Datalog 程序的右伴随与受限类上的同态对偶

BaldertenCate ！ 阿姆斯特丹大学逻辑、语言与计算研究所，荷兰

VíctorDalmau ！ 信息与通信技术系，庞培法布拉大学，西班牙

Jakub Opršal 奥地利科学技术研究所，克洛斯特新堡，奥地利

摘要

Datalog 程序可视为一种句法规范，用于描述从某个模式上的数据库实例到另一模式上的数据库实例的函子。对于∃Datalog，情况也更为一般地成立。我们确定了大类 Datalog 和 ∃Datalog 程序，其对应的函子具有广义右伴随。我们运用这些结果，深入探讨受限实例类中同态对偶的存在性及其构造方法。我们还推导出有关数据库查询的唯一表征数据示例是否存在的新结果。

2012年 美国计算机协会学科分类计算理论 → 逻辑与数据库

关键词与短语Datalog、伴随、同态对偶、数据库约束、合取查询

数字对象标识符10.4230/LIPIcs...

资助*Balder ten Cate*：获欧盟地平线2020研究与创新计划资助（MSCA-101031081）。

Jakub Opršal：获欧盟地平线2020研究与创新计划资助，资助协议为玛丽·斯克沃多夫斯卡-居里资助协议 No 101034413。

arXiv:2302.06366v1 [cs.LO] 13 Feb 2023

1 引言

Datalog 是一种基于规则的语言，用于定义从输入模式 $\mathbf{S}_{in}$上的数据库实例到输出模式 $\mathbf{S}_{out}$上的数据库实例的映射。

▶ **例 1.1.** 考虑由以下规则定义的 Datalog 程序：

$Path(x,\ y) :- Edge(x,\ y)$。$Path(x,$
$y) :- Edge(x,\ z)\ Path(z,\ y)$。
$Ans(x,\ y) :- Path(x,\ y)$。

该 Datalog 程序以输入模式$\{Edge\}$上的一个实例作为输入，并生成模式 $\{Ans\}$上的一个实例作为输出，其中 Ans 是 $Edge$的传递闭包。

用范畴论的术语来说，Datalog 程序定义了一个从 $\mathrm{Inst}[\mathbf{S}_{in}]$ 到 $\mathrm{Inst}[\mathbf{S}_{out}]$的函子，其中 $\mathrm{Inst}[\mathbf{S}]$ 表示模式 $\mathbf{S}$ 上的所有数据库实例构成的范畴，态射为同态。

我们研究此类函子的右伴随和广义右伴随是否存在。对于任意函子 $F\colon X \to Y$，我们将右伴随定义为一个函子 $G\colon Y \to X$，使得对于所有 $A \in X$ 和 $B \in Y$，$F(A) \to B$ 当且仅当 $A \to G(B)$。粗略地说，广义右伴随 的定义类似，只是允许 G 将一个对象 $B \in Y$映射到一个有限

©Balder ten Cate、Víctor Dalmau 和 Jakub Opršal；依据知识共享许可协议 CC-BY 4.0 授权。莱布尼茨国际信息学会议论文集，Schloss Dagstuhl – Leibniz-Zentrum für Informatik，Dagstuhl Publishing，德国

XX：2 Datalog 程序的右伴随，以及同态对偶

对象集 （属于 X），使得对于所有 $A \in X$ 和 $B \in Y$， $F(A) \to B$ 当且仅当存在某个$B' \in G$ 使得 $A \to B'(B)$。事实证明，例 1.1 中的 Datalog 程序 P 具有右伴随。还有一些 Datalog 程序不具有右伴随，但具有广义右伴随；也有一些 Datalog 程序不具有广义右伴随。

▶ **贡献 1** （第3节）。 我们引入一种称为$TAM\ Datalog$的新型 Datalog 片段（即树形近单子 *Datalog*）。我们从语义上将 TAM Datalog 刻画为单子二阶逻辑的一个片段，并证明 TAM Datalog 在复合运算下封闭。

▶ **贡献 2** （第4节）。 我们证明，每个连通 TAM Datalog 程序都有一个右伴随，并且每个 TAM Datalog 程序都有一个广义右伴随。我们通过反例证明，TAM Datalog 所施加的每个语法条件对于广义右伴随的存在都是必要的。我们还确定了 Datalog 的一个更大片段 $\exists$（即带存在量词的 Datalog 扩展），该片段允许广义右伴随。

我们研究（广义）右伴随的动机源于这样一个事实：它们为我们提供了构造同态对偶的方法。同态对偶是一对（F， D），其中 F 和 D 是实例的集合，任意实例 A 能够从 F 中的某个实例接受一个同态，当且仅当 A 不能同态到 D中的任何实例。换言之，同态对偶将一种同态存在等同于另一种同态不存在。同态对偶在约束满足问题的相关文献中得到了广泛研究，也已在其他领域得到应用，例如数据库理论和知识表示领域。

▶ **贡献 3** （第 6 节）。我们证明，广义右伴随可用于构造同态对偶，并针对受限实例类（例如传递有向图）中有限同态对偶的存在性得出了新的结果。

▶ **贡献 4** （第 7 节）。在 [2, 11], 中，同态对偶被用作研究合取查询及合取查询的并的唯一可刻画性和精确可学习性的工具。沿着这一思路，我们针对存在背景理论时合取查询的并的唯一可刻画性，得出了新的结果，并回答了 [11]中的一个开放问题。

相关工作 Foniok 和 Tardif [18] 研究了 Pultr 函子（其自身也是右伴随）的右伴随是否存在这一问题 [23]，特别考察了有向图的情形。用我们的术语来说，Pultr 函子是一种解释（将有向图解释为有向图）(ϕ_V， ϕ_E)，其中 ϕ_V和 ϕ_E是合取查询（分别具有 k 和 $2k$ 个自由变量，某个 $k \geq 1$），分别定义输出节点集和边集。对于 ϕ_V 仅返回输入节点集的特殊情形， [18] 证明，由 (ϕ_V， ϕ_E)定义的函子存在右伴随，当且仅当 ϕ_E 是连通且无环的。[13] 将这一设定和刻画推广到了任意关系结构。我们在很大程度上借鉴了 [13], 中的框架和概念，但允许通过任意 $\exists$Datalog 程序来指定解释，因此我们的设定能够涵盖常见的数据库依赖类型，例如包含依赖。基于 $\exists$Datalog 的设定可视为对 [13] 中设定的推广（参见附录 G）。

据我们所知，这是首次研究由 Datalog 程序定义的函子。此外，这也是首次将具有右伴随的函子应用于合取查询的唯一刻画。在另一种情境下，即近似

图着色中， [22] 使用了所谓的弧图函子；此外，文中还论证了，具有右伴随的函子也能更广泛地应用于承诺约束满足问题之间归约的设计与分析。 [14]讨论了如何使用 Datalog 程序在这类问题之间进行归约。

2 预备知识

模式、实例、同态 一个模式 $\mathbf{S}$ 是关系符号的有限集合R ，每个关系符号具有指定的元数 $arity(R) \geq 0$。一个 $\mathbf{S}$-实例 I 是一个事实集合，其中事实是形如 $R(a_1, \ldots, a_n)$ 的表达式，其中 $R \in \mathbf{S}$ 且 $n = arity(R)$。除非另有说明，实例均假定为有限的。 I 的活动域 $adom(I)$ 是 I的事实中出现的所有值 a_i 的集合。一个同态 $h\colon I \to J$，其中 I 和 J 是基于同一模式$\mathbf{S}$的实例，是从 $adom(I)$ 到 $adom(J)$ 的函数，并满足：h-像下的 I 的每个事实都是 J的事实。

我们将 Inst[$\mathbf{S}$] 记作所有有限$\mathbf{S}$-实例的集合，并类似地定义 Inst$^\infty$[$\mathbf{S}$] ，但同时允许无限$\mathbf{S}$-实例。从范畴论的角度看，我们也可以将 Inst[$\mathbf{S}$] 和 Inst$^\infty$[$\mathbf{S}$]视为范畴。在这种情况下，对象是实例，箭头是同态。这将使我们能够讨论例如从 Inst[$\mathbf{S}$] 到 Inst[$\mathbf{S}'$]的函子 。

一个 k 元带点$\boldsymbol{S}$-实例（针对 $k \geq 0$）是一个有序对 $(I,\mathbf{a})$，其中 I 是一个$\mathbf{S}$-实例，而$\mathbf{a}$ 是 $adom(I)$ 中元素组成的一个 k 元组，称为指定元素。同态 $h : (I,\mathbf{a}) \to (J,\mathbf{b})$ 是一个同态 $h : I \to J$，满足 $h(\mathbf{a}) = \mathbf{b}$。

关联图、连通性、C-无环性实例 I 的关联图是一个二分多重图，其节点为 I 的元素和事实；对于事实中元素 a 的每次出现，都有一条不同的（无向）边 (a, f)。如果一个实例的关联图是连通的，就称该实例为连通；如果其实例的关联图是无环的，就称该实例为无环。带点实例 $(I,\mathbf{a})$ 是c-无环的，当且仅当 I 的关联图中的每个环都至少包含元组$\mathbf{a}$中的一个元素。

Datalog Datalog 程序由一组规则指定，它定义了从模式 $\mathbf{S}_{in}$ （传统上称为 EDB 模式）上的实例到模式 $\mathbf{S}_{out}$ （传统上称为 IDB 模式）上的实例的映射。这里给出的表述还允许存在未包含在输出模式中的辅助 IDB 关系。

▶ **定义 2.1** （Datalog 程序）。 *Datalog* 程序 是一个元组 $P = (\boldsymbol{S}_{in}, \boldsymbol{S}_{out}, \boldsymbol{S}_{aux}, \Sigma)$，其中 $\boldsymbol{S}_{in}, \boldsymbol{S}_{out}, \boldsymbol{S}_{aux}$ 是两两不相交的模式，而 Σ 是由以下形式的规则构成的集合

$$R_0(\boldsymbol{x}_0) :- R_1(\boldsymbol{x}_1), \ldots, R_n(\boldsymbol{x}_n)$$

其中 $R_0 \in \boldsymbol{S}_{out} \cup \boldsymbol{S}_{aux}$, $R_1, \ldots, R_n \in \boldsymbol{S}_{in} \cup \boldsymbol{S}_{aux}$， 且 $\{\boldsymbol{x}_0\} \subseteq \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$。

若 P 是一个 Datalog 程序，则我们通常用记号$\mathbf{S}_{in}^P$, $\mathbf{S}_{out}^P$, $\mathbf{S}_{aux}^P$和Σ^P 来指代元组 P的各个组成部分。

规则的头部 是冒号:– 符号左侧的部分，而体部 是右侧的部分。Datalog 规则 $R_0(\mathbf{x}_0) :- R_1(\mathbf{x}_1) \ldots, R_n(\mathbf{x}_n)$ 的典范实例 是一个带点实例，其活动域为$\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$，其事实是规则体中的合取项，其指定元素序列是元组$\mathbf{x}_0$。如果每条规则的典范实例都是连通的，我们就称 Datalog 程序P 是连通的 。

若 P 是一个 Datalog 程序，且 I 是一个$\mathbf{S}_{in}^P$-实例，那么对于 I，相对于 P 的一个解是模式 $\mathbf{S}_{in} \cup \mathbf{S}_{out} \cup \mathbf{S}_{aux}$ 上的一个实例 J，满足 $I \subseteq J$，并且其中所有规则

都在 J 中成立（也就是说，每当一条规则的体成立时，其头部也成立）。众所周知，追赶过程提供了一种构造解的方法：给定一个 Datalog 程序 P 和一个 $\mathbf{S}_{in}^{P}$-实例 I，我们用 $\text{chase}_P(I)$ 表示由 I 通过应用所有规则直至收敛而得到的 $\mathbf{S}_{in}^{P} \cup \mathbf{S}_{out}^{P} \cup \mathbf{S}_{aux}^{P}$-实例。更确切地说，$\text{chase}_P(I)$ 可以定义为无限并 $\bigcup_{i\geq 0} \text{chase}_P^i(I)$，其中 $chase_P^0(I) = I$，并且 $\text{chase}_P^{i+1}(I)$ 是通过使用 Σ^P 中的一条规则，从 $\text{chase}_P^i(I)$ 中的事实推导出的所有事实对 $\text{chase}_P^i(I)$ 的扩展。更多详情请参阅 [1]。

▶ **引理 2.2.** 对于所有 *Datalog* 程序 P 和 $\mathbf{S}_{in}^{P}$-实例 I，追赶 $_P(I)$ 是关于 P 的 I 的一个解。此外，它是关于 P 的 I 的所有解的交集。

我们将追赶 $_P(I)$ 的 $\mathbf{S}_{out}^{P}$-约化结构记为 $P(I)$。

所谓布尔 Datalog 程序，是指一个 Datalog 程序 P，其中 $\mathbf{S}_{out}^{P}$ 由一个零元关系符号组成，通常记作 Ans。在这种情况下，如果 $P(I) = \{Ans()\}$，则记作 $P(I) = true$；否则（即如果 $P(I) = \emptyset$），记作 $P(I) = false$。

从追赶的角度来看，我们可以将上述 $P(I)$ 的定义视为对 Datalog 程序语义的自底向上描述。展开（又称扩展）则提供了一种互补的自顶向下描述。给定一个 Datalog 程序 P，P 的可推导规则集合是满足以下条件的最小规则集：(i) 包含 P 的所有规则；(ii) 对将规则头的出现替换为相应规则体的操作封闭（必要时重命名变量）。给定一个 Datalog 程序 P 和一个关系 $R \in \mathbf{S}_{out}^{P}$，Unfoldings$(P, R)$ 是由可推导规则的典范实例组成的集合，这些规则的头部包含 R，且体中仅含 $\mathbf{S}_{in}$-关系。请注意，该集合一般是无限的。

▶ **例 2.3.** 令 P 为由以下三条规则组成的 Datalog 程序

$$R(x, y) :- S(x, y) \qquad R(x, x) :- T(x, y) \qquad T(x, y) :- U(x, y), U(y, z)$$

其中 $\mathrm{S}_{in} = \{U, S\}$、$\mathrm{S}_{out} = \{R\}$ 和 $\mathrm{S}_{aux} = \{T\}$。那么，Unfoldings(P, R)（在同构意义下）由带点实例 $(\{U(a, b)\ U(b, c)\}, \mathrm{h}a, a\mathrm{i})$ 和 $(\{S(a, b)\}, \mathrm{h}a, b\mathrm{i})$ 构成。

▶ 引理 *2.4*（参见 [12]）。对于任意 *Datalog* 程序 P、实例 $I \in Inst[S_{in}^{P}]$ 以及关于 $adom(I)$ 的 S_{out}^{P}-事实 $R(a)$，$R(a) \in P(I)$ 当且仅当存在某个 $(J, b) \in Unfoldings(P, R)$，使得 $(J, b) \to (I, a)$。

∃Datalog ∃Datalog 的语言是在 Datalog 的基础上扩展了存在量词。

▶ **定义 2.5** (∃Datalog)**.** ∃Datalog 规则 是形如以下形式的表达式

$$\exists \boldsymbol{z}\big(R_1(\boldsymbol{x}_1), \ldots, R_n(\boldsymbol{x}_n)\big) :- S_1(\boldsymbol{y}_1), \ldots, S_m(\boldsymbol{y}_m)$$

其中 $\{\boldsymbol{x}_i\} \subseteq \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n, \boldsymbol{z}\}$。在此类规则中，$\boldsymbol{z}$ 中的变量称为存在变量。如果元组 $\boldsymbol{z}$ 非空，我们也称该规则为存在规则。导出变量 是指同时出现在规则体部和头部中的变量。

一个 ∃Datalog 程序 是一个元组 $P = (\boldsymbol{S}_{in}, \boldsymbol{S}_{out}, \boldsymbol{S}_{aux}, \Sigma)$，其中 $\boldsymbol{S}_{in}, \boldsymbol{S}_{out}, \boldsymbol{S}_{aux}$ 是互不相交的模式，且 Σ 是一个 ∃*Datalog* 规则集合；其中，规则体部中出现的每个关系都来自 $\boldsymbol{S}_{in} \cup \boldsymbol{S}_{aux}$，规则头部中出现的每个关系都来自 $\boldsymbol{S}_{out} \cup \boldsymbol{S}_{aux}$。

与 Datalog 的情形一样，对于一个 ∃Datalog 程序 P，相对于该程序的 I 实例的解是模式 $\mathbf{S}_{in} \cup \mathbf{S}_{out} \cup \mathbf{S}_{aux}$ 上的一个实例 J，满足 $I \subseteq J$，并且 P 的所有规则都在 J 中成立。不过，与 Datalog 的情形不同，我们现在允许解是无限的，原因马上就会明了。对于 I（相对于 P），泛解是 I 的一个解 J，并且对于 I 的每个解 J'，都有

满足 $J \to_{adom(I)} J'$。这里，为了方便记号，我们写作 $h: I \to_X J$ 当且仅当 $h: I \to J$ 且$h(x) = x$ 对所有 $x \in X$都成立。我们也将写作 $I \leftrightarrow_X I'$ 当且仅当 $I \to_X I'$ 且 $I' \to_X I$。众所周知，每个实例 I 都有一个（可能是无限的）泛解，并且泛解在同态等价意义下唯一。更准确地说，若 J 和 J' 是同一个$\mathbf{S}_{in}$-实例 I的泛解，则 $J \leftrightarrow_{adom(I)} J'$ （参见 [10]）。

我们用记号 $P(I)$ 表示任意一个 I 的泛解的 $\mathbf{S}_{out}$-约化结构。这唯一确定了 $P(I)$，使其在 $\leftrightarrow_{adom(I)}$-等价意义下唯一。

▶ **例 2.6.** 令 P 为由以下三条规则组成的 ∃Datalog 程序

$$R(x,y) :- R_{in}(x,y) \qquad \exists z R(y,z) :- R(x,y) \qquad R_{out}(x,y) :- R(x,y).$$

实例 $I = \{R_{in}(a_1, a_2)\}$ 相对于 P 不存在有限泛解，但存在一个无限泛解，即 $J = I \cup \{R(a_i, a_{i+1})\ R_{out}(a_i, a_{i+1}) \mid i = 1, 2, \ldots\}$。

由于 $P(I)$ 通常是无限的，因此人们常施加额外限制 P，以确保 $P(I)$ 是有限的。一种著名的此类限制是 弱无环性 [16]。这里省略其精确定义（参见附录 A）。弱无环性确保有限泛解存在，并且可以使用适当版本的追赶在多项式时间内计算出来。

▶ **命题 2.7** ([16])。 设定一个 ∃*Datalog* 程序 P。如果 P 是弱无环的，那么每个有限的 $\boldsymbol{S}_{in}$-实例都有一个有限泛解，并且可以在多项式时间内计算得到。

如果一个 ∃Datalog 程序 P 满足 非递归，则称其为非递归程序；如果 $\mathbf{S}^P_{aux} = \emptyset$。每个非递归∃Datalog 程序都是弱无环的。

▶ **引理 2.8**。 设 P 为任意一个 ∃*Datalog* 程序，并设 I、I' 为 $\boldsymbol{S}^P_{in}$-实例。每个同态 $h: I \to I'$ 都可以延拓为一个同态 $h': P(I) \to P(I')$。

如果两个 ∃Datalog 程序 P、P' 分别具有 $\mathbf{S}^P_{in} = \mathbf{S}^{P'}_{in}$和 $\mathbf{S}^P_{out} = \mathbf{S}^{P'}_{out}$，则称它们是等价的；如果对于所有 $\mathbf{S}^P_{in}$-实例 I、P，都有（I） $\leftrightarrow_{adom(I)}$ P'（I）。

3 TAM Datalog

TAM Datalog 是 Datalog 的一个片段，由两个要求定义：“树形”和
“近似单元”。我们先分别介绍这两个要求。

近似单元 Datalog 程序 回忆一下，如果一个 Datalog 程序中 $\mathbf{S}_{aux}$ 的所有关系都是一元的，那么它就是单目程序。众所周知，单目 Datalog 程序可以用单子二阶逻辑（MSO）表示。形式上，我们将模式 S 上的一个 k元 MSO 查询定义为 S 上的一个单子二阶逻辑公式 $\phi(x_1, \ldots, x_k)$。我们称一个 Datalog 程序 $P = (\mathbf{S}_{in}, \mathbf{S}_{out}, \mathbf{S}_{aux}, \Sigma)$ 与一个指定的 k 元关系 $R \in \mathbf{S}_{out}$ 一起定义了 $\mathbf{S}_{in}$ 上的一个 MSO 查询 $\phi_R(x_1, \ldots, x_k)$，如果对于所有 $\mathbf{S}_{in}$ 实例 I 和 $a_1, \ldots, a_k \in adom(I)$，$R(a_1, \ldots, a_k) \in P(I)$ 当且仅当 $I \models \phi_R(a_1, \ldots, a_k)$。以下结论在数据库文献中广为人知（明确证明见 [20] ）：

▶ **定理 3.1**。 设 P 是一个单目 *Datalog* 程序，且 $R \in \boldsymbol{S}^P_{out}$。那么 (P, R) 定义了一个单子二阶逻辑查询。

现在我们将定义一种较弱的限制，即近似单元 Datalog 程序，这类程序也满足相同的性质。这些程序中，每个 k 元辅助关系在其 k 个参数位置中至多有一个指定的“铰接位置”，并且规则的语法受到约束，使得出现在非铰接位置的变量只能用于向前传递信息，而不能用于执行连接。

XX:6 Datalog 程序的右伴随，以及同态对偶

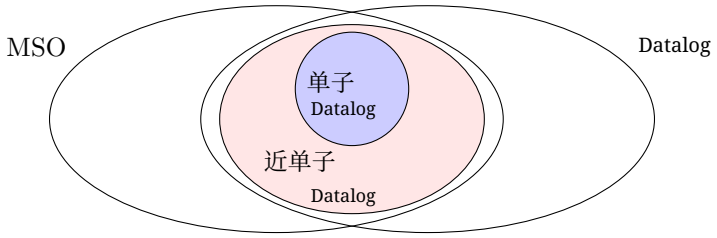


图 1 近单子 Datalog 与单子二阶逻辑和 Datalog 的关系。

▶ **定义 3.2** (近似单元 Datalog 程序)。 对于 *Datalog* 程序 P，铰接函数是一个偏函数 f，它将关系 $R \in \boldsymbol{S}^P_{aux}$ 映射到一个数$f(R) \in \{1, \ldots, arity(R)\}$，我们将其称为 R 的铰接位置 。除 $f(R)$ 之外的每个 $i \in \{1, \ldots, arity(R)\}$ 都称为 R 的非铰接位置 。如果存在一个铰接函数，使得在每条规则中，出现在规则体内辅助关系非铰接位置的每个变量在该规则体中只出现一次，且不出现在头部任何辅助关系的铰接位置上，则称该 *Datalog* 程序是近似单元 的。

注：铰接条件适用于辅助关系，而不适用于输出关系。

▶ **例 3.3**。 例 1.1 中的 Datalog 程序（该程序计算所有这样的有序对 (a, b)：存在一条从 a 到 b 的有向路径）是一个近单子 Datalog 程序：其见证铰接函数将辅助关系 R 的第一个位置指定为关节点位置。值得指出的是，如果我们用附加规则 $Ans(x, y) :- Path(y, x)$ 扩展该程序（使其计算所有这样的有序对 (a, b)：存在一条从 a 到 b 或从 b 到 a 的有向路径），所得程序仍然是近单子 Datalog 程序。这是因为对铰接函数的要求只涉及头部中的辅助关系，而不涉及输出关系。

▶**命题 3.4**。 例 *3.3* 中的近单子 *Datalog* 程序不等价于任何单子 *Datalog* 程序。

以下结果说明了“近似单元”这一术语的合理性。它表明，近单目 Datalog 程序可以在精确意义上由单目 Datalog 程序模拟。

▶**定理 3.5.** 对于每个近单目 *Datalog* 程序 P 和 k元关系符号$R \in \boldsymbol{S}^P_{out}$，都存在一个布尔单目 *Datalog* 程序 P'，其中 $\boldsymbol{S}^{P'}_{in} = \boldsymbol{S}^P_{in} \cup \{Q_1, \ldots, Q_k\}$，使得对于所有$\boldsymbol{S}^P_{in}$-实例 I和 $a_1, \ldots, a_k \in adom(I)$，以下条件等价：
1. $R(a_1, \ldots, a_k) \in P(I)$，2. $P'(I \cup \{Q_1(a_1), \ldots, Q_k(a_k)\}) = true$。
反过来，对于每个布尔单目 *Datalog* 程序 P'，其包含$\boldsymbol{S}^{P'}_{in} = \boldsymbol{S} \cup \{Q_1, \ldots, Q_k\}$，且其中每个 Q_i 都是一元的，都存在一个 *TAM Datalog* 程序 P，其包含$\boldsymbol{S}^P_{in} = \boldsymbol{S}$ 和 $\boldsymbol{S}^P_{out} = \{R\}$，使得上述等价关系成立。

由此可知，近单目 Datalog 包含于单子二阶逻辑。也就是说，对于近单目 Datalog 程序，我们有如下定理 3.1 的类似结果（参见图 1）：

▶ **推论 3.6.** 设 P是一个近单目 *Datalog* 程序，且 $R \in \boldsymbol{S}^P_{out}$。那么$(P, R)$定义了一个 *MSO* 查询。

我们不知道近单目 Datalog 是否严格包含于 MSO 与 Datalog 的交集中。不过，正如我们很快将看到的，树形 Datalog 与 MSO 的交集（在逻辑等价意义下）恰好就是树形近单子 Datalog。另请参阅 [8]，了解如何用无限约束满足问题来语义刻画 MSO 与 Datalog 的交集。另请参阅 [25]，其中介绍了另一种形式体系，它包含于 MSO 与 Datalog 的交集中。

树形性 我们称 Datalog 程序 P 为树形 的，若每条规则的关联图都是无环的。特别地，这意味着规则体中的同一个合取项里不会有变量重复出现（但规则头中可以重复出现变量）。请注意，我们不要求规则的关联图是连通的，也不对边的方向作任何要求（例如，在二元关系的情形下）。

▶ **例 3.7**。 考虑由以下两条规则给出的树形 Datalog 程序 P（其中 $\mathbf{S}_{in}$ 由两个二元关系 E, F组成）：

$$R(x,y) :- E(x,u), F(u,y) \qquad R(x,y) :- E(x,u), R(u,v), F(v,y)$$

那么， $R^{P(I)}$ 包含所有满足以下条件的有序对（a, b）：存在一条从 a到 b的有向路径，该路径先经过若干条 E边，再经过数量相同的 F边。

由关于单子二阶逻辑的已知事实（即单子二阶逻辑在单词上的表达能力涵盖正则语言）可知，(P, R) 不定义单子二阶逻辑查询。特别地， P 不等价于单目 Datalog 程序，甚至也不等价于近单目 Datalog 程序。

▶**引理 3.8**。 **</style}设 是任意树形 Datalog 程序。那么，对于每个 $\mathbf{S}_{out}^{P}$，展开(P， R) 由无环带标记实例组成。**

TAM Datalog 的*TAM Datalog* 程序 是树形的近单目 Datalog 程序。我们将用单子二阶逻辑给出 TAM Datalog 的精确模型论刻画。若对于每个带标记实例 (I， a_1，……， a_k)，都有 $I \models \phi(a_1$，……， $a_k)$ 当且仅当存在一个无环带标记实例 (J， b_1，……， b_k)，使得 $J \models \phi($ a_1，……， $a_k)$ 且 (J， b_1，……， b_k) $\rightarrow$ (I， a_1，……， a_k)，则称 MSO 查询 $\phi(x_1$，……， $x_k)$ 为树决定的 。注意， J必须是有限的，而 J 不要求连通。

▶ **定理 3.9**。 **</style}设 ($x_1, \ldots, x_n$) 是模式 $\boldsymbol{S}_{in}$ 上的一个单子二阶逻辑公式。以下条件等价：**

1. ϕ 可由 *TAM Datalog* 程序定义，**2.** ϕ 可由树形 *Datalog* 程序定义，**3.** ϕ是树决定的。

▶ 备注 3.10。值得将此结果与 [20] 中的结果进行比较；该结果指出，在有限树上，单子 Datalog 与单子二阶逻辑具有相同的表达能力。除了定理 3.9 是对任意（有限）实例的刻画，而 [20] 中的结果仅限于树之外，两者还有几个重要区别： [20], 中假设树表示为一种结构，其中每个节点的子节点都有序；签名包含用于标记根节点、叶节点、每个节点的第一个子节点和最后一个子节点的谓词；并且树中的每个节点都恰好由签名中（其他）某个一元谓词标记。这些假设共同意味着，这类树之间的每个同态必然都是同构。

▶ **推论 3.11** （TAM Datalog 在复合运算下封闭）。 对于所有满足 $\boldsymbol{S}_{in}^{P_2} = \boldsymbol{S}_{out}^{P_1}$ 的 *TAM Datalog* 程序P_1和 P_2，都存在一个 *TAM Datalog* 程序 $P_3 =$ （$\boldsymbol{S}_{in}^{P_1}, \boldsymbol{S}_{out}^{P_2}, \boldsymbol{S}'_{aux}, \Sigma'$），使得对于所有 $\boldsymbol{S}_{in}^{P_1}$-实例 I, P_3（I） $= P_2$（P_1（I））。

XX：8 Datalog 程序的右伴随与同态对偶

我们还为 TAM Datalog 程序提供了一种句法范式。如果每个规则体恰好包含一次来自 $\mathbf{S}_{in}$ 的关系，那么该 TAM Datalog 程序就是简单的 。例如，例 3.3 中给出的程序就是一个简单 TAM Datalog 程序。

▶定理 *3.12. </style}*每个（连通的）*TAM Datalog* 程序都可以在多项式时间内转换为一个等价的（连通的）简单 *TAM Datalog* 程序。

4 右伴随

回想一下，我们可以从语义上将一个 ∃Datalog 程序 P 视为从$\mathbf{S}_{in}^{P}$-实例到$\mathbf{S}_{out}^{P}$-实例的映射，并且该映射关于同态是单调的：每个同态h： $I \to I'$ 都会导出一个同态 h'： $P(I) \to P(I')$。用范畴论的语言来说，这意味着 P 是一个从$\boldsymbol{S}_{in}^{P}$-实例及其同态构成的范畴到$\boldsymbol{S}_{out}^{P}$-实例及其同态构成的范畴的函子。回想一下，对于函子 F： $X \to Y$和 G： $Y \to X$（其中 X 和 Y 是任意范畴），如果 $F(I) \to J$ 当且仅当 $I \to G(J)$，我们就称 G 是 F的右伴随，并称 F 是 G的左伴随。 [1] 本节将研究 ∃Datalog 程序的右伴随是否存在。

▶ 例 4.1。考虑 Datalog 程序 $P = (\mathbf{S}_{in}, \mathbf{S}_{out}, \emptyset, \Sigma)$，其中 $\mathbf{S}_{in} = \{R\}$、$\mathbf{S}_{out} = \{S\}$，而 Σ 由以下规则组成： $S(x,y) :- R(x,y)$ 和 $S(x,y) :- R(y,x)$。我们可以将 P 的输入实例视为有向图，并将 $P(I)$ 视为 I 的对称闭包。对于每个 $\mathbf{S}_{out}^{P}$-实例 J，令 $\Omega(J)$ 为 J 的极大对称子实例，也就是 $\mathbf{S}_{in}^{P}$-实例；具体而言，$\Omega(J)$ 由所有满足以下条件的事实 $R(x,y)$ 组成： J 同时包含 $S(x,y)$ 和 $S(y,x)$。不难看出，Ω 是 P 的右伴随。也就是说， $P(I) \to J$ 当且仅当 $I \to \Omega(J)$。

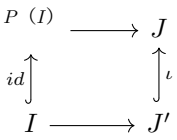
▶ 例 4.2。考虑 Datalog 程序 $P = (\mathbf{S}_{in}, \mathbf{S}_{out}, \emptyset, \Sigma)$，其中 $\mathbf{S}_{in} = \{Q_1, Q_2\}$、$\mathbf{S}_{out} = \{Q_3\}$，而 Σ 由规则 $Q_3() :- Q_1(x)\ Q_2(y)$ 组成。这个 Datalog 程序不具有上述意义下的右伴随。事实上，令 J 为一个空实例。那么，$P(I) \to J$ 成立，当且仅当 I 没有 Q_1-事实，或者 I 没有 Q_2-事实；这一条件无法等价地刻画为存在从 I 到某个固定单一实例 J' 的同态。不过，可以证明， $P(I) \to J$ 当且仅当 $I \to J_1'$ 或 $I \to J_2'$，其中 $J_1' = \{Q_1(a)\}$ 且 $J_2' = \{Q_2(a)\}$。如果推广右伴随的概念，允许 $\Omega(J)$ 是有限个实例组成的集合，那么正如我们稍后将看到的，P 确实具有右伴随。正如我们稍后将看到的，$\Omega(J)$ 由多个实例组成这一事实，与该程序包含一条关联图不连通的规则有关。

受上述例子以及其他考量的启发（这些考量将在第 6 节中变得明了），我们在此采用的右伴随的精确定义稍微更加细致：

▶定义*4.3*（广义右伴随）。设 P 为任意一个 ∃*Datalog* 程序。对于 P 的广义右伴随，我们指的是一个函数 Ω_P，它将每个 $J \in \mathit{Inst}[\mathbf{S}_{out}^{P}]$ 映射到由有序对（J'， ι）组成的有限集，其中 $J' \in \mathit{Inst}[\mathbf{S}_{in}^{P}]$ 且 ι： $adom$（J'） $\to adom$（J） 是一个偏函数，并且对于所有 $I \in \mathit{Inst}^{\infty}[\mathbf{S}_{in}^{P}]$， P（I） $\to J$ 当且仅当 $I \to J'$ 对于某个（J'， ι） $\in \Omega_P$（J） 成立，在这种

[1] 请注意，与标准范畴论定义不同，我们只关心箭头是否存在，而不要求两个箭头集合之间存在一一对应关系。

方式下，下面的图表交换：



此处，记号 $f\colon X, \to Y$ 表示 f 是一个从 X 到 Y的偏函数。

回想一下，当 P 是任意的 ∃Datalog 程序时，即使 I 是有限的， $P(I)$也可能是一个无限实例。正因如此（而且因为稍后我们会考虑 ∃Datalog 程序的复合），上述定义要求伴随算子即使在无限实例 I上也能良好运行。顺便一提，我们论文中的结果并不依赖于 J 是有限的，因此我们可以修改上述定义，使得 J， $J' \in \mathrm{Inst}^{\infty}[\mathbf{S}^{P}_{out}]$，但我们没有这样做，因为这对于我们的使用场景并非必要。

广义右伴随这一概念定义严谨，其行为也符合预期：

▶**定理 4.4**。 如果两个 ∃*Datalog* 程序都具有广义右伴随，那么它们的复合也具有广义右伴随。

本节的主要结果将表明，某些类别的 ∃Datalog 程序具有广义右伴随。在接下来的几节中，我们将应用这一结果，得出新的结论，例如关于同态对偶存在性方面的结论。

▶ 注 4.5。尽管本文特别关注右伴随，但也可以探讨一个 ∃Datalog 程序具有（广义）左伴随意味着什么。 ∃Datalog 程序的广义左伴随与查询重写密切相关，相关研究见数据集成和数据交换领域的文献。一个 Datalog 程序 P 具有广义左伴随，当且仅当 P等价于一个非递归 Datalog 程序。事实上，若 P具有广义左伴随 Θ，那么对于每个 $R \in \mathbf{S}^{P}_{out}$， $\Theta(\{R(a_1, \ldots, a_n)\})$ 中的 $\mathbf{S}^{P}_{in}$- 实例都与 Unfoldings(P, R) 的成员相对应（参见 [18, 13]）。

▶ **定理 4.6**。 每个 *TAM Datalog* 程序 P 都有一个广义右伴随 Ω_P。如果 P 是连通的，那么 $\Omega_P(J)$ 总是一个单例。此外，给定 J 和 P，$\Omega_P(J)$ 可在双指数时间内计算；当 P 的元数有界时，则可在指数时间内计算。

定理 4.6 的证明（见附录）关键性地利用了 Datalog 程序的树形性和近乎单子性。事实上，这两种性质对于广义右伴随的存在都很重要：

▶**命题 4.7**。 例 *3.7* 中的树形 *Datalog* 程序 P （它不是几乎单子式的）不具有广义右伴随。

▶ **命题 4.8**。 由单条规则 $Ans(x) :\!- E(x, x)$给出的单目 *Datalog* 程序不具有广义右伴随。

对于布尔非递归程序，定理 4.6 也有一个逆命题（另见 [18,定理 2.5]）。下述定理可由我们将在第 6 节证明的结果推出（具体而言，是定理 6.4 与定理 6.3 结合所得）：

▶ **定理 4.9.** 对于布尔非递归 *Datalog* 程序 P，以下条件等价：**1.** P 存在广义右伴随，**2.** P 等价于一个 *TAM Datalog* 程序。

XX:10 Datalog 程序的右伴随，以及同态对偶

接下来，我们确定另一类存在广义右伴随的 ∃Datalog 程序。若一个 ∃Datalog 程序满足每条规则的体都由单个原子组成（在$\mathbf{S}_{in} \cup \mathbf{S}_{aux}$上），且不重复使用变量，则称其为强线性 程序。强线性 ∃Datalog 规则也称为*LAV*（“局部即视图”）约束。需要注意的是，强线性比仅要求每条规则体至多包含一个$\mathbf{S}_{aux}$-原子的限制更严格（这种性质通常称为线性）。例如，例 1.1 中的程序就不是强线性的。显然，每个强线性程序都是树形的。

▶**例 4.10**。 考虑以下 ∃Datalog 程序 $P = (\mathbf{S}_{in}, \mathbf{S}_{out}, \mathbf{S}_{aux}, \Sigma)$，其中 $\mathbf{S}_{in} = \{R_{in}\}$、$\mathbf{S}_{out} = \{R_{out}\}$、$\mathbf{S}_{aux} = \{R\}$，且 Σ 由以下三条规则构成

$$R(x,y,z) :- R_{in}(x,y,z) \qquad \exists uv\ R(y,u,v) :- R(x,y,z) \qquad R_{out}(x,y,z) :- R(x,y,z)$$

一个 $\mathbf{S}_{in}$-实例是一个仅包含单个三元关系的数据库实例。我们可以认为 P 执行两项操作：将输入数据复制到输出，并添加额外事实，以确保满足包含依赖 $\forall xyz(R(x,y,z) \rightarrow \exists uv(R(y,u,v)))$。该程序 P是强线性的。注意， P是递归的，但不是弱无环的。事实上，即使 I 是有限的， $P(I)$ 也可能是一个无限实例。

▶**定理 4.11**。 每个强线性的 ∃*Datalog* 程序 P 都有一个广义右伴随Ω_P。此外，对于每个 $\boldsymbol{S}_{out}$-实例 J，$\Omega_P(J)$ 是一个单元素集合，并且可以根据 J和 P在指数时间内计算得出。如果 P固定，则可以在多项式时间内计算 $\Omega_P(J)$。

由单条非递归规则构成的强线性 ∃Datalog 程序的右伴随，最初以不同的术语发表于 [23]。

基于上述结果，我们可以定义一个更大的 ∃Datalog 片段，它具有广义右伴随：每个可表示为 TAM Datalog 程序与强线性 ∃Datalog 程序按任意顺序进行有限复合的 ∃ Datalog 程序，都具有广义右伴随。我们可以给出该语言的句法定义（使用分层），但此处不作介绍。

5 间奏：元组生成依赖

一个元组生成依赖（*TGD*）是一阶句子，其形式为

$$\forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow \exists \mathbf{y}\psi(\mathbf{x},\mathbf{y}))$$

其中 $\phi(\mathbf{x})$ 和 $\psi(\mathbf{x},\ \mathbf{y})$ 是关系原子公式的合取。元组生成依赖可以表达各种各样的约束，包括包含依赖等数据库依赖。

每个元组生成依赖的有限集自然会导出一个 ∃Datalog 程序。更确切地说，对于模式 S 上的任意元组生成依赖集 Σ，我们将由 Σ 中的元组生成依赖作为 ∃Datalog 规则构成的 ∃Datalog 程序记为 P_Σ，其中 $\mathbf{S}_{in}^P = \{R_{in} \mid R \in \mathbf{S}\}$、$\mathbf{S}_{out}^P = \{R_{out} \mid R \in \mathbf{S}\}$ 且 $\mathbf{S}_{aux}^P = \mathbf{S}$（其中 $\forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow \exists \mathbf{y}\psi(\mathbf{x},\mathbf{y}))$ 转换为 $\exists \mathbf{y}\psi(\mathbf{x},\mathbf{y}) :- \phi(\mathbf{x})$），并且还包括针对每个 $R \in \mathbf{S}$ 的“复制约束” $R(\mathbf{x}) :- R_{in}(\mathbf{x})$ 和 $R_{out}(\mathbf{x}) :- R(\mathbf{x})$。

尽管 P_Σ 的输入模式和输出模式都是 $\mathbf{S}$ 的重命名形式，但我们会略去细节，即使 I 是一个 $\mathbf{S}$ 实例，也写作 $P(I)$，并理解为关系符号已按照显然的方式进行了重命名；类似地，我们将 $P(I)$ 视为一个 $\mathbf{S}$ 实例。

∃Datalog 程序 P_Σ 在以下意义下“捕获”了 Σ：

▶**引理 5.1**。 设 Σ 为任意有限集的元组生成依赖。则：

1. 对于所有 $\boldsymbol{S}$-实例 I， $I \subseteq P_\Sigma(I)$ 和 $P_\Sigma(I) \models \Sigma$。

2. 对于所有 $\mathcal{S}$-实例 $I \models \Sigma$，$P_\Sigma(I) \to_{adom(I)} I$。

对于 Datalog 程序的任意性质 X（例如树形、弱无环或具有广义右伴随），如果有限集的元组生成依赖 Σ 具有性质 X，那么 P_Σ 也具有该性质。

接下来，我们将给出一些具有广义右伴随的元组生成依赖集合的示例。我们将在后续章节中看到这一性质的重要性。

▶ **例 5.2.** 以下元组生成依赖集合具有广义右伴随：

- $\Sigma_1 = \{\forall xyz(R(x,\ y) \wedge R(y,\ z) \to R(x,\ z))\}$。要看出 P_{Σ_1} 具有广义右伴随，只需注意它由左侧所示的规则组成：

$$R(x,y) :- \quad R_{in}(x,y)\ R(x,y) :- \quad R_{in}(x,y) R(x,z) :- \quad R(x,y)\ R(y,z)\ R(x,z) :- \quad R(x,y)\ R_{in}(y,z) R_{out}(x,y) :- \quad R(x,y)\ R_{out}(x,y) :- \quad R(x,y)$$

  P_{Σ_1} 既不是 TAM Datalog 程序，也不是强线性 $\exists$Datalog 程序。不过，它等价于程序 P'，该程序由右侧所示的规则组成。请注意，我们将 R 的一处替换成了 R_{in}。不难证明，P_{Σ_1} 与 P' 等价。此外，P' 是一个 TAM Datalog 程序（其中 R 的关节点位置是第二个位置）。由于 P_{Σ_1} 等价于 P'，且 P' 具有广义右伴随，因此 P_{Σ_1} 也具有广义右伴随（事实上，它具有相同的广义右伴随）。

- $\Sigma_2 = \{\forall xy(R(x,\ y) \to \exists z(R(y,\ z))\}$。要看出 P_{Σ_2} 具有广义右伴随，只需注意到它是一个强线性的 $\exists$Datalog 程序。事实上，对于仅由包含依赖组成的任意一组 Σ，这一点同样成立。

- $\Sigma_3 = \Sigma_1 \cup \Sigma_2$。虽然 Σ_3 既不是强线性的，也不等价于 TAM Datalog 程序，但它等价于两个 $\exists$Datalog 程序的复合，即 P_{Σ_2} 和P_{Σ_1}。要看出这一点，请注意，只要 $I \models \Sigma_2$，那么 $P_{\Sigma_1}(I) \models \Sigma_2$ 也成立。（巧合的是，此处执行复合的顺序很重要：如果 $I \models \Sigma_1$，一般并不能推出 $P_{\Sigma_2}(I) \models \Sigma_1$！）由于 P_{Σ_1} 和 P_{Σ_2} 各自都有广义右伴随，它们的复合也有广义右伴随。

- $\Sigma_4 = \{\forall xyzu(R(x,\ y) \wedge R(y,\ z) \wedge R(z,\ u) \to R(x,\ u))\ \forall xy(R(x,\ y) \to R(y,\ x))\}$。与 Σ_1的情形一样，尽管 P_{Σ_4} 不是 TAM Datalog 程序，但很容易将其改写为一个 TAM Datalog 程序。我们将在稍后的注 6.9 中回到这个例子。

- 若规则头部中的关系是单目的，我们就称该 TGD 是单目的。那么，对于每个由单目树形 TGD 组成的有限集 Σ，P_Σ 都是一个 TAM Datalog 程序。

▶ 注 5.3。上述大多数示例都涉及临时论证。我们将定义一个大型的、具有广义右伴随的元组生成依赖（集合）句法类作为一个开放问题；该句法类包含 Σ_1。定理 4.6 与定理 3.9 确实表明，对于由树形元组生成依赖组成的有限集 Σ，如果 P_Σ 可由单子二阶逻辑定义，那么 Σ 具有广义右伴随。

6 同态对偶

对于任意实例集 X，令 $X\uparrow = \{A \mid B \to A$ 对某个 $B \in X\}$成立，并令 $X\downarrow = \{A \mid A \to B$ 对某个 $B \in X\}$成立。同态对偶是实例集 (F, D) 构成的一对，使得$F\uparrow$ 是 $D\downarrow$的补。同一定义自然地扩展到带标记实例。有限同态对偶是指满足以下条件的同态对偶 (F, D)：F 和D 都是有限集。树对偶是指满足以下条件的同态对偶 (F, D)：F 是一个（可能无限的）无环实例集（不一定连通），且 D 是有限集。

XX:12 Datalog 程序的右伴随，以及同态对偶性

对偶性的研究起源于组合数学（参见 [21]），其动机是对偶性与同态偏序结构以及判定图之间、更一般地说关系结构之间是否存在同态（又称约束满足问题或 CSPs）的复杂度之间的联系。事实上，对偶性在约束满足问题的研究中发挥了重要作用。特别是，研究 [3] 表明，可由一阶逻辑定义的约束满足问题，恰好是模板位于有限对偶右侧的那些问题。同样地，可由著名的弧一致性算法求解的约束满足问题，恰好是模板位于树对偶右侧的那些问题。更一般地，可由局部一致性方法求解的约束满足问题，其模板位于某种同态对偶的右侧，且该对偶的左侧由树宽有界的实例构成。关于对偶性与一致性算法之间联系的综述，参见 [9] 。

▶ **例 6.1.** 设 $\mathbf{S} = \{R\}$，其中 R 是一个二元关系符号，并设 $n \geq 1$。令 L_n为长度为 n的有限线性序，令 P_{n+1}为长度为 $n+1$的有向路径。则 $(\{P_{n+1}\}, \{L_n\})$ 是一个有限同态对偶。

▶ **例 6.2.** 设 $\mathbf{S} = \{P_0, P_1, E\}$，其中 P_0 和 P_1 是一元关系符号，E 是二元关系符号，并考虑由两个元素构成的 $\mathbf{S}$ 实例 $I = \{P_0(0)\ P_1(1)\ E(0,0)\ E(1,1)\}$（不含指定元素）。容易看出，对于所有 $\mathbf{S}$ 实例 J，$J \to I$ 成立，当且仅当 J 的每个连通分量中都不同时包含一个 P_0-事实和一个 P_1-事实。这可以表示为树对偶的形式：令 F 为所有（无环）实例构成的集合，其中每个实例都包含一条连接 P_0-节点与 P_1-节点的定向路径。那么 $(F, \{I\})$ 是一个同态对偶。

▶ **定理 6.3** ([17, 11])。 固定一个模式$\boldsymbol{S}$ 和 $k \geq 0$。设 F 是 k 元带标记实例的任意有限集，且其中任意两个实例在同态意义下不可比，并且这些实例都定义在$\boldsymbol{S}$上。以下条件等价：
1. 存在一个有限集，其中的实例均为 k 元带标记实例 D，且均定义在 $\boldsymbol{S}$ 上，使得 (F, D) 构成一个同态对偶。2. 中的每个带标记实例 F 都与某个 $\boldsymbol{c}$-无环带标记实例同态等价。此外（对于固定的 $\boldsymbol{S}$ 和 k），给定一个由 $\boldsymbol{c}$-无环带标记实例组成的集合 F，可以在指数时间内计算出这样的集合 D 。

指数时间上界在 [11] 中并未明确说明，但可由该文中的结果推出。

通过伴随构造对偶 以下定理揭示了广义右伴随与同态对偶之间的密切关系。具体而言，该定理表明，广义右伴随可用于构造对偶。这种方法最早见于 [18] 和 [13]，其中通过应用右伴随推导出树的对偶。

▶**定理 6.4**。 设 P 是任意具有广义右伴随的 *Datalog* 程序。那么，对于每个 $R \in \boldsymbol{S}^P_{out}$，存在一个有限的带指定元素的 $\boldsymbol{S}_{in}$-实例集合 D，使得 (展开(P, R) D)构成一个同态对偶。

证明。我们不妨设 $\mathbf{S}^P_{out} = \{R\}$。令 J 为这样的 $\mathbf{S}^P_{out}$-实例：其中 $adom(J) = \{b_1, \ldots, b_k, c\}$（对于 $k = arity(R)$），并且包含关于 $adom(J)$ 的所有 R-事实，但不包含 $R(b_1, \ldots, b_k)$。令 $D = \{(J', \mathbf{b}') \mid (J', \iota) \in \Omega_P(J),\ \mathbf{b}' \in adom(J')^k,$ 以及 $\iota(\mathbf{b}') = \mathbf{b}\}$，其中 $\mathbf{b} = b_1, \ldots, b_k$。我们断言，(展开$(P, R)$ D) 是一个同态对偶。设 $(C, \mathbf{c})$ 是任意具有 k 个指定元素的 $\mathbf{S}_{in}$-实例。那么，展开(P, R) 中的一个实例能够同态映射到 $(C, \mathbf{c})$，当且仅当 $R(\mathbf{c}) \in P(C)$；当且仅当 $(P(C), \mathbf{c}) \not\to (J, \mathbf{b})$；根据伴随性质，当且仅当对于所有 $(J', \iota) \in \Omega_P(J)$ 且 $\mathbf{b}'$ 满足 $\iota(\mathbf{b}') = \mathbf{b}$，有 $(C, \mathbf{c}) \not\to (J', \mathbf{b}')$。◀

特别是，这意味着，对于每个 TAM Datalog 程序 P 以及每个 $R \in \mathbf{S}^P_{out}$，都存在一个有限指向实例集 D，使得 (展开(P, R) D) 是一种同态

对偶。由于当 P 是 TAM Datalog 程序时，展开(P, R) 由无环实例构成，因此这为我们提供了一种系统地构造树对偶的方法。

注意，在特殊情况下，如果 P 是一个 TAM Datalog 程序，或一个强线性 ∃Datalog 程序，那么定理 6.4 的证明给出了一个指数时间算法，可在关系的元数有界时，根据 $(P,\ R)$ 计算 $D\ P$。

事实上，以下定理表明，每一种树对偶性都可以通过这种方式得到。

▶**推论 6.5**。设 F 是任意一组无环带标记实例。以下两者等价：**1.** 存在一个有限指向实例集 D 使得 $F(\ D,\ D)$ 是一个同态对偶**2.** $F\uparrow =$ 展开$(P,\ R)\uparrow$ 对于某个 *TAM Datalog* 程序 P 以及 $R \in \boldsymbol{S}^{P}_{out}$。

证明。从 1 推出 2：众所周知，对于任意有限指向实例集 D，都存在一个定义 $D\downarrow$ 的单子二阶逻辑公式 ϕ。因此，根据对偶性，$\neg\phi$ 定义 $F\uparrow$。此外，F 由无环带标记实例组成这一事实意味着 $\neg\phi$ 是树决定的。因此，根据定理 3.9，可得方向 1 到 2。方向 2 到 1 则由定理 6.4 得出。◀

通过证明上述条件 (1) 和 (2) 依次等价于以下事实，可以加强推论 6.5：$F\uparrow = G\uparrow$ 对于某个无环查询的正则集合 G（其中“正则”需要以适当的方式定义，参见 [15]）。这是因为定理 3.9 在证明中将树自动机作为中间步骤。我们注意到，对于有向图上的布尔合取查询，这一等价关系的特殊情形已在 [15] 中得到证明。

定理 6.4 还意味着，每个有限的无环带标记实例集合 F 都是某个有限同态对偶的左侧：只需令 P 为 TAM Datalog 程序，其中每个 $(I, \mathbf{a}) \in F$ 对应一条非递归规则，且该规则的典范实例为 $(I, \mathbf{a})$。那么，P 的展开在同构意义下恰好是 F 中的带标记实例。由定理 6.4 可知，存在一个有限集 D，使得 (F, D) 是一个同态对偶。

▶ 注记 6.6。回顾一下，当且仅当一个带点实例是 c-无环的（在同态等价意义下）时，它才是有限同态对偶的左侧（定理 6.3）。鉴于此，自然会问：上述“通过伴随构造对偶性”的技术是否可用于为任意 c-无环带点实例构造一个有限同态对偶。命题 4.8 表明这是不可能的。注意，命题 4.8 中程序的规则的典范实例为 $(\{E(a, a)\}\ a)$，它是 c-无环的（但不是无环的）。

相对于背景理论的同态对偶 在许多情形下，人们会关注关系结构的受限类，例如线性序、等价关系、满足给定完整性约束的数据库实例、知识库模型等。本节考察这样一个问题：在满足给定背景理论的所有实例所构成的范畴中，何时存在同态对偶。

已有一些相关结果。无向图可以看作模式 S 上的一个实例，其中 S 由单个二元关系符号 E 构成，并满足元组生成依赖 $\forall xy(E(x, y) \rightarrow E(y, x))$ 和 $\forall x\neg E(x, x)$。已知，无向图与同态构成的范畴，除平凡对偶 $(\{K_2\}, \{K_1\})$ 外，不存在其他有限对偶（在同态等价意义下）；其中 K_1 和 K_2 分别是含两个元素的团和空图（参见 [21]）。类似地，相似类型为 σ 的有限代数可以看作一个 S-实例，其中 S $=\{R_f \mid f \in \sigma\}$ 满足 $\Sigma = \{\forall \mathbf{x} \exists y R_f(\mathbf{x}\ y)\ \forall \mathbf{x} yz(R_f(\mathbf{x}\ y) \wedge R_f(\mathbf{x}\ z) \rightarrow y = z) \mid f \in \sigma\}$；并且已知，在有限代数范畴中，不存在非平凡的有限对偶 [5]。

下一个结果展示了如何从由具有广义右伴随的弱无环 TGD 集定义的类中，得到有限同态对偶。

XX：14 Datalog 程序的右伴随与同态对偶

▶**定理 6.7**。 设 Σ 是一个具有广义右伴随的有限弱无环 *TGD* 集。设 F 是任意有限带标记实例集。若 F 的每个成员都具有$(P_\Sigma(A),\boldsymbol{a})$ 的形式，其中$(A, \boldsymbol{a})$是某个 *c*-无环带标记实例，那么 F相对于 Σ具有有限对偶。

关于复杂度，考虑这样的情形： Σ 是一个固定的元组生成依赖集合（不作为输入的一部分处理），且 P_Σ 等价于一个 TAM Datalog 程序。那么，根据定理 6.7，可以用 2ExpTime 算法从 F计算对偶集 D，前提是 F 由底层的 c-无环结构集合（A，$\mathbf{a}$）指定。因此，例如，对于传递有向图类（如我们之前所见，该类可由 TAM Datalog 程序刻画），我们可以用 2ExpTime 算法为那些被指定为无环有向图的传递闭包的有向图构造对偶。当 Σ 是弱无环的包含依赖集合时，或者更一般地，当 P_Σ 等价于一个弱无环的强线性 Datalog 程序时，结论也同样成立。事实上，在后一种情形下，我们可以得到一个指数时间上界。

对于单目树形 TGD 的特殊情形，我们可以证明一个逆命题：

▶**定理 6.8**。 设 Σ 是任意一组树形单目 *TGD*。设 F 是任意有限个两两不同态不可比的带标记实例$(A,\boldsymbol{a})$，且满足 $A \models \Sigma$。那么，以下条件等价：

1. F 相对于 Σ 具有有限对偶，2. 每个 $(A,\ \boldsymbol{a}) \in F$都与某个 $(P_\Sigma(A'),\ \boldsymbol{a})$ 同态等价，其中 $(A',\ \boldsymbol{a})$ 是 *c*- 无环的。

▶ 注 6.9。定理 6.8 无法推广到任意有限的、可取广义右伴随的弱无环元组生成依赖集。考虑一组弱无环元组生成依赖（定义在仅含一个二元关系的模式上） $\Sigma = \{\forall xyzu(R(x,y) \wedge R(y,z) \wedge R(z,u) \rightarrow R(x,u))\ \forall xy(R(x,y) \rightarrow R(y,x))\}$。正如我们在例 5.2 中指出的， P_Σ 等价于一个 TAM Datalog 程序，因此 Σ 具有广义右伴随。设 A 为实例（不含指定元素）$\{R(a,a)\}$，设 B为实例$\{R(a,b)\ R(b,a)\}$。那么，$(\{A\},\{B\})$ 是相对于 Σ的一个同态对偶。事实上，设 C 为一个满足 Σ的实例，并假设 $A \not\rightarrow C$（即 C 没有自环）。由于 C 满足 Σ，因此 C 没有奇圈，故可同态映射到 B。然而，容易看出，任何满足 $P_\Sigma(A') = A$ 的实例A' 都必定含有一个环。

ABox 范畴中的对偶固定一个有限的元组生成依赖集 Σ。直观而言，$ABox$ 是一个有限数据库实例 I，它被视为不完整数据库，其完备化为 $P_\Sigma(I)$（参见 [4]）。我们不假设 Σ 是弱无环的。因此，可以将 ABox I 看作可能无限的实例 $P_\Sigma(I)$ 的有限表示。

虽然在规格说明层面，ABox 只是一个有限实例，但它自然带有另一种态射类型，用以刻画其预期语义：若 h 是从 $adom(I)$ 到 $adom(J)$ 的偏函数，且可以扩展为同态 $h' : P_\Sigma(I) \rightarrow P_\Sigma(J)$，则记作 $h : I \rightarrow_\Sigma J$。

我们用 $ABox_\Sigma[\mathbf{S}]$ 表示 ABox 范畴，其中的态射是上述所述的 $\rightarrow_\Sigma$-态射。我们关注这个范畴，是因为它在知识表示，尤其是描述逻辑中发挥着基础作用。定理 6.7 的证明还蕴含了以下关于范畴 $ABox_\Sigma[\mathbf{S}]$ 中对偶的结果（其复杂度界类似）：

▶**定理 6.10**。 设 Σ 是任意一个具有广义右伴随的有限元组生成依赖集。每个 F *c*-无环带基点 *ABox* 的有限集在范畴 $ABox_\Sigma[\boldsymbol{S}]$ 中都具有有限对偶性。

7 应用：唯一可刻画的并合连言查询

在本节中，我们将展示同态对偶研究结果的一项应用。已有研究表明，[17, 11] 每个 c-无环合取查询的并（UCQ）都可以由一个有限的带标签示例集合唯一刻画。事实上，一个 UCQ q 可以由一个有限的带标签示例集合唯一刻画，当且仅当 q 等价于某个 c-无环合取查询的并。在本节中，我们研究对于由有限弱无环 TGD 集定义的受限类，是否也成立同样的结论。

我们假设读者熟悉合取查询的并（UCQ）的定义（参见附录 A）。如果每个合取查询的典范（带点）实例都是 c-无环的，我们就称合取查询的并 q *c*-无环 。

设 S 是一个模式，Σ 是 S 上的一阶理论。若对于所有满足 $I \models \Sigma$, $q(I) = q'(I)$ 的 $\in$ I nst[S]，两个（模式 S 上的）合取查询的并 q, q' 都是等价的，我们称它们关于 Σ 等价。对于一个 k元查询，带标签示例的集合是指一对 (E^+, E^-)，其中包含指定元素为 k 的有限个带标记实例。若对于所有 $(A, \mathrm{a}) \in E^+$，均有一个 $\in q(A)$，且对于所有 $(A, \mathrm{a}) \in E^-$，均有一个 $\notin q(A)$，则称合取查询的并 q 拟合此类 (E^+, E^-)。若合取查询的并 q 拟合带标签示例集合 (E^+, E^-)，且每个拟合 (E^+, E^-) 的合取查询的并都关于 Σ等价于 q，则称该集合关于 Σ唯一刻画 q。

▶**定理 7.1**。 设 S 是一个模式，Σ 是一个具有广义右伴随的有限弱无环 *TGD* 集。那么，每个定义在 S 上的 *c*-无环合取查询 q，都能相对于 Σ 由满足 Σ的有限带标签示例集合唯一刻画。

受知识表示中的应用场景启发（参见 [19]），我们还给出了定理 7.1 针对 ABox 示例的一个变体。回顾一下，在存在一组元组生成依赖 Σ的情况下，ABox 是一个（有限）实例，它不一定满足 Σ，但被视为可能的无限实例 P_Σ的有限描述I。从这个角度来看，ABox 示例只是一个带点实例（A，a）。如果对于所有（A，a）$\in E^+$，a $\in q$（P_Σ（A））， 且对于所有（A，a）$\in E^-$，a $\notin q$（P_Σ（A））， 我们就称合取查询的并 q 拟合相对于 Σ的一组带标签 ABox 示例（E^+，E^-）。如果合取查询的并 q拟合（E^+，E^-），并且每个拟合（E^+，E^-）的合取查询的并 q'都与 q相对于 Σ等价，那么一组带标签 ABox 示例（E^+，E^-）就唯一刻画了合取查询的并 q。

这里的“关于 Σ 等价”这一表述需要进一步讨论，因为前文定义该概念时仅涉及有限实例，而在当前语境下，考虑有限实例和无限实例更为自然；或者至少考虑所有形如 $P_\Sigma(I)$、其中 I 是某个有限实例的实例。为简化表述并避免混淆，我们将讨论范围限制在具有有限可控性 性质的元组生成依赖集 Σ 上。这意味着，对于所有合取查询的并 q, q'，在满足 Σ 的有限实例上，q 与 q' 等价，当且仅当在满足 Σ 的所有（有限和无限）实例上，q 与 q' 等价。这意味着前面提到的三种等价概念完全一致。例如，已知所有有限的强线性元组生成依赖集（特别是包含依赖）都是有限可控的 [24, 6]。

▶**定理 7.2.** 设S 是一个模式，且 Σ 是具有广义右伴随并且有限可控的有限元组生成依赖集。那么，S 上的每个 *c*-无环合取查询 q 都能由有限个带标签 *ABox* 示例唯一刻画（关于 Σ）。

请注意，与定理 7.1 不同，这一结果不要求弱无环性。特别是，当 Σ 是一组强线性元组生成依赖时，该结果适用。

8 结论

我们引入了 Datalog 的一个新片段 TAM Datalog。该片段在语义上性质良好（在复合下封闭且具有自然的语义刻画），并允许广义右伴-

随。我们还证明，强线性 ∃Datalog 程序允许广义右伴随。我们利用这些结果，获得了在背景理论下构造新的同态对偶的方法，并进一步获得了在背景理论下为合取查询的并构造唯一刻画的方法（回答了 [11] 中提出的一个开放问题）。

为了说明后者，考虑由 C 传递有向图组成的类。由我们的结果可知，每个 c-无环合取查询 q 相对于 C都存在唯一刻画，即存在一个来自 C 的带标签示例有限集，能够唯一刻画 q 相对于 C。此外，这些示例可以在双指数时间内构造出来。类似地，如果我们考虑满足包含依赖 $\forall xy(R(x,\ y) \rightarrow \exists z R(y,\ z)$ 的有向图类，那么每个 c-无环合取查询都存在一个唯一刻画它的 ABox 示例集，并且该集合可以在指数时间内计算出来。

我们将以下问题留待未来研究：（i）确定一种语法判据，以保证给定的元组生成依赖集存在广义右伴随；（ii）将我们的结果扩展到包含等值生成依赖的情形；（iii）为从 TAM Datalog 程序构造同态对偶这一任务获得紧确的复杂度界。

参考文献

1 Serge Abiteboul、Richard Hull 和 Victor Vianu。《数据库基础》。Addison Wesley，1995年。 2 Bogdan Alexe、Balder ten Cate、Phokion G. Kolaitis 和 Wang-Chiew Tan。通过数据示例刻画模式映射。《ACM 数据库系统汇刊》，36(4):23:1–23:48，2011年12月。URL: http://doi.acm.org/10.1145/2043652.2043656, doi:10.1145/2043652.2043656。 3 Albert Atserias。论有向图着色问题与树宽对偶。《欧洲组合学杂志》，29(4):796–820，2008年。 doi:10.1016/j.ejc.2007.11.004。 4 Franz Baader 和 Werner Nutt。《描述逻辑基础》，第43–95页。剑桥大学出版社，美国，2003年。 5 Richard N. Ball、Jaroslav Nešetřil 和 Aleš Pultr。满同态中的对偶性。《欧洲组合学杂志》，31(1):106–119，2010年。URL: https://www.sciencedirect.com/science/article/pii/S019566980900122X, doi:10.1016/j.ejc.2009.04.004。 6 Vince Bárány、Georg Gottlob 和 Martin Otto。查询守卫片段。《计算机科学中的逻辑方法》，10(2)，2014年。 doi:10.2168/LMCS-10(2:3)2014。 7 Michael Benedikt、Pierre Bourhis、Balder Ten Cate、Gabrieled Puppis 和 Michael Vanden Boom。根据可见信息和背景知识进行推理。《ACM 计算逻辑汇刊》，22(2)，2021年6月。 doi:10.1145/3452919。 8 Manuel Bodirsky、Simon Knäuer 和 Sebastian Rudolph。单目与守卫二阶逻辑的 Datalog可表达性。载于 ICALP 2021，第120:1–120:17页，2021年。 9 Andrei A. Bulatov、Andrei A. Krokhin 和 Benoît Larose。约束满足问题的对偶性。载于 Nadia Creignou、Phokion G. Kolaitis 和 Heribert Vollmer 编，《约束的复杂性——当前研究主题概览》 /达格施图尔研讨会成果/，计算机科学讲义第5250卷，第93–124页。施普林格，2008年。 doi:10.1007/978-3-540-92800-3_5。 10 Andrea Calì、Georg Gottlob 和 Thomas Lukasiewicz。一个用于本体上易处理查询回答的通用基于Datalog的框架。《网络语义学杂志》，14:57–83，2012年。专题：应对数据网络的混乱。URL: https://www.sciencedirect.com/science/article/pii/S1570826812000388, doi:10.1016/j.websem.2012.03.001。 11 Balder Ten Cate 和 Victor Dalmau。合取查询：唯一刻画与精确可学习性。《ACM 数据库系统汇刊》，47(4)，2022年11月。 doi:10.1145/3559756。 12 Surajit Chaudhuri 和 Moshe Y Vardi。论递归与非递归Datalog程序的等价性。《计算机与系统科学杂志》，54(1):61–78，

1997年。URL: https://www.sciencedirect.com/science/article/pii/S0022000097914525, doi:10.1006/jcss.1997.1452。 13 Victor Dalmau、Andrei Krokhin 和 Jakub Opršal。关系结构上同时具有左伴随和右伴随的函子。准备中，2023年。 14 Victor Dalmau 和 Jakub Opršal。作为约束满足问题之间的一种归约的局部一致性，2023年。 arXiv:2301.05084。 15 Péter L. Erdös、Dömötör Pálvölgyi、Claude Tardif 和 Gábor Tardos。关系结构的正则森林族、反链和对偶对。《组合数学》，37(4):651–672，2017年。doi:10.1007/s00493-015-3003-4。 16 R. Fagin、P. G. Kolaitis、R. J. Miller 和 L. Popa。数据交换：语义与查询回答。载于国际数据库理论会议（ICDT），第207–224页，2003年。 17 Jan Foniok、Jaroslav Nešetřil 和 Claude Tardif。关系结构同态序中的广义对偶性与极大有限反链。《欧洲组合学杂志》，29(4):881–899，2008年。 18 Jan Foniok 和 Claude Tardif。同时具有左伴随和右伴随的有向图函子。《离散数学》，338(4):527–535，2015年。URL: https://www.sciencedirect.com/science/article/pii/S0012365X14003938, doi:10.1016/j.disc.2014.10.018。 19 Maurice Funk、Jean Christoph Jung 和 Carsten Lutz。DL-Lite 本体下 ELI 查询的前沿与精确学习。载于 Lud De Raedt 编，《第三十一届国际人工智能联合会议论文集》（IJCAI-22），第2627–2633页。国际人工智能联合会议组织，2022年7月。主会场。doi:10.24963/ijcai.2022/364。 20 Georg Gottlob 和 Christoph Koch。单子 Datalog 与网络信息抽取语言的表达能力。《ACM 杂志》，51(1):74–113，2004年1月。 doi:10.1145/962446.962450。 21 Pavol Hell 和 Jaroslav Nešetřil。《图与同态》，牛津数学及其应用讲座系列第28卷。牛津大学出版社，2004年。 22 Andrei A. Krokhin、Jakub Opršal、Marcin Wrochna 和 Stanislav Živný。承诺约束满足中的拓扑与伴随，2020年。arXiv:2003.11351。 23 Aleš Pultr。关系系统范畴中的右伴随。载于《中西部范畴研讨会报告 IV》，数学讲义第137卷，第100–113页。施普林格，1970年。 24 Riccardo Rosati。开放世界假设下数据库中合取查询回答的有限可控性。《计算机与系统科学杂志》，77(3):572–594，2011年。数据库理论。URL: https://www.sciencedirect.com/science/article/pii/S002200001000053X, doi:10.1016/j.jcss.2010.04.011。 25 Sebastian Rudolph 和 Markus Krötzsch。标记与检查：使用单子定义查询的数据访问。载于《第32届 ACM SIGMOD-SIGACT-SIGAI 数据库系统原理研讨会论文集》，PODS '13，第151–162页，美国纽约州纽约市，2013年。美国计算机协会。doi:10.1145/2463664.2465227。

更多预备知识

弱无环性 ∃Datalog 程序的依赖图 是一个有向图，其节点是所有形如（R，i）的二元组，其中 $R \in \mathbf{S}_{aux}$ 和 $i \in \{1$、.....、$arity$（R）$\}$。该图有两种类型的边：

1. 如果在规则体中的一个 R-原子的位置i 上出现某个变量，并且该变量也出现在规则头中的一个 S-原子的位置 j 上，则存在一条从 (R, i) 到 (S, j) 的“普通”边。2. 如果 R 出现在规则体中，并且规则头中存在 S 的一次出现，其第 j 个位置上有一个存在变量，则存在一条从 (R, i) 到 (S, j) 的“特殊”边。

如果一个 ∃Datalog 程序的依赖图中不包含经过特殊边的有向环，则称其为弱无环。

（合取查询的并）对于S 这一模式以及 $k \geq 0$， k元合取查询（*CQ*）（定义在$\boldsymbol{S}$ 上）具有如下形式：

$$\phi(y_1, \ldots, y_k) :- \exists \mathbf{x}(\phi_1 \wedge \cdots \wedge \phi_n) \qquad \text{(Eq. 1)}$$

其中，每个 ϕ_i 都是一个关系原子公式，并且每个变量 y_i 都至少出现在一个合取项 ϕ_j 中。 k元合取查询的并（*UCQ*）（定义在$\boldsymbol{S}$ 上）是有限个定义在S上的 k 元合取查询的析取。我们用 $q(I)$ 表示满足 $I \models q(\mathbf{x})$ 的元组a 的集合。

形式为（式 1）的合取查询的典范实例是带点实例（I, y），其中 I 是活动域为 $\{y_1, \ldots, y_n, \mathbf{x}\}$、其事实为 ϕ 的合取项的实例，且 $\mathbf{y} = y_1 \ldots y_k$。反过来，带点实例（ I, a ）的典范合取查询（其中 $\mathbf{a} = a_1 \ldots a_k$）可通过以下步骤得到：(i) 为每个 $a \in adom(I)$ 关联一个唯一变量 y_a ；(ii) 将 x 设为所有满足 $a \in adom(I) \setminus \{a_1, \ldots, a_n\}$ 的变量 x_a 的一个枚举；(iii) 取查询 $q(y_{a_1}, \ldots, y_{a_n}) :- \exists \mathbf{x} \bigwedge_{R(b_1,\ldots,b_n) \in I} R(y_{b_1}, \ldots, y_{b_n})$。著名的钱德拉-梅林定理指出，元组 a 属于 $q(I)$ 当且仅当 q 的典范实例同态映射到（I, a）。

B 第 3 节的证明

▶**命题 3.4.** 例 *3.3* 中的近单目 *Datalog* 程序不等价于任何单目 *Datalog* 程序。

证明。反设存在一个等价的单目 Datalog 程序 P。令 n 为 P 中任意一条 Ans 规则所含变量数的最大值。考虑由事实 $R(a_0, a_1)$ $R(a_1, a_2) \ldots, R(a_n, a_{n+1})$ 以及事实 $R(b_0, b_1)$ $R(b_1, b_2) \ldots, R(b_n, b_{n+1})$ 构成的 S_{in}-实例 I。那么， $Ans(a_0, a_{n+1})$ 是 $P(I)$ 的一个事实，而 $Ans(a_0, b_{n+1})$ 则不是。简单的同构论证表明，对于所有 $i \leq n+1$ 以及所有 $S \in \mathsf{S}_{aux}$， $S(a_i)$ 属于 $\mathsf{chase}_P(I)$ 当且仅当 $S(b_i)$ 属于 $\mathsf{chase}_P(I)$。由此很容易看出，使用 P 中的一条规则推导出 $Ans(a_0, a_{n+1})$，也意味着可以使用同一条规则推导出 $Ans(a_0, b_{n+1})$。矛盾。◀

▶**定理 3.5**。对于每个近单目 *Datalog* 程序 P 以及 k 元关系符号$R \in \boldsymbol{S}^P_{out}$，都存在一个布尔单目 *Datalog* 程序 P'，其中 $\boldsymbol{S}^{P'}_{in} = \boldsymbol{S}^P_{in} \cup \{Q_1, \ldots, Q_k\}$，使得对于所有 $\boldsymbol{S}^P_{in}$-实例 I和 $a_1, \ldots, a_k \in adom(I)$，以下条件等价：

1. $R(a_1, \ldots, a_k) \in P(I)$,2. $P'(I \cup \{Q_1(a_1)$ $\ldots, Q_k(a_k)\}) = true$。

反过来，对于每个布尔单目 *Datalog* 程序 P'，其包含 $\boldsymbol{S}^{P'}_{in} = \boldsymbol{S} \cup \{Q_1, \ldots, Q_k\}$，且其中每个 Q_i 都是一元的，都存在一个 *TAM Datalog* 程序 P，其包含 $\boldsymbol{S}^P_{in} = \boldsymbol{S}$ 和 $\boldsymbol{S}^P_{out} = \{R\}$，使得上述等价关系成立。

证明。为简化论述，我们可以假设每个关系的关节点位置（如果存在）都是第一个位置。令 $\mathsf{Q} = \{Q_1, \ldots, Q_k\}$。对于每个满足 $arity(S) > 0$ 的关系$S \in \mathsf{S}^P_{out} \cup \mathsf{S}^P_{aux}$， 以及每个偏函数 $f\colon \{1, \ldots, arity(S)\} \to \mathsf{Q}$，我们创建一个一元关系 S^f。 $S^f(x)$ 的直观含义是：

$$\exists y_1 \ldots y_k (S(y_1, \ldots, y_k) \wedge x = y_1 \wedge \bigwedge_{f(i) = Q_j} Q_j(y_i)) \,.$$

令 $\mathbf{S}'_{aux}$ 为所有这些新一元关系构成的集合。最后，我们定义新程序 P' 的规则集合 $\Sigma^{P'}$。取 $\rho \in \Sigma^P$ 中任意一条规则。不失一般性，我们可以假设 ρ 的形式为

$$R_0(\mathbf{x}_0) :- R_1(\mathbf{x}_1), \ldots, R_n(\mathbf{x}_n), E_{n+1}(\mathbf{x}_{n+1}), \ldots, E_{n+m}(\mathbf{x}_{n+m})$$

其中每个 $R_i \in \mathbf{S}^P_{aux} \cup \mathbf{S}^P_{out}$ 以及每个 $E_i \in \mathbf{S}_{in}$。对于 $0 \leq i \leq n$，令 $f_i \colon \{1, \ldots, arity(R_i)\} \to \mathcal{Q}$ 为一个偏函数，并满足以下一致性要求：当某个变量在上述规则的多个 $\mathbf{S}_{aux} \cup \mathbf{S}_{out}$-原子中出现时，设它分别出现在原子 $R_i(\mathbf{x}_i)$ 的第 j 个参数位置和原子 $R_{i'}(\mathbf{x}_{i'})$ 的第 j' 个参数位置，则 $f_i(j) = f_{i'}(j')$（这里允许 $f_i(j)$ 和 $f_{i'}(j')$ 均未定义）。

对于每条规则 $\rho \in \Sigma^P$，以及对于每组满足上述一致性要求的偏函数 $f_0, \ldots, f_n$，我们将规则添加到 $\Sigma^{P'}$：

$$R_0^{f_0}(x_{0,1}) :- R_1^{f_1}(x_{1,1}), \ldots, R_n^{f_n}(x_{n,1}), E_{n+1}(\mathbf{x}_{n+1}), \ldots, E_{n+m}(\mathbf{x}_{n+m}), \bigwedge_{f_0(i)=Q_j} Q_j(x_{0,i})$$

其中 $x_{i,j}$ 表示变量元组 $\mathbf{x}_i$ 中的第 j 个变量。

最后，我们添加规则

$$Ans() :- R^f(x)$$

其中，R 是命题陈述中提到的关系，而 $f \colon \{1, \ldots, arity(R)\} \to \mathcal{Q}$ 是由 $f(i) = Q_i$ 给出的全函数。这就完成了单目 Datalog 程序 P' 的定义。

设 I 为任意一个 $\mathbf{S}^P_{in}$-实例，并令 $I' = I \cup \{Q_1(a_1) \ldots, Q_k(a_k)\}$。

▷ 断言：对于所有 $R^f \in \mathbf{S}'_{aux}$ 和 $c \in adom(I)$，以下两者等价：
1. $c \in (R^f)^{P'(I)}$ **2.** 存在一个元组 $(b_1, \ldots, b_n) \in R^{P(I)}$，使得 $b_1 = c$，并且对于所有 $i \leq n$，如果 $f(i) = Q_j$，那么 $b_i = a_i$。

该断言的两个方向都可以通过对推导长度进行归纳法来证明。特别地，由该断言可得，$Ans() \in P'(I')$ 当且仅当 $R(a_1, \ldots, a_k) \in P(I)$。

现在让我们证明逆向蕴含。对于每个 $S \in \mathbf{S}^P{}_{aux} \cup \{Q_1, \ldots, Q_k\}'$，$\mathbf{S}^P_{aux}$ 包含一个 $(1+k)$ 元符号。我们将使用 Ans^*（其中 Ans 是 P'中的输出谓词）来表示 P的输出谓词 R。 $S^*(\mathbf{x}\ y_1, \ldots, y_k)$ 的直观含义是

$$S(\mathbf{x}) \wedge \bigwedge Q_j(y_j)$$

我们注意到，当 $S \in \mathbf{S}^{P'}_{aux} \cup \{Q_1, \ldots, Q_k\}$ 时，$\mathbf{x}$ 由单个变量组成；当 $S = Ans$时，它为空。

为此，我们向 Σ^P 中加入以下规则。首先，对于每个 Q_i，我们添加如下规则：

$$Q_i^*(x_i, x_1, \ldots, x_k) :- (\text{empty body})$$

我们注意到，尽管该规则是不安全的（即头部中的变量并未出现在体中），但可以通过将规则体扩展为一个 $\mathbf{S}_{in}$-原子来轻松修正：该原子包含头部中的每个变量，而原子其他位置均使用新变量（实现这一点有多种方式，我们将以此方式得到的所有安全规则都添加进来）。

其次，对于 $\Sigma^{P'}$ 中的每条规则 ρ，我们向 Σ^{P} 添加一条由 ρ得到的新规则。不失一般性，我们可以假设 ρ 具有如下形式

$$R_0(\mathbf{x}_0) :- R_1(x_1), \ldots, R_n(x_n), E_{n+1}(\mathbf{x}_{n+1}), \ldots, E_{n+m}(\mathbf{x}_{n+m})$$

其中 $R_0 \in \mathbf{S}^{P'}_{aux} \cup \mathbf{S}^{P'}_{out}$，每个 $R_i \in \mathbf{S}^{P'}_{aux} \cup \{Q_1, \ldots, Q_k\}$，以及每个 $E_i \in \mathbf{S}^{P'}_{in} \setminus \{Q_1, \ldots, Q_k\}$。然后我们向 Σ^{P} 添加以下规则：

$$R_0(\mathbf{x}_0, y_1, \ldots, y_k) :- R_1(x_1, y_1, \ldots, y_k) \ldots, R_n(x_n, y_1, \ldots, y_k)\ E_{n+1}(\mathbf{x}^{n+1}) \ldots, E_{n+m}(\mathbf{x}^{n+m})$$

令 I 为任意一个 $\mathbf{S}^{P}_{in}$-实例，并令 $I' = I \cup \{Q_1(a_1) \ldots, Q_k(a_k)\}$。以下断言可通过对推导长度进行归纳证明。

▷ 断言。对于所有 $S^* \in \mathbf{S}^{P}_{aux}$ 和 $c \in adom(I)$，以下条件等价：
1. $c \in S^{P'(I)}$ **2.** $(c,\ a_1,\ \ldots\ldots,\ a_k) \in (S^*)^{P(I)}$注意，由此可得 $Ans() \in P'(I')$ 当且仅当 $Ans^*(a_1,\ \ldots\ldots,\ a_k) \in P(I)$。 ◀

▶ **推论 3.6**。设 P 是一个近单目 *Datalog* 程序，且 $R \in \boldsymbol{S}^{P}_{out}$。则 (P, R) 定义一个单子二阶逻辑查询。

证明。设 P' 如定理 3.5 中所述。根据定理 3.1，存在一个单子二阶逻辑句子 ϕ，使得对于所有 $\mathbf{S} \cup \{Q_1,\ \ldots\ldots,\ Q_k\}$-实例 I，$Ans() \in P'(I)$ 当且仅当 $I \models \phi$。设

$$\psi(x_1, \ldots, x_k) = \exists Q_1 \ldots Q_k(\phi \wedge \bigwedge_i \forall z(Q_i(z) \leftrightarrow z = x_i))$$

那么，对于所有$\mathbf{S}^{P}_{in}$-实例，I, $I \models \psi(a_1, \ldots, a_k)$当且仅当 $R(a_1, \ldots, a_k) \in P(I)$。 ◀

▶**引理 3.8.** 设 P 是任意树形 *Datalog* 程序。那么，对于每个 $R \in \boldsymbol{S}^{P}_{out}$，展开$(P, R)$ 由无环带标记实例组成。

证明。回忆一下，展开(P, R) 由可推导规则的典范实例组成，其中可推导规则是指通过将规则头的出现替换为相应规则体，从 P 的规则中得到的规则。容易看出，这种替换操作保持树形性，因此每个导出的规则都是树形的。由此可知，每个展开(P, R) 都由无环带标记实例组成。 ◀

▶ **定理 3.9.** 设 $\phi(x_1, \ldots, x_n)$ 是模式$\boldsymbol{S}_{in}$上的一个单子二阶逻辑公式。以下命题等价：

1. ϕ 可由一个 *TAM Datalog* 程序定义，**2.** ϕ 可由一个树形 *Datalog* 程序定义，**3.** ϕ是树决定的。
证明推迟至附录 F。

▶ **推论 3.11** (TAM Datalog 在复合下封闭)。对于所有 *TAM Datalog* 程序P_1 和 P_2，其中$\boldsymbol{S}^{P_2}_{in} = \boldsymbol{S}^{P_1}_{out}$，都存在一个 *TAM Datalog* 程序 $P_3 = (\boldsymbol{S}^{P_1}_{in},\ \boldsymbol{S}^{P_2}_{out},\ \boldsymbol{S}'_{aux},\ \Sigma')$，使得对于所有$\boldsymbol{S}^{P_1}_{in}$-实例 I，$P_3(I) = P_2(P_1(I))$。

证明。P_1 与 P_2 的复合显然可以表示为一个树形 Datalog 程序：我们可以假设 $\mathbf{S}^{P_1}_{aux}$ 和 $\mathbf{S}^{P_2}_{aux}$ 互不相交。令 $P_3 = (\mathbf{S}^{P_1}_{in}, \mathbf{S}^{P_2}_{out}, \mathbf{S}^{P_1}_{aux} \cup \mathbf{S}^{P_1}_{out} \cup \mathbf{S}^{P_2}_{aux}, \Sigma^{P_1} \cup \Sigma^{P_2})$。那么 P_3 定义了 P_1 与 P_2 的复合。注意，P_3 是树形的，但不再必然是几乎单子式的。不过，正如我们将要证明的，P_3 仍然等价于一个 TAM Datalog 程序。

对于每个 k元关系 $R \in \mathbf{S}_{out}^{P_1}$，令 $\phi_R(x_1, \ldots, x_k)$为模式 $\mathbf{S}_{in}^{P_1}$上的 MSO 查询，由(P_1, R)定义。类似地，对于每个 k元关系 $S \in \mathbf{S}_{out}^{P_2}$，令 $\phi_S(x_1, \ldots, x_k)$为模式 $\mathbf{S}_{in}^{P_2}$ 上的 MSO 查询，由(P_2, S)定义。我们可以在 ϕ_S中，将关系符号 $R \in \mathbf{S}_{out}^{P_1}$ 的所有出现替换为其定义公式 ϕ_R。这样，对于每个$S \in \mathbf{S}_{out}^{P_2}$，我们得到一个模式 $\mathbf{S}_{in}^{P_1}$上的 MSO 查询 ϕ'_S。注意，ϕ'_S 恰好是由(P_3, S)定义的 MSO 查询。

根据定理 3.9，对于每个 $S \in \mathbf{S}_{out}^{P_2}$，MSO 查询 ϕ'_S 都可以由一个 TAM Datalog 程序定义。最后，我们将相关的 TAM Datalog 程序合并，得到一个与 P_3等价的 TAM Datalog 程序。◀

▶**定理 3.12**。每个（连通的）*TAM Datalog* 程序都可以在多项式时间内转换为一个等价的（连通的）简单 *TAM Datalog* 程序。

证明。首先，我们将说明如何确保每条规则至多包含一个来自$\mathbf{S}_{in}$的关系出现。考虑任意一条规则，其体中有两个或更多合取项涉及来自$\mathbf{S}_{in}$的关系。由于该程序是树形的，规则体的关联图是无环的。因此，相关规则可以（按需重新排列体中的原子）写成如下形式：

$$R_0(\mathbf{x}_0) :- R_1(\mathbf{x}_1), \ldots, R_i(\mathbf{x}_i), R_{i+1}(\mathbf{x}_{i+1}), \ldots, R_n(\mathbf{x}_n)$$

其中，关系 $R_1, \ldots, R_i$ 中有一个属于 $\mathbf{S}_{in}$，关系 $R_{i+1}, \ldots, R_n$ 中有一个属于 $\mathbf{S}_{in}$，且交集 $\{\mathbf{x}_1, \ldots, \mathbf{x}_i\} \cap \{\mathbf{x}_{i+1}, \ldots, \mathbf{x}_n\}$ 至多包含一个变量 z。事实上，如果程序是连通的，就必定存在这样的变量 z。

令 $\mathbf{u}$ 为 $\{\mathbf{x}_{i+1}, \ldots, \mathbf{x}_n\}$ 中变量的一个无重复枚举，并以 z 开头；否则，则以 $R_{i+1}(\mathbf{x}_{i+1})$ $\ldots, R_n(\mathbf{x}_n)$ 中输入关系原子里出现的任意变量开头。

我们可以用以下两条规则替换上述规则：

$$R_0(\mathbf{x}_0) :- R_1(\mathbf{x}_1), \ldots, R_i(\mathbf{x}_i), R'(\mathbf{u})$$
$$R'(\mathbf{u}) :- R_{i+1}(\mathbf{x}_{i+1}), \ldots, R_n(\mathbf{x}_n)$$

其中 R 是一个元数适当的新辅助关系，其关节点位置是第一个位置。注意，两条新规则中来自 $\mathbf{S}_{in}$ 的关系出现次数都严格少于原规则，并且这一构造保持连通性。重复这一过程，最终得到至多线性数量的规则，且每条规则的大小都不超过原规则的大小。此外，这显然可以在多项式时间内完成。

接下来，我们说明如何确保每个规则体至少包含一个（因此恰好包含一个）来自$\mathbf{S}_{in}$的关系。这里我们利用这样一个事实：派生到定义关系中的每个元组，都必须由源自输入实例中的事实的值组成。具体而言，给定一条规则

$$R_0(\mathbf{x}_0) :- R_1(\mathbf{x}_1), \ldots, R_n(\mathbf{x}_n)$$

其中不包含任何来自$\mathbf{S}_{in}$的关系。令 x 为规则体中某个原子的关节点位置上出现的任意变量，并用所有可能的规则替换该规则：这些规则的规则体都扩展了一个额外的原子 $R'(\mathbf{u}\ x, \mathbf{v})$，其中 $R' \in \mathbf{S}_{in}$，且 $\mathbf{u}, \mathbf{v}$ 是不同的新变量。同样，连通性得以保留。◀

C 第4节的证明

▶**定理 4.4**。如果两个 $\exists$*Datalog* 程序都具有广义右伴随，那么它们的复合也具有广义右伴随。

XX：22 Datalog 程序的右伴随，以及同态对偶性

证明。 只需定义 $\Omega_{P_2 \cdot P_1}(J) = \{(J'', \kappa \cdot \iota) \mid (J', \iota) \in \Omega_{P_2}(J), (J'', \kappa) \in \Omega_{P_1}(J')\}$，参见下面的交换图：

$$
\begin{array}{ccc}
P_2(P_1(I)) & \longrightarrow & J \\
{\scriptstyle id}\uparrow & & \uparrow{\scriptstyle \iota} \\
P_1(I) & \longrightarrow & J' \qquad (J', \iota) \in \Omega_{P_2}(J) \\
{\scriptstyle id}\uparrow & & \uparrow{\scriptstyle \kappa} \\
I & \longrightarrow & J'' \qquad (J'', \kappa) \in \Omega_{P_1}(J')
\end{array}
$$

◀

▶ **定理 4.6。** 每个 *TAM Datalog* 程序 P 都有一个广义右伴随 Ω_P。如果 P 是连通的，则 $\Omega_P(J)$ 始终是一个单例。此外，给定 J 和 P，$\Omega_P(J)$ 可在双指数时间内计算；当 P 的元数有界时，则可在指数时间内计算。

证明。 我们首先考虑连通程序这一特殊情形。不失一般性，我们可以假设 P 是简单的。这意味着每条规则都具有以下形式：

$$R_0(\mathbf{x}_0) :- E(\mathbf{y}), R_1(\mathbf{x}_1), \ldots, R_m(\mathbf{x}_m) \qquad \text{(Eq. 2)}$$

$$R(\mathbf{x}) :- E(\mathbf{y}), R_1(\mathbf{x}_1), \ldots, R_m(\mathbf{x}_m) \qquad \text{(Eq. 3)}$$

其中，E 是输入关系，每个 R_i 是辅助关系，而 R 是输出关系。

为简化下文的表述，我们引入一些进一步的记号。对于上述规则类型中的每个原子$R_i(\mathbf{x}_i)$，我们将 $p_i \in \{1, \ldots, n\}$（其中 $n = arity(E)$）记为满足以下条件的唯一编号：y_{p_i} 等于 $R_i(\mathbf{x}_i)$ 中的关节变量。根据 TAM Datalog 的定义，以及对 P 的连通性和简单性的假设，确实可以得到这样的索引存在且唯一。

我们构造一个 $\mathbf{S}_{in}$-实例 J'。该实例 J' 由所有事实 $E((b_1, X_1).....(b_n, X_n))$ 组成，其中

1. 每个 b_i 都是 $adom(J) \cup \{\bot\}$ 的元素，而 X_i 是关于 $adom(J) \cup \{\bot\}$ 的一组 $\mathbf{S}_{aux}$-事实（不一定是 J 的事实），其中 b_i 出现在关节点位置；2. 对于上述形式 (1) 的每条规则，以及每个映射 g：$\{\mathbf{y}, \mathbf{x}1, \ldots, \mathbf{x}_m\} \to adom(J) \cup \{\bot\}$，如果对于每个 $1 \leq i \leq m$，$R_i(g(\mathbf{x}_i)) \in X_{p_i}$，那么 $R_0(g(\mathbf{x}_0)) \in X_{p_0}$。3. 对于上述形式 (2) 的每条规则，以及每个映射 g：$\{\mathbf{y}, \mathbf{x}1, \ldots, \mathbf{x}_m\} \to adom(J) \cup \{\bot\}$，如果对于每个 $1 \leq i \leq m$，$R_i(g(\mathbf{x}_i)) \in X_{p_i}$，那么 $R(g(\mathbf{x}))$ 是 J 的一个事实。

请注意，J' 中可能的事实 $E((b_1, X_1).....(b_n, X_n))$ 的总数相对于 P 和 J 的组合规模呈双指数增长；若 P 的元数有界，则相对于 J 仅呈指数增长。

最后，$\Omega_P(J) = \{(J', \iota)\}$，其中 ι 是从 J' 到 J 的自然投影，将所有形如 (a, X) 的元素映射到 a（而对形如 $(\bot, X)$ 的元素未定义）。

▷ 断言。对于所有 $\mathbf{S}_{in}$-实例 I，$P(I)$ 当且仅当 $I \to J'$。此外，可以构造见证同态，使定理陈述中的图表可交换。

证明。 $[\Rightarrow]$ 令 h： $P(I) \to J$。回顾一下，我们用 $\mathsf{chase}_P(I)$ 表示由 I 追赶得到的 $\mathbf{S}_{in} \cup \mathbf{S}_{out} \cup \mathbf{S}_{aux}$-实例（其中 I 和 $P(I)$分别是其 $\mathbf{S}_{in}$-约化结构和 $\mathbf{S}_{out}$-约化结构）。我们将 h 扩展到 $\mathsf{chase}_P(I)$ 的整个活动域：对于不属于 $adom(P(I))$的每个元素 a，均将其映射到 $\bot$。在稍微滥用记号的情况下，以下将扩展后的映射记为 h，其从 $adom(\mathsf{chase}_P(I))$映射到 $adom(J) \cup \{\bot\}$。对于每个

$a \in adom(\text{chase}_P(I))$，令 F_a 为 $\text{chase}_P(I)$ 中所有在关节点位置出现 a 的 $\mathbf{S}_{aux}$-事实所组成的集合。我们定义 $h'(a) = (h(a)\ h(F_a))$。我们断言，h' 是从 I 到 J'的同态。为证明这一点，令 $E(a_1, \ldots, a_n)$为 I的任意事实。我们必须证明，事实$E((h(a_1)\ h(F_{a_1})) \ldots, (h(a_n)\ h(F_{a_n})))$属于 J'。也就是说，我们必须证明上述条件 1–3 均得到满足。

显然，第一个要求得到满足，即 $h(X_{a_i})$ 由事实构成，其中 $h(a_i)$ 出现在关节点位置。

为证明第二个要求成立，考虑一个形式为 (1) 的规则，以及任意映射 $g\colon \{\mathbf{y}, \mathbf{x}1, \ldots, \mathbf{x}_m\} \to adom(J)$，使得对于每个 $1 \leq i \leq m$，都有 $R_i(g(\mathbf{x}_i)) \in X_{p_i}$。根据构造，这意味着每个事实 $R_i(g(\mathbf{x}_i))$ 都是追赶 $_P(I)$ 中某个事实 $R_i(\mathbf{b}_i)$ 的 h-像。现在考虑映射 $\tilde{g}\colon \{\mathbf{y}, \mathbf{x}1, \ldots, \mathbf{x}_m\} \to adom(I)$，其定义为 $\tilde{g}(\mathbf{y}) = (a_1 \ldots, a_n)$，且 $\tilde{g}(\mathbf{x}_i) = \mathbf{b}_i$ 对于 $i \leq i \leq m$ 成立。注意，$\tilde{g}$ 是一个定义良好的函数。事实上，对于每个 $i \leq i \leq m$ 和每个 $z \in R(\mathbf{x}_i)$，如果 z 在规则中出现多次，那么 z 必然是出现在 $R(\mathbf{x}_i)$ 的关节点位置 p_i 上的变量。因此，z 出现在 $\mathbf{y}$ 的位置 p_i，故必然有 $R_i(g(\mathbf{x}_i))$ 属于 $X_{a_{p_i}}$ 的 h-像，因此 $\tilde{g}(z) = a_{p_i}$。

由于 $\text{chase}_P(I)$ 对 P 的规则封闭，我们可以得出，$R_0(\tilde{g}(\mathbf{x}_0))$ 属于 $\text{chase}_P(I)$。

令 z 为在 $R_0(\mathbf{x}_0)$ 中出现在关节点位置的变量。回想一下，z 在 $\mathbf{y}$ 的位置 p_0 出现，并且 $\tilde{g}(z) = a_{p_0}$。那么 $R_0(\tilde{g}(\mathbf{x}_0)) \in F_{a_{p_0}}$，因此，$h(F_{a_{p_0}})$ 包含 $R_0(h \circ \tilde{g}(\mathbf{x}_0))$。注意，根据定义，$h \circ \tilde{g} = g$。特别地，$R_0(h \circ \tilde{g}(\mathbf{x}_0)) = R_0(g(\mathbf{x}_0))$。因此我们有 $R_0(g(\mathbf{x}_0)) \in h(F_{a_{p_0}})$，证毕。

为证明第三个要求成立，考虑一个形如 (2) 的规则以及任意映射 $g\colon \{\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_m\} \to adom(J)$，使得对于每个 $1 \leq i \leq m$，$R_i(g(\mathbf{x}_i)) \in X_{p_i}$。按照与之前完全相同的推理，规则头 $R(g(\mathbf{x}))$ 的一个 h-原像属于 $\text{chase}_P(I)$。因此，它属于 $P(I)$，所以 $R(g(\mathbf{x}))$ 是 J 的一个事实。

从构造中还可以清楚地看出，$h \circ id = \iota \circ h'$，其中 id 是 $adom(I) \cap adom(P(I))$ 上的恒等函数。也就是说，该图表交换。

[⇐] 反之，令 $h\colon I \to J'$。注意，$adom(\text{追赶}_P(I)) = adom(I)$，因此对于所有 $a \in adom(\text{追赶}_P(I))$，我们有$h(a)$定义良好。令 $h'\colon adom(I) \to adom(J)$ 为满足以下条件的映射：当 $h(a) = (b, X)$ 时，$h'(a) = b$。

▷ 子断言 1。对于所有 $a \in adom(\text{追赶}_P(I))$，如果 $h(a) = (b, X)$，则追赶$_P(I)$ 中每个 h'-事实的 X h'-像，其中 a 出现在关节点位置，均属于 X。

▷ 子断言 2。 h' 是从 $P(I)$ 到 J 的同态。

对子断言 1，可以通过对所讨论事实的推导长度进行归纳来证明。

为了证明子断言 2，设 $R(\mathbf{a})$ 是属于 $P(I)$ 的一个 $\mathbf{S}_{out}$-事实。其推导必然使用上述形式 (2) 的规则，并采用赋值 g（其中 $g(\mathbf{x}) = \mathbf{a}$）。根据子断言 1，我们有 $R_i(h'(g(\mathbf{x}_i)))$ 属于 $h(g(y_{p_i}))_2$，其中 y_{p_i} 是 $\mathbf{x}_i$ 中的关节变量。此外，$E(g(\mathbf{y}))$ 在 I 中成立，因此 $E(h(g(\mathbf{y})))$ 在 J' 中成立。根据 J' 的构造，这意味着 $R(h'(g(\mathbf{x})))$，也就是 $R(h'(\mathbf{a}))$，属于 J。这就完成了对连通 TAM Datalog 程序这一情形的证明。

从构造也显然可知，$\iota \circ h = h' \circ id$，其中 id 是 $adom(I) \cap adom(P(I))$ 上的恒等函数。也就是说，该图表交换。◁

最后，我们展示如何处理非连通 TAM Datalog 程序。设 P 是一个非连通 TAM Datalog 程序。设 P' 由 P 添加一个新的二元输入关系 S 得到，并利用该关系以某种方式使每条规则连通

XX：24 Datalog 程序的右伴随与同态对偶

（更准确地说，每当规则体的关联图包含多个连通分量时，我们就在体中添加 S-原子以连接这些分量，同时保持树形性和近乎单子性。对于每个输入实例 I，我们将由 I 表示的 $\mathsf{S}_{in} \cup \{S\}$-实例 $\widehat{}$ 定义为：在 I 中加入所有形如 $S(a,b)$ 的事实，其中 $a,b \in adom(I)$。此外，给定模式 $\mathsf{S}_{in} \cup \{S\}$ 上的一个实例 J'，我们将 J' 的“S-分量”定义为 J' 的一个完全 S-连通子实例的 S_{in}-收缩。显然，如果 J 是一个 $\widehat{\mathsf{S}_{in}}$-实例，且 J' 是一个 $\mathsf{S}_{in} \cup \{S\}$-实例，那么 $J \to J'$ 当且仅当对某个 S-分量 J''（其属于 J'）有 $J \to J''$。现在，我们只需将 $\Omega_P(J)$ 定义为实例 $\Omega_{P'}(J)$ 的所有 S-分量的集合。$\widehat{}$ $\widehat{}$于是我们有：$P(I) \to J$ 当且仅当 $P'(I) \to J$ 当且仅当 $I \to \Omega_{P'}(J)$，当且仅当对于某个 $J' \in \Omega_P(J)$ 有 $I \to J'$。

顺带一提，我们指出，呈现最后一个论证还有另一种方式：在这种方式中，$\widehat{}$我们将连通情形推广到一般情形。我们可以把将 I 映射到 I 的函数视为一个本身具有广义右伴随的函子（将 I 映射到它的 S-连通分量）。因此，我们可以运用定理 4.4，通过复合来进行论证。◀

▶**命题 4.7.** 例 *3.7* 中的树形 *Datalog* 程序 P（它不是近似单目的）不具有广义右伴随。

证明。 反设 P 具有广义右伴随 Ω_P。令 J 为由事实 $R(0,1)$ 和 $R(1,0)$构成的二元素 R}-实例。

对于 $n \geq 1$，令 C_n 为由事实 $E(v_0,v_1), \ldots, E(v_{n-1},v_n)$,$E(v_n,v_0)$ 构成的 $\{E,F\}$-实例，也就是长度为 n 的有向 E-环。显然，对于所有 $n \geq 1$，$P(C_n) \to J$。因此，对每个 $n \geq 1$，存在某个 $J' \in \Omega_P(J)$，使得 $C_n \to J'$。对于每个 $J' \in \Omega_P(J)$ 以及 J' 中的每个元素 b，我们将 $n_{J',b}$ 定义为任意选取的值，使得 $(C_n,v_0) \to (J',b)$；若不存在这样的值，则将其设为未定义。根据我们先前的观察可知，至少存在一对 (J',b)，其中 $J' \in \Omega_P(J)$，使得 $n_{J',b}$ 有定义。令 m 为所有已定义的$n_{J',b}$的公倍数。

对于每一对正整数 $e \leq f$，令 $I_{e,\,f}$ 为如下所示的 $\{E,\ F\}$-实例：

$$u_0 \xrightarrow{E} v_0 \xrightarrow{F} u_1 \xrightarrow{E} v_1 \xrightarrow{F} u_2 \xrightarrow{\text{sequence of } e \ E\text{-edges}} z \xrightarrow{\text{sequence of } f \ F\text{-edges}} u_0$$

▷ 断言1 对于所有 $1 \leq e \leq f$，以下两者等价：**1.** $I_{e,\,f} \to J'$ 对于某个 $J' \in \Omega_P(J)$**2.** $e \neq f$。

证明。 若 $e = f$，则 $P(I_{e,\,f})$ 包含一个奇数长度的 R-环，即 $u_0 \xrightarrow{R} u_1 \xrightarrow{R} u_2 \xrightarrow{R} u_0$，因此 $P(I_{e,\,f}) \not\to J$。因此，对于所有 $J' \in \Omega_P(J)$，$I_{e,\,f} \not\to J'$。另一方面，若 $e < f$，则 $P(I_{e,\,f})$ 是若干 R-路径的不交并，显然 $P(I_{e,\,f}) \to J$。因此，对于某个 $J' \in \Omega_P(J)$，$I_{e,\,f} \to J'$。◁

现在，令 e 大于 $\Omega_P(J)$ 中所有实例的论域，并令 $f = e + m$。根据断言 1，存在一个同态 $h : I_{e,f} \to J'$，其中 $J' \in \Omega_P(J)$。我们将证明 h可以扩展为一个同态 $h' : I_{f,f} \to J'$，这与断言 1 矛盾。

Let

$$u_2 = x_0 \xrightarrow{E} x_1 \cdots \xrightarrow{E} x_e = z$$

是由连接 u_2 和 z 的 E-边组成的 $I_{e,\,f}$ 的子实例。同样，令

$$u_2 = x_0 \xrightarrow{E} x_1 \cdots \xrightarrow{E} x_e \xrightarrow{E} x_{e+1} \ldots \xrightarrow{E} x_f = z$$

是由连接 u_2 和 z 的 E-边组成的 $I_{f,\ f}$ 的子实例。回想一下， $f = e + m$。由于 e 大于 J' 的域大小，因此必有 $h(x_i) = h(x_j) = b$，其中某个 $i < j \leq e$ 对应于 J' 的某个元素 b。这意味着 b 位于 J' 中的一个有向 E-环上，因此特别地，它位于一个长度为 m 的有向 E-环上，例如 $b = b_0 \xrightarrow{E} b_1 \cdots \xrightarrow{E} b_m = b$。映射 $h' : I_{f,\ f} \to J'$ 可以通过扩展 h 并将 x_{e+i} 映射到 b_i（对于 $1 \leq i \leq m$）来直接构造。◀

▶ **命题 4.8**。 **</style}由单条规则 (x)**： $- E(x, x)$给出的单目 *Datalog* 程序不具有广义右伴随。

证明。令 P 为所讨论的布尔 Datalog 程序。注意，Unfoldings(P, Ans) 由一个带标记结构组成，该结构是 c-无环的，但不是无环的。它不具有广义右伴随：令 J 为一个空的 $\mathbf{S}^P_{out}$- 实例，并反设存在有限集 $\{J_1, \ldots, J_n\}$，使得对于所有 $\mathbf{S}^P_{in}$- 实例 I， $P(I) \to J$ 当且仅当对于某个 $i \leq n$， $I \to J_i$。令 I_c 为由一个形如 $Ans(a, a)$ 的自反 Ans-边构成的实例。显然， $P(I_c) \not\to J$，因此 $I_c \not\to J_i$。也就是说， $J_1, \ldots, J_n$ 都不包含自反 Ans-边。接下来，令 I_n 为一个大小为 n 的（非自反） Ans-团，其中 n 是一个大于每个 J_i 大小的任意数。于是， $I_n \not\to J_i$（因为如果存在这样的同态， J_i 必然会包含一条自反 Ans-边），但显然 $P(I_n) \to J$。◀

下一个引理列出了强线性 ∃Datalog 程序的主要区别性性质，我们将用到这些性质。

▶ **引理 C.1.** 设 $P = (\boldsymbol{S}_{out}, \boldsymbol{S}_{aux}, \Sigma)$ 是一个强线性 ∃*Datalog* 程序，且 I, I'是 $\boldsymbol{S}_{in}$-实例。

(a) $P(I \cup I') \leftrightarrow_{adom(I \cup I')} P(I) \cup P(I')$. *(b)* 对于 I的每个解 J，以及满足 $dom(f) = adom(I)$ 的每个函数 f， $f[J]$ 都是 $f[I]$的一个解，其中 $f[K]$ 表示通过将 K 中的每个值 $a \in dom(f)$ 替换为 $f(a)$ 得到的实例。

证明. (a) 回顾一下， $P(I)$可以定义为相对于 P， I的一个（任意选取的）泛解的 $\mathbf{S}_{out}$-约化结构。设 J, J', J'' 分别是 I, I'和$I \cup I'$的泛解。不失一般性，我们可以假设 $adom(J) \cap adom(J') \subseteq adom(I \cup I')$。容易看出， $J \cup J'$ 是 $I \cup I'$的一个解。（事实上，每当某个强线性规则的体在 $J \cup J'$中成立时，它就在 J 或 J'中成立，因此规则头也在同一个实例中成立，从而也在 $J \cup J'$中成立。）因此，根据泛解的定义， $J'' \to_{adom(I \cup I')} J \cup J'$。反过来，由于 J'' 既是 I 的解，也是 I'的解，根据泛解的定义，我们有$J \to_{adom(I \cup I')} J''$ 且 $J' \to_{adom(I \cup I')} J''$。由此可得 $J \cup J' \to_{adom(I \cup I')} J''$。总之， $J \cup J' \leftrightarrow_{adom(I \cup I')} J''$。因此，它们的 $\mathbf{S}_{out}$-约化结构之间也存在相同的关系。

（**b**）由于 $I \subseteq J$，显然也有 $h[I] \subseteq h[J]$。此外， $h[J]$ 对程序的规则封闭：如果某个强线性约束的体由$\mathbf{a}$ 中的某个元组 $h[J]$满足，那么存在某个元组$\mathbf{b} \in h^{-1}[\mathbf{a}]$ 中的元组满足 J中的相同规则体。因此，该规则的头部在 J 对$\mathbf{b}$成立，进而在 I 对$\mathbf{a}$成立。◀

▶**定理 4.11**。 每个强线性 ∃*Datalog* 程序 P 都有一个广义右伴随Ω_P。此外，对于每个 $\boldsymbol{S}_{out}$-实例 J， $\Omega_P(J)$ 是一个单元素集合，并且可以根据 J和 P在指数时间内计算得到。如果 P固定，则 $\Omega_P(J)$ 可以在多项式时间内计算得到。

XX：26 Datalog 程序的右伴随，以及同态对偶

证明。令 $P = (\mathbf{S}_{in}, \mathbf{S}_{out}, \mathbf{S}_{aux}, \Sigma)$，并令 J为任意一个 $\mathbf{S}_{out}$-实例。令 $D = adom(J) \cup \{\bot\}$，其中 $\bot$ 是一个新值。我们将 J' 定义为由域 D 上的所有事实 $R(\mathbf{d})$ 构成的 $\mathbf{S}_{in}$-实例，其中满足 $P(\{R(\mathbf{d})\}) \rightarrow_{adom(J)} J$。

由 [7, 推论 5.10] 可知， J' 可在指数时间内计算；若 P 固定，则可在多项式时间内计算。为完整起见，我们在此简要给出一个直接论证：容易看出， J'恰好是定义域为 D 、满足以下两个条件的最大$\mathbf{S}$-实例的$\mathbf{S}_{in}$-约化结构：

1. The $\mathbf{S}_{out}$-约化结构 K 包含于 J。**2.** 的所有规则在 K中均成立。

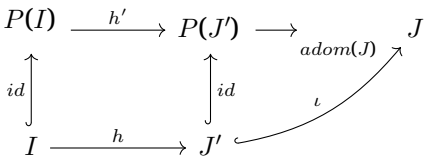
注意到， K 可通过贪心方式在指数时间内计算：首先将其设为满足上述 (1) 的最大实例，然后迭代地移除任何违反某条规则的$\mathbf{S}_{in} \cup \mathbf{S}_{aux}$-事实 $S(\mathbf{d})$

$$\exists \mathbf{z}\big(R_1(\mathbf{x}_1), \ldots, R_n(\mathbf{x}_n)\big) :- S(\mathbf{y})$$

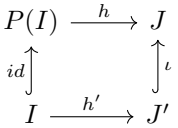
P的规则。此外，注意到如果$\mathbf{S}_{in}$-关系的元数有界，那么 J' 的规模为多项式级；并且，如果 P 固定，则 J' 可在多项式时间内计算。

根据 J' 的定义可知， $P(J') \rightarrow_{adom(J)} J$ （参见引理 C.1(a)）。令$\iota : adom(J') \rightarrow adom(J)$为 $adom(J)$上的恒等函数。我们断言，(J', ι)是一个右伴随。

在一个方向上，令 I 为任意一个$\mathbf{S}_{in}$-实例。如果存在同态 $h\colon I \rightarrow J'$，则 n 我们得到如下交换图，其中 h' 由引理 2.8 给出：



反过来，设 $h\colon P(I) \rightarrow J$。令 h' 通过将 $adom(I) \setminus adom(P(I))$ 中的所有元素映射到 $\bot$，来扩展 h。我们断言 $h'\colon I \rightarrow J'$。考虑 I 中任意一个事实 $S(\mathbf{b})$。我们必须证明 $S(h'(\mathbf{b})) \in J'$，换言之，证明 $P(\{S(h'(\mathbf{b}))\}) \rightarrow_{adom(J)} J$。由于 $\{S(h'(\mathbf{b}))\} \subseteq h'[I]$，后者的每个解都是前者的解。根据引理 C.1(b)， $h'[P(I)]$ 是 $h'[I]$的一个解。由于 h 和 h' 在 $adom(P(I))$ 上一致，因此 $h'[P(I)] = h[P(I)]$。综合以上各点可知， $h[P(I)]$ 是 $\{S(h'(\mathbf{b}))\}$ 的一个解。根据泛解的定义，存在一个同态 $g\colon P(\{S(h'(\mathbf{b}))\}) \rightarrow_{\{h'(\mathbf{b})\}} h[P(I)] \subseteq J$。由此证明了 $h'\colon I \rightarrow J'$。此外，从构造可清楚看出，以下图表可交换：



◀

D 第 6 节证明

▶**定理 6.7**。 设 Σ 是一个具有广义右伴随的有限弱无环 *TGD* 集。设 F 是任意有限带标记实例集。若 F 的每个成员都具有$(P_\Sigma(A), \boldsymbol{a})$ 的形式，其中$(A, \boldsymbol{a})$是某个 c-无环带标记实例，则 F 相对于 Σ具有有限对偶性。

证明。设 D 是由定理 6.3 给出的有限集，使得 $(\{(A', \mathbf{a}) \mid (A, \mathbf{a}) \in F\}\ D)$ 构成一个同态对偶。令 $D' = \{(P(B'), \mathbf{b}') \mid (B, \mathbf{b}) \in D, (B', \iota) \in \Omega_{P_\Sigma}(B)\ \iota(\mathbf{b}') = \mathbf{b}\}$。注意，$D'$ 由满足 Σ 的带标记实例组成。我们将证明 (F, D') 相对于 Σ构成一个同态对偶。

设 $(C,\mathbf{c})$ 是一个带点实例，其中 $C \models \Sigma$。以下等价链成立：

$$(C, \mathbf{c}) \in F\uparrow \tag{1}$$

$$\Updownarrow \text{Lemma 2.8 and } P_\Sigma(C) \leftrightarrow_{adom(C)} C$$

$$(C, \mathbf{c}) \in \{(A', \mathbf{a}) \mid (A, \mathbf{a}) \in F\}\uparrow \tag{2}$$

$$\Updownarrow \text{对偶性假设}$$

$$(C,\ \mathbf{c}) \notin D\downarrow \tag{3}$$

$$\Updownarrow \text{将在下文证明}$$

$$(C,\ \mathbf{c}) \notin D'\downarrow \tag{4}$$

还需证明 (3) 与 (4) 等价。

从 (3) 推出 (4)：用逆否命题证明：假设对于某个 $(B, \mathbf{b}) \in D$、$(B', \iota) \in \Omega_{P_\Sigma}(B)$，且 $\iota(\mathbf{b}') = \mathbf{b}$，有 $(C, \mathbf{c}) \to (P_\Sigma(B'), \mathbf{b}')$。显然，我们有 $id : (B', \mathbf{b}') \to (B', \mathbf{b}')$。由广义伴随性质可得 $(P_\Sigma(B'), \mathbf{b}') \to (B, \iota(\mathbf{b}'))$。因此，由传递性，且由于 $\iota(\mathbf{b}') = \mathbf{b}$，我们有 $(C, \mathbf{c}) \to (B, \mathbf{b})$，从而 $(C, \mathbf{c}) \in D\downarrow$。

从 (4) 推出 (3)：同样用逆否命题证明：假设 $(C, \mathbf{c}) \in D\uparrow$。由于 $P_\Sigma(C) \leftrightarrow_{adom(C)} C$，可得对于某个 $(B, \mathbf{b}) \in D$，有 $(P_\Sigma(C), \mathbf{c}) \to (B, \mathbf{b})$。由伴随性质可得，对于某个 $(B, \mathbf{b}) \in D$、$(B', \iota) \in \Omega_{P_\Sigma}(B)$，且 $\mathbf{b}' \in \iota^{-1}(\mathbf{b})$，有 $(C, \mathbf{c}) \to (B', \mathbf{b}')$。于是还有 $(C, \mathbf{c}) \to (P_\Sigma(B'), \mathbf{b}')$。这意味着 $(C, \mathbf{c}) \in D'\downarrow$。◀

我们将带点实例 $(A,\ \mathbf{a})$ 的*c*-围长 定义为其 A 关联图中不经过$\mathbf{a}$ 中任何元素的最短环的长度（如果不存在这样的环，则为 ∞ ）。注意，当且仅当带点实例的 c-围长为 ∞ 时，它是 c-无环的。

▶ 引理 *D.1*（带指定元素的稀疏不可比引理）。对于任意带点实例 $(I, \boldsymbol{a})$ 和 $m > 0$，存在一个带点实例 $(I', \boldsymbol{a})$，其 *c*-围长至少为 m，使得 $(I', \boldsymbol{a}) \to (I, \boldsymbol{a})$，并且对于所有大小至多为 m 的带点实例 $(J, \boldsymbol{b})$，$(I, \boldsymbol{a}) \to (J, \boldsymbol{b})$ 当且仅当 $(I', \boldsymbol{a}) \to (J, \boldsymbol{b})$。

证明。设给定带点实例 $(I, \mathbf{a})$，其中 $\mathbf{a} = a_1 \dots a_k$，且 I 是模式 S 上的一个实例。令 $\widehat{\ }I$ 为模式 $\mathbf{S} = \mathbf{S} \cup \{Q_1, \dots, Q_k\}$ 上的实例，它通过添加一元事实 $\widehat{\ }Q_i(a_i)$ 扩展了 I。根据稀疏不可比引理的标准版本，存在一个 S-实例 $\widehat{\ }I''$，其围长至少为 m，使得 $I'' \to I$，并且对于所有大小至多为$\widehat{\ }m$的 S-实例 J，$I'' \to J$ 当且仅当 $I \to J$。现在，令 I' 为由 I'' 得到的 S-实例，具体操作为：(i) 将满足一元谓词 Q_i 的每个元素替换为 a_i；(ii) 删除一元谓词 Q_i。此操作可能引入新的环，但不难看出，任何因此新引入的短环都必须经过某个指定元素。因此，$(I', \mathbf{a})$ 的 c-围长至少为 m。此外，对于所有大小至多为 m 的 S-实例 $(J, \mathbf{b})$，我们有$\widehat{\ }I \to \widehat{\ }J$，即 $(I, \mathbf{a}) \to (J, \mathbf{b})$ 当且仅当 $I \to J$，当且仅当 $I'' \to J$，当且仅当 $(I', \mathbf{a}) \to (J, \mathbf{b})$。◀

▶**定理 6.8.** 设 Σ 为任意一组树形单目元组生成依赖。设 F 为任意一个由两两不同态不可比的带标记实例 $(A,\boldsymbol{a})$ 构成的有限集，且满足 $A \models \Sigma$。那么，以下各项等价：

1. F 相对于 Σ具有有限对偶，**2.** 每个 $(A, \boldsymbol{a}) \in F$都与 $(P_\Sigma(A'),\boldsymbol{a})$ 同态等价，其中某个 $(A', \boldsymbol{a})$是 *c*-无环的。

证明。从 2 到 1 的方向立即由前述定理得出。从 1 到 2：假设 F相对于 Σ具有有限对偶，由 $(D_1, \mathrm{d}_1) \ldots, (D_n, \mathrm{d}_n)$ 构成，其中 $D_i \models \Sigma$，并令 $(A, \mathrm{a}) \in F$。则 $(A, \mathrm{a}) \not\to (D_i, \mathrm{d}_i)$。令 $m = s \cdot t$，其中 s 是 A中事实的数量，且t 是 Σ中某个元组生成依赖的体内合取项数目的最大值。根据引理 D.1，存在一个 c-围长至少为 m的带点实例 (A', a)，使得 $(A', \mathrm{a}) \to (A, \mathrm{s})$，且对所有 $i \leq n$都有 $(A', \mathrm{a}) \not\to (D_i, \mathrm{d}_i)$。由于 $(A', \mathrm{a}) \to (A, \mathrm{a})$ 且 $A \models \Sigma$，我们有 $(P(A'), \mathrm{a}) \to (P(A), \mathrm{a}) \to (A, \mathrm{a})$（根据引理 2.8 和引理 5.1(2)）。

我们不妨假设 A' 包含属于$P(A')$的所有一元事实。这是因为：(i) 添加这些一元事实不会改变该实例的 c-围长；(ii) 用 $P(A')$中的事实扩展 A'时，条件 $(A', \mathbf{a}) \to (A, \mathbf{s})$ 得以保持，因为 $P(A', \mathbf{a}) \to (A, \mathbf{s})$；(iii) 用额外事实扩展 A' 时，条件 $(A', \mathbf{a}) \not\to (D_i, \mathbf{d}_i)$ 显然也得以保持。

我们已经注意到 $(P(A'), \mathrm{a}) \to (A, \mathrm{a})$。此外，$(P_\Sigma(A'), \mathrm{a}) \not\to (D_i, \mathrm{d}_i)$（因为，否则，既然 $(A', \mathrm{a}) \subseteq (P_\Sigma(A'), \mathrm{a})$，就会有 $(A', \mathrm{a}) \to (D_i, \mathrm{d}_i)$）。由于$P_\Sigma(A') \models \Sigma$ （根据引理 5.1(1)），且 $(P_\Sigma(A'), \mathrm{a}) \not\to (D_i, \mathrm{d}_i)$，根据对偶性假设，$F$ 中存在某个带点实例可同态映射到 $(P_\Sigma(A'), \mathrm{a})$。事实上，相关的带点实例必定是 (A, a)（否则就会与 F 中的元素两两同态不可比这一事实矛盾）。令 $h : (A, \mathrm{a}) \to (P_\Sigma(A'), \mathrm{a})$。

令 (B, a) 为 $(P_\Sigma(A'), \mathrm{a})$ 的子实例，它是 (A, a) 在 h 下的像。由于 $P_\Sigma(A')$ 中的所有一元事实已经属于 A'，且 Σ 是单目的，因此 $P_\Sigma(A')$ 中的每个事实要么属于 A'，要么可以通过一次规则应用由 A' 中的事实推导出来。由此可知，存在 A 的一个子实例 B'，其大小至多为 $|B| \cdot t$，使得 $B \subseteq P_\Sigma(B')$。由于 $|B| \leq s$，因此 B' 是 c-无环的。此外，$(A, \mathrm{a}) \to (P_\Sigma(B'), \mathrm{a})$，且 $(P_\Sigma(B'), \mathrm{a}) \subseteq (P_\Sigma(A'), \mathrm{a}) \to (A, \mathrm{a})$，因此 $(P_\Sigma(B'), \mathrm{a})$ 也 $\to (A, \mathrm{a})$。所以，(A, a) 与 $(P_\Sigma(B'), \mathrm{a})$ 同态等价。◀

▶**定理 6.10.** 设 Σ 是任意一个具有广义右伴随的有限元组生成依赖集。每个有限的 F *c*-无环带基点 *ABox* 集，在范畴 $ABox_\Sigma[\boldsymbol{S}]$ 中都具有有限对偶。

证明。设 D 是一个有限带点实例集，使得 (F, D) 构成同态对偶，如定理 6.3 所述。令 $D' = \{(B, \mathrm{b}') \mid (B, \mathrm{b}) \in D, (B', \iota) \in \Omega^{P_\Sigma}(B)\ \iota(\mathrm{b}') = \mathrm{b}\}$。注意，$D'$ 由不一定满足 Σ 的实例组成。通过与定理 6.7 的证明中相同的论证，我们可以证明 (F, D') 在范畴 $ABox_\Sigma[\mathrm{S}]$ 中构成同态对偶。◀

E 第 7 节的证明

▶**引理 E.1.** 设 $\boldsymbol{S}$ 是任意模式，Σ 是关于 $\boldsymbol{S}$ 的任意一阶理论。设 q 是关于$\boldsymbol{S}$ 的任意合取查询的并，且设 E^+, E^- 是有限带点实例集$(I, \boldsymbol{a})$，满足 $I \models \Sigma$。那么下列条件等价：

1. 带标签示例的集合 (E^+, E^-) 唯一刻画 q 相对于 Σ2. q 拟合 (E^+, E^-) 并且 (E^+, E^-) 是相对于 Σ的有限同态对偶。

证明。从 1 到 2，若 (E^+, E^-) 关于 Σ 唯一刻画 q，则根据定义，q 拟合 (E^+, E^-)。此外，由此可知，E^+ 中不存在带点实例同态映射到 E^- 中的带点实例（否则，根据合取查询的并的单调性可知，q 不拟合负例）。接下来，反设 (E^+, E^-) 关于 K 不是同态对偶。于是存在一个带点实例 (I, a)，满足 $I \models \Sigma$，它既不属于 $E^+\uparrow$，也不属于 $E^-\downarrow$。令 q_1 为典范

E^+ 的合取查询的并，并令 q_2 为 $E^+ \cup$ 的典范合取查询的并$\{(I,\mathbf{a})\}$。那么 q_1 与 q_2 不等价，且二者均拟合 (E^+, E^-)，矛盾。

从 2 到 1，令 q' 为任意一个拟合 (E^+, E^-) 的合取查询的并。我们必须证明，关于 Σ，q' 与q 等价。考虑任意一个满足 $I \models \Sigma$ 的带点实例 $(I,\mathbf{a})$。如果 $\mathbf{a} \in q(I)$，那么 $(I,\mathbf{a}) \in E^+\uparrow$，因此 $\mathbf{a} \in q'(I)$。另一方面，如果 $\mathbf{a} \notin q(I)$，那么 $(I,\mathbf{a}) \notin E^+\uparrow$，于是 $(I,\mathbf{a}) \in E^-\downarrow$，进而 $\mathbf{a} \notin q'(I)$。◀

▶**定理 7.1**。令 $\boldsymbol{S}$ 为一个模式，Σ 为一个具有广义右伴随的有限弱无环 *TGD* 集。那么，$\boldsymbol{S}$ 上的每个 *c*-无环合取查询 q，关于 Σ，都可以由满足 Σ 的有限个带标签示例唯一刻画。

证明。令 E^+ 为所有带标记实例 $(P_\Sigma(I), \mathbf{a})$ 的集合，其中 $(I, \mathbf{a})$ 是 q 中合取查询的（c-无环）典范实例。由定理 6.7，存在一个有限集 E^-，使得 (E^+, E^-) 是关于 Σ 的态射对偶性。由引理 E.1 可知，(E^+, E^-) 关于 Σ 唯一刻画 q。◀

▶**定理 7.2**。设 $\boldsymbol{S}$ 是一个模式，Σ 是一个具有广义右伴随且有限可控的有限元组生成依赖集。那么，$\boldsymbol{S}$ 上的每个 *c*-无环合取查询的并 q 都可以由一组有限的带标签 *ABox* 示例关于 Σ 唯一刻画。

证明。令 E^+ 为 q 中所有（c-无环）合取查询的典范实例集合。由定理 6.10，存在一个有限集 E^-，使得 (E^+, E^-) 是范畴 $ABox_\Sigma[\mathbf{S}]$中的有限对偶。我们断言，(E^+, E^-)作为带标签 ABox 示例的集合，可以关于 q Σ唯一刻画。证明与引理 E.1 的证明类似：

从构造可以清楚看出，q拟合 E^+。任取一个$(I, \mathbf{a}) \in E^-$。由于(E^+, E^-)是有限对偶，因此不存在$(J, \mathbf{b}) \in E^+$使得$(J, \mathbf{b}) \to_\Sigma (I, \mathbf{a})$。等价地，不存在$(J, \mathbf{b}) \in E^+$使得$(P_\Sigma(J), \mathbf{b}) \to (P_\Sigma(I), \mathbf{a})$，因此不存在$(J, \mathbf{b}) \in E^+$使得$(J, \mathbf{b}) \to (P_\Sigma(I), \mathbf{a})$。由于 E^+ 由 q的典范合取查询组成，这意味着$\mathbf{a} \notin q(P_\Sigma(I))$。换言之，$q$ 拟合负例 E^-。最后，令 q' 为任意一个拟合(E^+, E^-)的合取查询的并，并令$(I, \mathbf{a})$为满足 Σ的任意实例。如果$\mathbf{a} \in q(I)$，则可知，对于该合取查询的典范查询$(J, \mathbf{b})$，有$(J, \mathbf{b}) \to_\Sigma (I, \mathbf{a})$；由此可知（因为 q'拟合 E^+），$\mathbf{a} \in q'(P(I))$，进而（因为 $P(I) \leftrightarrow_{adom(I)} I$），$\mathbf{a} \in q'(I)$。另一方面，如果$\mathbf{a} \notin q(I)$，则由对偶性可知，对于某个$(J, \mathbf{b}) \in E^-$，有$(I, \mathbf{a}) \to_\Sigma (J, \mathbf{b})$。由此可知（因为 q'拟合 E^-），$\mathbf{a} \notin q'(P(I))$，进而（因为 $P(I) \leftrightarrow_{adom(I)} I$），$\mathbf{a} \notin q'(I)$。◀

F TAM Datalog 的表达能力完备性（定理 3.9 的证明）

固定一个模式 $\mathbf{S}$，并令 $\mathbf{X} = \{X_1, \ldots, X_n\}$（我们可以将其视为由一元谓词组成的模式）。按照 [13]，我们考虑由以下算子归纳定义的形式“树项”集合。

- 对于每个 $S \subseteq \mathbf{X}$，$\bullet_S$ 都是一个树项。对于每个 $R \in \mathbf{S}$，对于所有满足 $k = arity(R)$ 的树项 $t_1, \ldots, t_k$，以及对于每个 $i \in [k]$，$\blacktriangledown_i^R(t_1, \ldots, t_k)$ 都是一个树项。我们为每个树项 t 归纳定义一个与之关联的带点树 $(T(t)\ r(t))$，如下所示。如果 $t = \bullet_S$，那么 $T(t)$ 是仅包含一个节点 v 的树（因此 $r(t) = v$），并且对每个 $X_i \in S$，都包含事实 $X_i(v)$。
-

XX：30 Datalog 程序的右伴随，以及受限类上的同态对偶

- 如果 $t = \blacktriangledown_i^R(t_1, \ldots, t_k)$，那么 $T(t)$ 是通过对 $T(t_1) \ldots, T(t_k)$ 作不相交并，并添加事实 $f = R(r(t_1) \ldots, r(t_k))$ 得到的树。此外， $r(t) = r(t_i)$。

▶ **引理 F.1**。 对于每个有限连通无环的带指针 $(\boldsymbol{S} \cup \boldsymbol{X})$-实例 (I, a)，存在一个树项 t，使得 $(T(t)\ r(t))$ 与 (I, a) 同构。

证明。 证明对实例 I的规模进行归纳，规模按**S**-事实的数量计算。归纳的基例是 I 不包含任何 **S**-事实的情形。在这种情况下，由连通性可知， I 必定是一个只包含某些**X**-事实的单元素结构。在此情形下，结论显然成立：只需取 t 为项 $\bullet_S$，其中 S 是出现在 I中的全部 $X_i \in \mathbf{X}$ 的集合。

如果 I 包含 n 个 S-事实，且有 $n > 0$，那么根据连通性， a 必定出现在至少一个 S-事实中，即 I包含一个形如 $R(a_1, \ldots, a_n)$ 的事实，其中不妨设 $a_i = a$。令 I'为从 I中移除事实 $R(a_1, \ldots, a_n)$ 后得到的子实例。对于每个 $j \leq n$， 令 I_j为包含 a_j的 I' 的连通分量。根据归纳法，存在一个项 t_j ，使得 $(T(t_i)\ r(t_i))$ 与 (I_j, a_j) 同构。令 $t = \blacktriangledown_i^R(t_1, \ldots, t_n)$。于是容易看出，$(T(t)\ r(t))$ 与 (I, a) 同构。◀

- 在此处，自动机是指一个元组 (**S**、**X** Q、 F、 δ)，由以下部分组成：模式**S**、**X**。一个 Q 状态的有限集，其中有一个特殊子集 $F \subseteq Q$对于每个 o 形如 $\bullet_S$ 或 $\blacktriangledown_i^R$的运算符，
- 其元数为 r （其中我们将 $\bullet_S$ 视为零元运算），都有一个转移关系 $\delta_o \subseteq Q^r \times Q$
-

接受的定义正如预期。若我们可以为树项 t 的每个子项 t' 关联一个状态$q_{t'}$ ，使得 $q_t \in F$，且映射 $t' \mapsto q_{t'}$ 遵循转移关系（也就是说，如果 $t' = o(t'_1$、……、 $t'_r)$，那么 $(q_{t'_1}$、……、 $q_{t'_r}$、 $q_{t'}) \in \delta_o$，则称该树项被接受。

给定一个带有模式 **S** 的单子二阶逻辑公式 $\phi(x_1$、……、 $x_n)$，我们将考虑以下与之关联的公式 ϕ'，其定义为带有模式 $\mathbf{S} \cup \mathbf{X}$ 的单子二阶逻辑句子，定义如下

$$\exists x_1, \ldots, x_n (\phi(x_1, \ldots, x_n) \wedge \bigwedge_{i=1 \ldots n} X_i(x_i)) \qquad \text{(Eq. 4)}$$

▶**引理 F.2.** 如果 ϕ 是单调的，那么 ϕ' 也是单调的。

证明。假设 h： $A \to B$，其中 A 和 B 是 S ∪ X-实例，并且假设 A满足 ϕ'。令 $x_i \mapsto a_i$ 为见证该结论的实例化。由于 X 的谓词没有出现在 ϕ 中，因此 A 的 S-约化满足 $\phi(a_1, \ldots, a_n)$。由于 ϕ 是单调的，因此 B 的 S-约化满足 $\phi(h(a_1) \ldots, h(a_n))$。由于对每个$1 \leq i \leq n$ 都有 $h(a_i) \in B^{X_i}$，因此 B 满足 ϕ'。◀

▶ **定理 F.3.** 设 ϕ' 是一个模式为 $\boldsymbol{S} \cup \boldsymbol{X}$ 的 *MSO*句子。那么存在一个有限自动机，它接受所有满足 ϕ' 的树项 t 的集合，其中 $T(t)$ 。

证明。 证明过程完全是标准的。为求完整起见，我们详细说明构造，但省略正确性论证。按照自动机理论和单子二阶逻辑文献中的惯例，我们将通过假设单子二阶逻辑公式具有句法范式来简化处理，其中所有量化都是二阶的。更准确地说，我们考虑由以下形式的原子公式构成的公式

- $R(X_1$、……、 $X_n)$，作为 $\exists x_1$、……、 $x_n(R(x_1$、……、 $x_n) \wedge X_1(x_1) \wedge \cdots \wedge X_n(x_n))$ 的简写，$X_1 \subseteq X_2$作为 $\forall y(X_1(y) \to X_2(y))$ 的简写，并且
-

- $Singleton(X)$作为使用析取、否定和存在二阶量化的 $\exists x(X(x) \wedge \forall y(X(y) \rightarrow y = x))$ 的简写。为上述每个原子公式构造一个自动机很容易。连接词通过以下关于自动机的标准闭包运算处理：

- 两个自动机 $(S, X, Q^i, F^i, \delta^i)$ $i = 1, 2$（假设 Q^1 和 Q^2 不相交）的并是自动机 $(S, X, Q^1 \cup Q^2, F^1 \cup F^2, \delta)$，其中 $\delta_o = \delta^1_o \cup \delta^2_o$。自动机 (S, X, Q, F, δ) 的补是（确定性）自动机 $(S, X, 2^Q, F', \delta')$，其中 $F' = \{Q' \subseteq Q \mid F \cap Q' = \emptyset\}$，并且 $(Q_1, \ldots, Q_r, Q_{r+1}) \in \delta'_o$ 当且仅当 $Q_{r+1} = \{q \in Q \mid (q_1, \ldots, q_r, q) \in \delta_o$ 对于某个 $q_1 \in Q_1, \ldots, q_r \in Q_r\}$。将 (S, X, Q, F, δ) 投影到 $X' \subseteq X$ 定义为 (S, X', Q, F, δ')，其中 δ' 通过如下方式修改 δ 得到。对于每个 $S' \subseteq X'$, $\delta'_{\bullet_{S'}} = \bigcup_{S \cap X' = S'} \delta_{\bullet_S}$.◀

▶ **定理 F.4**。 设 $A = (\boldsymbol{S}, \boldsymbol{X}, Q, F, \delta)$ 是一个自动机。存在一个连通的布尔单子树形 *Datalog* 程序 P，其头部为 $\boldsymbol{S}^P_{in} = \boldsymbol{S} \cup \boldsymbol{X}$，使得对于所有 $(\boldsymbol{S} \cup \boldsymbol{X})$-实例 I，以下两者等价：**1.** $Ans() \in P(I)$。**2.** 存在某个树项 t 被 A 接受，使得 $T(t) \rightarrow I$。

证明。 对于每个状态 q，$\mathbf{S}^P_{aux}$ 都有一个一元符号 E_q。下面描述 P中的规则：

- 对于每个 $o = \bullet_S$ 以及每个 $q \in \delta_o$， Σ^P 都包含一条头部为 $E_q(x)$ 的规则，其体包含每个 $X_i \in S$ 对应的 $X_i(x)$。对于每个 $o = \blacktriangledown^R_i$以及每个 $(q_1, \ldots, q_k, q) \in \delta_o$， Σ^P包含该规则

$$E_q(x_i) :- R(x_1, \ldots, x_k), E_{q_1}(x_1), \ldots, E_{q_k}(x_k)$$

- 对于每个 $q \in F$，我们引入规则

$$Ans() :- E_q(x)$$

令 I 为任意一个（$\mathbf{S} \cup \mathbf{X}$）-实例。该构造的正确性可由以下断言得出：

▷ 断言。对于每个 $a \in adom(I)$ 和 $q \in Q$，以下两者等价：**1.** $E_q(a) \in P(I)$ **2.** 存在某个树项 t，使得 (i) $(T(t)\ r(t)) \rightarrow (I, a)$，并且 (ii) 存在一个$A$ 在输入 t 上的运行，该运行最终到达状态 q

证明从略，因为这是相当标准的。(1) → (2) 方向通过对推导长度进行归纳证明，而 (2) → (1) 方向则通过对 t 进行结构归纳证明。◀

▶ **定理 3.9**。 设 $\phi(x_1, \ldots, x_n)$ 是一个模式 $\boldsymbol{S}_{in}$ 上的单子二阶逻辑公式。以下a者等价：**1.** ϕ 可由 *TAM Datalog* 程序定义，**2.** ϕ 可由树形 *Datalog* 程序定义，**3.** ϕ是树决定的。

证明。 由 1 推出 2 是显然的。由 2 推出 3 可由引理 3.8 和引理 2.4 立即得到。接下来，我们证明 (3) →(1)。

假设 $\phi(x_1, \ldots, x_n)$ 是一个关于 $\mathbf{S}_{in}$ 且满足 (3) 的 MSO 公式。设 R 是一个新的二元关系符号，且不属于 $\mathbf{S}_{in}$。特别地， R 不出现在 ϕ 中。令 $\mathbf{S} = \mathbf{S}_{in} \cup \{R\}$。为证明接下来的步骤，我们将把 ϕ 视为一个

关于 S 的公式。令 ϕ' 为根据式 4 定义的关于 $\mathbf{S} \cup \mathbf{X}$ 的 MSO 句子（其中 $\mathbf{X} = \{X_1, \ldots, X_n\}$）。令 A 为定理 F.3 中与 ϕ' 对应的自动机，令 P 为定理 F.4 中的布尔连通树形单调 Datalog 程序，并令 P' 为定理 3.5 中与 P 对应的几乎单子式程序。检查定理 3.5 的证明可知，树形性得以保留，因此 P' 是一个连通 TAM Datalog 程序。

▷ 断言 2. P' 在连通 S-实例上与 ϕ 等价。

证明。假设 P' 在连通 S-实例 I 上产生 $Ans(a_1, \ldots, a_n)$。令 $\widehat{I}$ 为将 $Q_1(a_1) \ldots, Q_n(a_n)$ 加入 I 后扩展得到的连通 (S ∪X)-实例。那么，由此可得 $\widehat{\ }P(I) = true$。于是，存在某个被 A 接受的树项 t，使得 $T(t) \to I$。由此 $\widehat{\ }$可知，$T(t)$ 满足 ϕ'。因此，我们有 I 满足 ϕ'。由此可得 I 满足 $\phi(a_1, \ldots, a_n)$。请注意，在这一方向上，我们并未使用树确定性的完整条件，只用到了单调性。

反过来，假设 I 满足 $\phi(a_1, \ldots, a_n)$。那么根据 (3)

$$(J, b_1, \ldots, b_n) \to (I, a_1, \ldots, a_n)$$

$\widehat{\ }$对于某些 J 和 $b_1, \ldots\ldots, b_n$，使得 J 满足 $\phi(b_1, \ldots\ldots, b_n)$。令 J 为一个扩展了 J 的（$\mathbf{S} \cup \mathbf{X}$）-实例，其中包含 $Q_1(b_1), \ldots\ldots, Q_n(b_n)$。令 t 为一个树项，使得根据引理 F.1，$T(t)$ 与 J 同构，$\widehat{\ }$。由此可知，A 接受 t。因此，$P(I) = true$。由此可知，$Ans(a_1, \ldots\ldots, a_n)$ 属于 $P'(I)$。◁

最后，令 P'' 为从 P' 删除关系 R（它未出现在 ϕ 中）的所有出现后得到的 TAM Datalog 程序，具体做法是从 P' 的每条规则的体中删除该关系的所有出现。删除 R 的所有出现可能会导致某些规则不安全。也就是说，规则头部中出现的一个或多个变量 x 可能不再出现在体中。不过，这很容易修复：在规则体中添加一个包含 x 的 S_{in}-原子，并在该原子的所有其他位置使用新变量（有多种添加方式，我们将所有能以这种方式得到的安全规则都添加进去）。于是，根据断言 2，P'' 与 ϕ 等价：任取一个 S_{in}-实例 I，并令 I' 为一个扩展了 I 的 S-实例，其中包含 $adom(I)$ 上所有可能的 R-事实。那么，$I \models \phi(a_1, \ldots, a_n)$ 当且仅当 $I' \models \phi(a_1, \ldots, a_n)$，当且仅当 $Ans(a_1, \ldots, a_n) \in P'(I')$，当且仅当 $Ans(a_1, \ldots, a_n) \in P''(I)$。◀

G Pultr 函子作为 ∃Datalog 的一个特例

在本附录中，我们将说明 Pultr 函子可以视为 ∃Datalog 程序的一个特例。为便于阐述，我们遵循 [18]，仅考虑有向图函子。不过，以下论证同样适用于任意关系结构，如 [13] 中所示。

回顾一下，一个 k 元 Pultr 函子（针对有向图）由一对 $F = (\phi_V, \phi_E)$ 指定，其中 ϕ_V 是一个元数为 k 的合取查询，而 ϕ_E 是一个元数为 $2k$ 的合取查询。假设这两个合取查询都定义在签名 $\mathbf{S} = \{V, E\}$ 上，其中 V 是一元关系，E 是二元关系。给定一个有向图 $G = (V, E)$，$F(G)$ 是一个有向图，其顶点是所有满足 ϕ_V 的 k 元组 $\mathbf{a} \in adom(V)^k$，边则是所有满足 ϕ_E 的序对 $(\mathbf{a},\mathbf{b})$。

▶ 例 G.1. 弧图函子定义为 $F = (\phi_V, \phi_E)$，其中 $\phi_V(x, y) = E(x, y)$，且 $\phi_E(x, y, y, z) = E(x, y) \wedge E(y, z)$。那么，对于每个图 G，$F(G)$ 都是一个图，其节点集由 G 的边构成，并且当且仅当 $v = u'$ 时，存在一条边连接 (u, v) 和 (u', v')。弧图函子的右伴随的存在性已在 [22] 中得到应用，以提升近似图着色的当前最佳水平。

由下一个命题可知，每个 Pultr 函子都可以由一个∃Datalog 程序模拟。

▶ **命题 G.2**。对于每个 *Pultr*函子 F（ϕ_V、ϕ_E），存在一个弱无环 ∃*Datalog* 程序 P，使得对于所有有向图 I，P（I）$\leftrightarrow_{adom(I)} F$（$I$）。

证明。令 $F = (\phi_V, \phi_E)$，其中 ϕ_V分别为 k元和 ϕ_E $2k$元。取 $P = (\mathsf{S}_{in}、\mathsf{S}_{out}、\mathsf{S}_{aux}、\Sigma)$，其中 $\mathsf{S}_{in} = \{V_{in}、E_{in}\}$、$\mathsf{S}_{out} = \{V_{out}、E_{out}\}$、$\mathsf{S}_{aux} = \{R_1、\ldots、R_k\}$，且 Σ 由以下部分组成：

$$\begin{array}{lll} \exists y \bigwedge_{i=1\ldots k} R_i(y, x_i) & :- & \phi_V(x_1, \ldots, x_n) \\ E(u, v) & :- & \phi_E(x_1, \ldots, x_k, y_1, \ldots, y_k), \bigwedge_{i=1\ldots k}(R_i(u, x_i) \wedge R_i(v, y_i)) \end{array}$$

容易看出，对于所有有向图 I，$P(I) \leftrightarrow F(I)$。事实上，由于 $P(I)$ 和 $F(I)$ 均不包含来自 $adom(I)$ 的任何元素，因此有 $P(I) \leftrightarrow_{adom(I)} F(I)$。◀