要目标是确定寻找缺失答案修复时可处理性与不可处理性之间的分界线。在联合复杂性方面，我们针对合取查询的并集的若干重要子类回答了这一问题。我们还证明了**EXP**-困难的**Datalog** 和**sp-Datalog**，并给出了相匹配的**FEXP** （分别为**F2EXP**）完全性结果，适用于$\mathbf{MA}^{\mathbf{Datalog}}_{min}$ （分别适用于$\mathbf{MA}^{\mathbf{sp\text{-}Datalog}}_{min}$ ）。在数据复杂性方面，我们证明，对于表达能力较强的**sp-Datalog** 查询，计算缺失答案修复是可处理的。我们还讨论了将可处理性扩展到具有更强形式否定的查询时所面临的主要困难。

未来研究的一个方向是填补表 1 中的空缺，例如$\mathbf{MA}^{\mathbf{sp\text{-}Datalog}}_{size}$、$\mathbf{MA}^{\mathbf{sp\text{-}Datalog}}_{min}$ 和 $\mathbf{MA}^{\mathbf{RA}}_{bound}$对应的空缺。如引言中所述，缺失答案修复也是对数据库完整性约束修复问题的一种优雅推广，因此，将本文的工作扩展到能够以有限方式表达全称量化的查询类，从而建立更一般的一致性修复理论，将十分有价值。此外，尽管本文着重研究最小化基数修复，但探究对所计算出的修复施加其他期望约束也很有意义，例如尽量减少对查询答案的副作用。另一个研究方向是将该问题推广为允许输入中出现任意数量的元组，并要求通过修复将所有这些元组都添加到查询答案中。

另一个重要的未来研究方向，是考虑将缺失答案修复问题推广到半环语义 [25, 24]。在这一推广中，数据库中的事实不再只是非真即假，而是带有固定半环中的值作为注释。查询求值的结果通过简单的半环运算传播这些注释，为每个输出生成一个注释。本文可以视为研究集合语义这一特殊情形，而集合语义是由布尔半环自然导出的。

参考文献

1 Serge Abiteboul、Richard Hull 和 Victor Vianu。《数据库基础：逻辑层》。Addison-Wesley Longman Publishing Co., Inc.，美国，第一版，1995年。2 Marcelo Arenas、Leopoldo Bertossi 和 Jan Chomicki。用于不一致数据库一致查询回答的答案集。 理论与实践逻辑程序，3(4):393–424，2003年7月。doi:10.1017/S1471068403001832。

3 Marcelo Arenas、Leopoldo Bertossi 和 Jan Chomicki。用于不一致数据库一致查询回答的答案集。《理论与实践逻辑程序》，3(4):393–424，2003年7月。doi:10.1017/S1471068403001832。 4 F. Bancilhon 和 N. Spyratos。关系视图的更新语义。《ACM 数据库系统汇刊》，6(4):557–575，1981年12月。 doi:10.1145/319628.319634。 5 Leopoldo Bertossi。通过修复衡量并计算数据库不一致性。载于 Davide Ciucci、Gabriella Pasi 和 Barbara Vantaggi（编），《可扩展不确定性管理》，第368–372页，Cham，2018年。Springer 国际出版。 doi:10.1007/978-3-030-00461-3_26。 6 Nicole Bidoit、Melanie Herschel 和 Katerina Tzompanaki。为等价合取查询不可变地回答为何不问题。载于 Adriane Chapman、Bertram Ludäscher 和 Andreas Schreiber（编），第六届数据溯源理论与实践研讨会（TaPP'14），德国科隆，2014年6月12–13日。USENIX 协会，2014年。网址： https://www.usenix.org/conference/tapp2014/agenda/presentation/bidoit。 7 Nicole Bidoit 、 Melanie Herschel 和 Katerina Tzompanaki。使用 NedExplain 的基于查询的为何缺失溯源。载于 Sihem Amer-Yahia、Vassilis Christophides、Anastasios Kementsietsidis、Minos N. Garofalakis、Stratos Idreos 和 Vincent Leroy（编），第十七届数据库技术拓展国际会议论文集（EDBT 2014），希腊雅典，2014年3月24–28日，第145–156页。OpenProceedings.org，2014年。 doi:10.5441/002/EDBT.2014.14。 8 Peter Buneman、Sanjeev Khanna 和 Wang-Chiew Tan。删除与注释在视图中的传播。载于第二十一届 ACM SIGMOD-SIGACT-SIGART 数据库系统原理研讨会论文集（PODS '02），第150–158页，美国纽约州纽约市，2002年。计算机协会。 doi:10.1145/543613.543633。 9 Peter Buneman、Sanjeev Khanna 和 Tan Wang-Chiew。原因与位置：数据溯源的刻画。载于 Jan Van den Bussche 和 Victor Vianu（编），《数据库理论 — ICDT 2001》，第316–330页，德国柏林、海德堡，2001年。Springer Berlin Heidelberg。 doi:10.1007/3-540-44503-X_20。 10 Nofar Carmeli、Martin Grohe、Benny Kimelfeld、Ester Livshits 和 Muhammad Tibi。基于软函数依赖的数据库修复。《ACM 数据库系统汇刊》，49(2)，2024年4月。doi:10.1145/3651156。 11 Ashok K. Chandra 和 Philip M. Merlin。关系数据库中合取查询的最优实现。载于第九届 ACM 年度计算理论研讨会论文集（STOC '77），第77–90页，美国纽约州纽约市，1977年。计算机协会。 doi:10.1145/800105.803397。 12 Adriane Chapman 和 H. V. Jagadish。为何不？载于 Ugur Çetintemel、Stanley B. Zdonik、Donald Kossmann 和 Nesime Tatbul（编），ACM SIGMOD 数据管理国际会议论文集（SIGMOD 2009），美国罗德岛州普罗维登斯，2009年6月29日至7月2日，第523–534页。ACM，2009年。 doi:10.1145/1559845.1559901。 13 Surajit Chaudhuri 和 Moshe Y Vardi。递归与非递归 Datalog 程序的等价性。《计算机与系统科学杂志》，54(1):61–78，1997年。 doi:10.1006/jcss.1997.1452。 14 Gao Cong、Wenfei Fan 和 Floris Geerts。键保持视图中的注释传播再探。载于第十五届 ACM 信息与知识管理国际会议论文集（CIKM '06），第632–641页，美国纽约州纽约市，2006年。计算机协会。 doi:10.1145/1183614.1183705。 15 Gao Cong、Wenfei Fan、Floris Geerts、Jianzhong Li 和 Jizhou Luo。视图更新分析的复杂性及其在注释传播中的应用。《IEEE 知识与数据工程汇刊》，24(3):506–519，2012年。 doi:10.1109/TKDE.2011.27。 16 Stavros S. Cosmadakis 和 Christos H. Papadimitriou。关系视图的更新。《ACM 杂志》，31(4):742–760，1984年9月。 doi:10.1145/1634.1887。

17 Evgeny Dantsin、Thomas Eiter、Georg Gottlob 和 Andrei Voronkov。逻辑编程的复杂性与表达能力。载于《第十二届 IEEE 年度计算复杂性会议论文集》，德国乌尔姆，1997年6月24日至27日，第82–101页。IEEE计算机学会，1997年。 doi:10.1109/CCC.1997.612304。18 Umeshwar Dayal 和 Philip A. Bernstein。关系视图上更新操作的正确转换。《ACM数据库系统汇刊》，7(3):381–416，1982年。 doi:10.1145/319732.319740。19 Irit Dinur 和 David Steurer。并行重复的分析方法。载于《第四十六届 ACM 年度计算理论研讨会论文集》，STOC '14，第624–633页，美国纽约州纽约，2014年。计算机协会。 doi:10.1145/2591796.2591884。20 Wenfei Fan 和 Floris Geerts。《数据质量管理基础》。Morgan & Claypool 出版社，2012年。21 Cibele Freire、Wolfgang Gatterbauer、Neil Immerman 和 Alexandra Meliou。无自连接合取查询的韧性与责任度的复杂性。《VLDB基金会会刊》，9(3):180–191，2015年。 doi:10.14778/2850583.2850592。22 Cibele Freire、Wolfgang Gatterbauer、Neil I mmerman 和 Alexandra Meliou。含自连接的二元合取查询的韧性复杂性新结果。载于 Dan Suciu、Yufei Tao 和 Zhewei Wei 编，《第39届 ACM SIGMOD-SIGACT-SIGAI 数据库系统原理研讨会论文集》，PODS 2020，美国俄勒冈州波特兰，2020年6月14日至19日，第271–284页。ACM，2020年。 doi:10.1145/3375395.3387647。23 William Gasarch、M. Krentel 和 K. Rappoport。OptP 是 NP完全问题的常规行为。《计算系统理论 / 数学系统理论》，28，1995年11月。 doi:10.1007/BF01204168。24 Erich Grädel 和 Val Tannen。一阶逻辑的溯源分析与半环语义。载于 Klaus Meer、Alexander Rabinovich、Elena Ravve 和 Andrés Villaveces 编，《模型论、计算机科学与图多项式：纪念 Johann A. Makowsky 文集》，第351–401页。Springer Nature Switzerland，Cham，2025年。 doi:10.1007/978-3-031-86319-6_21。25 T. Green、G. Karvounarakis 和 V. Tannen。溯源半环。载于《数据库系统原理 PODS》，第31–40页。ACM，2007年。 doi:10.1145/1265530.1265535。26 Martin Grohe 和 Goetz Schwandtner。线性序上 Datalog 的复杂性。《计算机科学中的逻辑方法》，第5卷，第1期，2009年2月。 doi:10.2168/LMCS-5(1:4)2009。27 Melanie Herschel 和 Mauricio A. Hernández。解释 SPJUA 查询中的缺失答案。《VLDB基金会会刊》，3(1):185–196，2010年。 doi:10.14778/1920841.1920869。28 Melanie Herschel、Mauricio A. Hernández 和 Wang Chiew Tan。Artemis：一种分析缺失答案的系统。《VLDB基金会会刊》，2(2):1550–1553，2009年。 doi:10.14778/1687553.1687588。29 Jiansheng Huang、Ting Chen、AnHai Doan 和 Jeffrey F. Naughton。抽取数据上查询非答案的来源。《VLDB基金会会刊》，1(1):736–747，2008年。doi:10.14778/1453856.1453936。30 Neil Immerman。可在多项式时间内计算的关系查询。《信息与控制》，68(1):86–104，1986年。 doi:10.1016/S0019-9958(86)80029-8。31 Benny Kimelfeld。函数依赖下删除传播复杂性的二分性。载于 Michael Benedikt、Markus Krötzsch 和 Maurizio Lenzerini 编，《第31届 ACM SIGMOD-SIGACT-SIGART 数据库系统原理研讨会论文集》，PODS 2012，美国亚利桑那州斯科茨代尔，2012年5月20日至24日，第191–202页。ACM，2012年。doi:10.1145/2213556.2213584。32 Benny Kimelfeld、Jan Vondrák 和 Ryan Williams。删除传播中合取视图的最大化。《ACM数据库系统汇刊》，37(4):24:1–24:37，2012年。 doi:10.1145/2389241.2389243。33 Solmaz Kolahi 和 Laks V. S. Lakshmanan。关于函数依赖违反的最优修复近似。载于《第12届国际数据库理论会议论文集》，

ICDT '09，第53–62页，美国纽约州纽约，2009年。计算机协会。doi:10.1145/1514894.1514901。34 Mark W. Krentel。优化问题的复杂性。《计算机与系统科学杂志》，36(3):490–509，1988年。 doi:10.1016/0022-0000(88)90039-6。35 Alon Levy、Inderpal Singh Mumick、Yehoshua Sagiv 和 Oded Shmueli。Datalog 扩展中的等价性、查询可达性与可满足性。载于《第十二届 ACM SIGACT-SIGMOD-SIGART 数据库系统原理研讨会论文集》，PODS '93，第109–122页，美国纽约州纽约，1993年。计算机协会。 doi:10.1145/153850.153860。36 Andrei Lopatenko 和 Leopoldo Bertossi。基于基数和增量修复语义下数据库一致查询回答的复杂性。载于 Thomas Schwentick 和 Dan Suciu 编，《数据库理论——ICDT 2007》，第179–193页，柏林、海德堡，2006年。Springer Berlin Heidelberg。doi:10.1007/11965893_13。37 Neha Makhija 和 Wolfgang Gatterbauer。利用整数线性规划（整数线性规划）和线性规划松弛统一处理韧性与因果责任度。《ACM数据管理会议录》，1(4):228:1–228:27，2023年。 doi:10.1145/3626715。38 A. Meliou、W. Gatterbauer、K. Moore 和 D. Suciu。为什么是这样？还是为什么不是？用于解释查询答案的函数因果性。载于 VLDB 不确定数据管理研讨会（MUD 2010），CTIT 研讨会论文集系列第 WP10-04 卷，第3–17页，2010年。39 Alexandra Meliou、Wolfgang Gatterbauer、Joseph Y. Halpern、Christoph Koch、Katherine F. Moore 和 Dan Suciu。数据库中的因果性。《IEEE数据工程简报》，33(3):59–67，2010年。40 Dongjing Miao 和 Zhipeng Cai。聚合查询韧性复杂性研究。载于 Donghyun Kim、R. N. Uma 和 Alexander Zelikovsky 编，《组合优化与应用》，第696–706页，Cham，2018年。Springer 国际出版。 doi:10.1007/978-3-030-04651-4_47。41 Dongjing Miao、Zhipeng Cai 和 Jianzhong Li。合取查询有界视图传播的复杂性研究。《IEEE 知识与数据工程汇刊》，30(1):115–127，2018年。 doi:10.1109/TKDE.2017.2758361。42 Dongjing Miao、Zhipeng Cai 和 Jianzhong Li。多重键保持视图的删除传播再探。《IEEE 知识与数据工程汇刊》，35(3):2445–2456，2023年。 doi:10.1109/TKDE.2021.3110851。43 Dongjing Miao、Zhipeng Cai、Xianmin Liu 和 Jianzhong Li。函数依赖约束下插入传播的复杂性研究。载于 Thang N. Dinh 和 My T. Thai 编，《计算与组合数学》，第623–632页，Cham，2016年。Springer 国际出版。 doi:10.1007/978-3-319-42634-1_50。44 Dongjing Miao、Zhipeng Cai、Xianmin Liu 和 Jianzhong Li。受函数依赖约束的插入传播。《理论计算机科学》，819:1–8，2020年。计算与组合数学。 doi:10.1016/j.tcs.2017.03.043。45 Dongjing Miao、Jianzhong Li 和 Zhipeng Cai。删除传播复杂性的新结果。载于 Zhao Zhang、Wei Li 和 Ding-Zhu Du 编，《信息与管理中的算法问题》，第336–345页，Cham，2020年。Springer 国际出版。doi:10.1007/978-3-030-57602-8_30。46 Pekka Orponen 和 Heikki Mannila。关于保持近似性的归约：完全问题与稳健度量。赫尔辛基大学，1987年。47 Christos H. Papadimitriou。《计算复杂性》。Addison-Wesley，1994年。48 Christos H. Papadimitriou 和 Mihalis Yannakakis。优化、近似与复杂性类。《计算机与系统科学杂志》，43(3):425–440，1991年。 doi:10.1016/0022-0000(91)90023-X。49 Sudeepa Roy、Laurel J. Orr 和 Dan Suciu。利用可解释数据库解释查询答案。《VLDB基金会会刊》，9(4):348–359，2015年。 doi:10.14778/2856318.2856329。50 Sudeepa Roy 和 Dan Suciu。为数据库查询寻找解释的形式化方法。载于 Curtis E. Dyreson、Feifei Li 和 M. Tamer Özsu 编，《国际会议论文集》

数据管理，SIGMOD 2014，美国犹他州斯诺伯德，2014年6月22—27日，第1579–1590页。ACM，2014年。 doi:10.1145/2588555.2588578。51 Oded Shmueli。DATALOG 查询的等价性是不可判定的。《逻辑程序设计杂志》，15(3):231–241，1993年。 doi:10.1016/0743-1066(93)90040-N。52 Wang-Chiew Tan。带注释传播的关系查询包含问题。载于 Georg Lausen 和 Dan Suciu 编，《数据库编程语言》，第37–53页，柏林、海德堡，2004年。Springer Berlin Heidelberg。 doi:10.1007/978-3-540-24607-7_4。53 Boris Trakhtenbrot。有限类上的判定问题不存在算法。苏联科学院院刊，70:569–572，1950年。54 Quoc Trung Tran 和 Chee-Yong Chan。如何解决为何不问题。载于 Ahmed K. Elmagarmid 和 Divyakant Agrawal 编，《ACM SIGMOD 数据管理国际会议论文集》，SIGMOD 2010，美国印第安纳州印第安纳波利斯，2010年6月6—10日，第15–26页。ACM，2010年。 doi:10.1145/1807167.1807172。55 Moshe Y. Vardi。关系查询语言的复杂性（扩展摘要）。载于 Harry R. Lewis、Barbara B. Simons、Walter A. Burkhard 和 Lawrence H. Landweber 编，《第14届 ACM 计算理论研讨会论文集》，1982年5月5—7日，美国加利福尼亚州旧金山，第137–146页。ACM，1982年。 doi:10.1145/800070.802186。56 Eugene Wu 和 Samuel Madden。Scorpion：解释聚合查询中的离群值。《VLDB基金会会刊》，6(8):553–564，2013年。 doi:10.14778/2536354.2536356。57 J. Xu、W. Zhang、A. Alawini 和 V. Tannen。缺失答案与完整性修复的溯源分析。《IEEE数据工程简报》，41(1):39–50，2018年。URL： http://sites.computer.org/debull/A18mar/p39.pdf。

A 附录

回顾一下，我们将 **Datalog** 定义为仅包含正文字的程序。本节中，我们通过从二元素线性序上的 Datalog 非空性问题进行多项式时间归约，证明$\mathbf{dbSAT}^{\mathbf{Datalog}}$ 是**EXP**-难的。据作者所知，$\mathbf{dbSAT}^{\mathbf{Datalog}}$ 的这一下界是未发表的民间结果。

▶ **定义 A.1**（针对数据库实例 **I** 的 Datalog 非空性问题）。 设 **I** 表示一个固定的数据库实例。在 **I** 上，Datalog 非空性问题 是指：给定一个**Datalog** 程序 P （基于**I**的 *EDB* 模式），判断 P（**I**） 是否非空。

▶ **命题 A.2** （[26]）。 在$\mathbf{I}_2 = \{R(a,\ b)\}$（即表示一个2元素线性序的数据库）上的 *Datalog* 非空性问题是**EXP**-难的。

▶**定理 A.3.** $dbSAT^{\mathbf{Datalog}}$ 是**EXP**-难的。

证明。 不失一般性，我们可以假设 Σ 中的 EDB 至少包含一个关系符号（否则无法编写安全程序），并且二元关系符号 R 不出现在 Σ中。给定模式 r 上的规则$\{R\}$，令 π（r）为通过以下变换得到的规则：

1. 对于 r 中出现的每个等式原子 $s = t$，我们将 $s = t$ 放入 $\pi(r)$;
2. 对于 r 中的每个原子 $R(s,t)$，我们将一个原子 $S(s,t)$ 放入 $\pi(r)$;
3. 对于 r 中出现的每个变量 z，我们将一个原子 $V(z)$ 放入 $\pi(r)$。

令 P 表示模式 $\{R\}$ 上的任意 Datalog 程序，该程序包含规则 $\widehat{r_1, \ldots, r_m}$，答案谓词为 $\mathbf{ans}_P(x)$。我们定义一个模式$\{U\}$上的程序 P，其答案

谓词为 $\mathbf{ans}_{\widehat{P}}$，该程序包含规则 $\pi(r_i)$ for $i \leq m$，以及以下规则

$$V(x) \ :- \ x = a, U(x, \ldots, x), \tag{2}$$
$$V(x) \ :- \ x = b, U(x, \ldots, x), \tag{3}$$
$$S(x, y) \ :- \ x = a, y = b, U(x, \ldots, x), U(y, \ldots, y), \tag{4}$$
$$\mathbf{ans}_{\widehat{P}} \ :- \ U(a, \ldots, a), U(b, \ldots, b), \mathbf{ans}_P(\overline{x}), \tag{5}$$

分别，其中 V、 S 和 $\mathrm{ans}_{\widehat{P}}$ 是在 P 中未出现的 IDB。注意，对于任意数据库实例 I，若 $P(\mathrm{I}) \neq \emptyset$，则 $U(a, \ldots, a)\ U(b, \ldots, b) \in \mathrm{I}$，这进一步意味着 $\sim S(\mathrm{I}) = \{(a, b)\}$ 且 $V(\mathrm{I}) = \{a, b\}$。显然，映射 $P \mapsto \widehat{P}$ 可在多项式时间内计算，因此只需证明 $P(\mathrm{I}_2) \neq \emptyset$ 当且仅当 $\widehat{P}$ 可满足。

对于“仅当”方向，假设 $P(\mathrm{I}_2) \neq \emptyset$。那么存在某个合取查询 $Q_1 \in \mathrm{Unfoldings}(P)$，使得 $Q_1(\mathrm{I}_2)$ 非空。令 g 为 Q_1 在 I_2 中的一个满足赋值。根据安全条件，我们可以假设 $\mathrm{rng}(g) \subseteq \{a, b\}$。注意， $\pi(Q_1)\widehat{\ }$ 属于 P^∞，并且注意到 $\pi(Q_1)$ 不包含任何形如 $V(t)$ 的原子，其中 t 是常量。令 Q_1' 为由规则 r' 定义的合取查询，它由 $\pi(Q_1)$ 通过将每个原子 $S(s, t)$ 替换为原子 $s = a, t = b, U(s, \ldots, s)\ U(t, \ldots, t)$，并将 $\pi(Q_1)$ 中的每个原子 $\sim V(z)$ 替换为事实 $z = g(z)\ U(z, \ldots, z)$ 得到。显然， $Q_1' \in P^\infty$，且只包含 EDB 谓词。最后，令 Q_1'' 为由下述规则 r'' 定义的合取查询

$$\mathbf{ans}_{\widehat{P}} \ :- \ U(a, \ldots, a), U(b, \ldots, b), \mathbf{Body}_{r'}.$$

显然 $Q_1'' \in$ **展开集**$(\widehat{P})$，且 g 是**I**中 Q_1''的满足赋值。因此 $\widehat{P}(\mathbf{I}) \neq \emptyset$， $\widehat{\ }$，所以 P 是可满足的。

$\widehat{\ }$对于“若”方向，假设 P 是可满足的。那么存在某个**I** ，使得$\widehat{P}(\mathbf{I}) \neq \emptyset$。因此存在某个 $Q_2 \in$ **展开集**$(\widehat{P})$，使得 $Q_2(\mathbf{I})$非空。令$\sim g$ 为**I**中 Q_2 的满足赋值。根据 P的构造，必定存在一个合取查询 $Q_2'\widehat{\ }$属于 P^∞ ，其中包含 V 和 S 谓词，并且 Q_2' 是通过将这些谓词替换为规则 (2)-(4) 的主体的适当替换而得到的。这意味着存在某个 $Q \in$ **展开集**(P)，使得 $\pi(Q) = Q_2'$，且 g 是 $\mathbf{I}$ 中 Q 的满足赋值 $_2$。因此 $P(\mathbf{I}_2)$ 非空。◀