arXiv:2304.10381v1 [cs.LO] 20 Apr 2023

强化版 PDL：关于带交与逆的 PDL 表达性扩展

DIEGO FIGUEIRA,波尔多大学、法国国家科学研究中心、波尔多国立综合理工学院、LaBRI、UMR 5800，法国塔朗斯 F-33400SANTIAGO FIGUEIRA,布宜诺斯艾利斯大学、FCEN、DC 与 CONICET-UBA、ICC，阿根廷
EDWIN PIN, 布宜诺斯艾利斯大学、FCEN、DM 与 CONICET-UBA、ICC，阿根廷

我们引入CPDL⁺，这是一个植根于命题动态逻辑（PDL）的表达性逻辑系列。在表达能力方面，CPDL⁺ 严格强于加入交与逆扩展的 PDL （又称ICPDL），也严格强于合取查询（CQ）、合取正则路径查询（CRPQ）及其某些已知扩展（正则查询和 CQPDL）。我们研究CPDL⁺的表达能力、通过双模拟刻画的不可区分性、可满足性和模型检测。

我们论证，可以根据公式底层图的树宽来定义 CPDL⁺ 的自然子类。我们证明，树宽为 2 的 CPDL⁺ 公式类与 ICPDL 等价，且也与树宽为 1 的 CPDL⁺ 公式类一致。然而，树宽超过 2 后，树宽每增加一级，表达能力都会严格增强。我们通过带卵石的双模拟博弈刻画每个固定树宽公式类的表达能力。基于这一刻画，我们证明 CPDL⁺ 具有树状模型性质。我们证明，固定树宽公式的可满足性问题可在双指数时间内判定，其复杂性与 ICPDL 相同。我们还给出了一些可满足性问题可简化至指数时间的类。最后，我们证明固定树宽公式的模型检测问题属于多项式时间，这与完整的 CPDL⁺ 类相反。

☞ 此 PDF 包含内部链接：点击概念即可跳转至其定义。[1]
☞ 本文是 [9]的完整版。

1 引言

本文可以被视为一项概念验证。由于 Kripke结构 和 图数据库（即，二元签名上的结构）本质上是相同的：我们能否将（针对Kripke 结构的）表达性逻辑与（针对 图数据库的）查询语言融合为一个行为良好的统一框架?

我们的出发点是命题动态逻辑，或 PDL，并对其扩展以包含交和逆。这是一种经过深入研究的逻辑，也是 Kripke 结构上表达能力最强的可判定逻辑之一。但另一方面，我们希望能够检验 合取查询，或更一般的 CRPQ；它们是 图数据库查询语言的基本构件。其结果是一种可以说既自然又具有高度表达力的逻辑，它继承了 PDL 和 CRPQ的所有良好的计算性质与模型论性质，同时保留了PDL 的风格，但允许更丰富的“合取”测试。我们将这种逻辑命名为“CPDL⁺”。

命题动态逻辑 最初被构想为一种用于程序推理的逻辑 [10]。然而，命题动态逻辑 的各种变体如今已应用于计算机科学的多个领域，尤其是描述逻辑、认知逻辑、程序验证以及数据集查询（更多应用见 [11, 19] ）。命题动态逻辑 的一个研究最广泛的扩展是加入逆向导航、程序交和程序补。尤其是，加入逆和程序交运算符后，便得到一种性质良好的逻辑，称为ICPDL （其中‘I’代表交，‘C’代表逆）。ICPDL 的可满足性问题可在多项式时间内进行模型检测，并具有“类树”模型性质

[1] https://ctan.org/pkg/knowledge

作者地址：Diego Figueira，diego.figueira@cnrs.fr，波尔多大学、法国国家科学研究中心、波尔多国立综合理工学院、LaBRI、UMR 5800，F-33400 塔朗斯，法国，F-33400；Santiago Figueira，santiago@dc.uba.ar，布宜诺斯艾利斯大学、FCEN、DC 与 CONICET-UBA、ICC，阿根廷；Edwin Pin，epin@dc.uba.ar，布宜诺斯艾利斯大学、FCEN、DM 与 CONICET-UBA、ICC，阿根廷。

[11]。此前研究过一种更简单的程序交扩展，即在CPDL 中加入程序循环，它可以表达程序从某一点出发并最终返回该点，这种逻辑称为loop-CPDL，其表达能力严格弱于ICPDL。然而，加入程序补后，会得到一种可满足性问题高度不可判定的逻辑 [11, 16], ，尽管其模型检测仍可在多项式时间内完成 [19]。

在这项工作中，我们探讨了一类逻辑，其中尤其包括对程序交的推广。直观地说，两个 PDL 程序 π 和 π' 的交，可以看作是在由程序π 和 π' 表示的二元关系上，两个原子 $R_\pi(x,y)$ $\wedge R_{\pi'}(x,y)$的合取（即本质上是一个合取查询）。基于这一思路，我们引入了 CPDL 的一种扩展，称为 CPDL$^+$ ，它允许对任意数量的原子进行测试。例如，一个公式可以测试是否存在由 $\bigwedge_{1\le i<j\le n} R_\pi(x_i,x_j)$ 相关元素构成的 n-团。除此之外，这些测试还可以像传统的 PDL 程序一样嵌套、组合或迭代。从表达能力的角度来看，所得逻辑颇具吸引力：它不仅能够表达 ICPDL，还能表达近来为寻找良性且表达力强的查询语言而研究的几种图数据库查询语言。这些语言包括 C2RPQ、正则查询和 CQPDL。图 1 展示了 CPDL$^+$ 所处位置的概况。

此外，CPDL$^+$ 的子类可以通过限制这类新测试所允许的底层图（也称为盖夫曼图）来自然地定义。因此，CPDL$^+(\mathcal{G})$ 是将测试限制为其底层图属于图类 $\mathcal{G}$的情形。特别地，对于合适（且非常简单）的类，我们得到ICPDL 和 loop-CPDL。

1.1 结果概述

表达能力。如前所述，CPDL$^+$ 包含一些先前研究过的图数据库查询语言（见图1）。此外，根据图的类$\mathcal{G}$ ，CPDL$^+(\mathcal{G})$ 可以对应于不同的先前形式体系。特别是，CPDL 在空类上对应于CPDL$^+(\emptyset)$。

在第 4 节中，我们另一方面证明，逻辑 loop-CPDL 也可以表示为 CPDL$^+(\mathrm{G}_{\circlearrowleft})$，其中 $\mathrm{G}_{\circlearrowleft}$ 是仅包含一个具有自环顶点的图的类（命题 2）。不难看出，ICPDL 和 CPDL$^+(\mathrm{G}_{\cap})$ 的表达能力相同，其中 $\mathrm{G}_{\cap}$ 是仅包含一个图的类，该图含有两个顶点以及连接它们的一条边。此外，我们还证明了 ICPDL 与 CPDL$^+(\mathrm{TW}_2)$ 的表达能力相同。因此，CPDL$^+(\mathrm{TW}_1)$、CPDL$^+(\mathrm{TW}_2)$、CPDL$^+(\mathrm{G}_{\cap})$ 和 ICPDL 的公式和程序分别定义相同的一元和二元关系（定理 1）。

然而，正如我们在第 5 节所示，TW_2其表达能力开始形成无限层次结构，其中 CPDL$^+$ (TW_k) 在表达能力上严格包含于 CPDL$^+(\mathrm{TW}_{k+1})$，对每个 $k \ge 2$ （定理 4）。

不可区分性。我们通过一个k-棋子局部一致性博弈来刻画 CPDL$^+(\mathrm{TW}_k)$ 的模型不可区分性。也就是说，我们研究一对模型在什么条件下无法被 CPDL$^+(\mathrm{TW}_k)$ 中的公式或程序区分。在第 5 节中，我们分别通过正公式（定理 2）和任意公式（定理 3）刻画不可区分性。这些概念在某种意义上对应于将模拟关系从模态逻辑提升到 CPDL$^+(\mathrm{TW}_k)$（不过我们将这一联系留待附录 C.3 讨论）。我们证明 CPDL$^+(\mathrm{TW}_k)$ 具有 TW_k-模型性质：如果 CPDL$^+(\mathrm{TW}_k)$ 中的公式可满足，那么它在一个树宽为 k 的模型中可满足（推论5）。

可满足性。在第 7节中，我们研究可满足性问题，即对于给定的公式，是否存在一个模型和其中的一个元素使其成立。我们证明，对于任意 k，CPDL$^+(\mathrm{TW}_k)$ 的可判定性处于2指数时间，即，其复杂性与ICPDL 和loop-CPDL [12]相同。我们还确定了

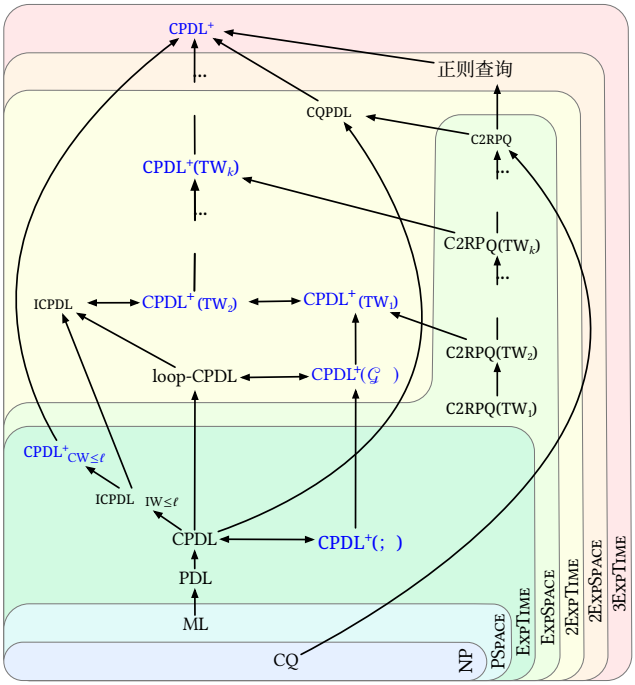

图 1。PDL 类和 CQ 类形式体系的表达能力与复杂性全景图。箭头指向表达能力更强的语言，除从正则查询出发的箭头为指数时间外，所有箭头均通过多项式时间翻译给出。对于查询语言 CQ、C2RPQ、正则查询和 CQPDL，为了能够比较表达能力，我们将其限制为元数为 1 的连通查询。用 C2RPQ(TW$_k$) 表示底层图的树宽为 $\leq k$ 的（连通）C2RPQ。对于任意 ℓ，ICPDLIW$\leq\ell$ 是所有满足 IW(φ) $\leq \ell$ 的 ICPDL 公式 φ 的集合，该集合在[12] 中引入，并在第 7.2 节定义。对于任意 ℓ，CPDL$^+$CQ$\leq\ell$ 是所有满足 CW(φ) $\leq \ell$ 的 CPDL$^+$ 公式 φ 的集合，定义见第 7.2 节。复杂性对应于每种形式体系的基本推理问题：对于在否定下封闭的逻辑，是可满足性；对于 CQ/C2RPQ 系列的查询语言，则是包含性。我们用蓝色突出显示本文引入的逻辑系列。

一种基于我们所称的合取宽度而非树宽的层次结构，从而使得任何具有有界合取宽度的类均可在指数时间内判定，即其复杂性与CPDL相同。对于不受限制的逻辑，即CPDL$^+$，我们得到了一个3指数时间上界（定理7）。这些结果利用了上述CPDL的有界树宽模型性质$^+$ 。

模型检测。最后，我们将在第8 节论证，对于任意有界树宽的CPDL$^+$(TW$_k$) 类，在有限结构上的模型检测仍可在多项式时间内完成；这与不受限的CPDL$^+$ (定理8)逻辑不同。

所有缺失的证明均收录于附录中。

1.2 相关逻辑

如前所述，CPDL$^+$ 可视为一种“伞式逻辑”，涵盖了命题动态逻辑的表达性扩展，例如ICPDL，以及在合取查询领域中研究的表达性扩展图数据库。接下来，我们将详细介绍一些与ICPDL$^+$相关的查询语言。

图数据库通常抽象为有限的边标记图，其中标签取自某个有限的关系字母表（更多详情参见例如，[2]），但只需进行简单调整，就可以在Kripke结构上使用图查询语言结构，反之亦然。

合取查询（CQs）是关系原子的合取，并且在投影下封闭。例如，在边带标签的图上，合取查询“$q(x,x,y) = R(x,y) \wedge S(y,z) \wedge T(z,x)$”返回所有满足 u,v 参与一个 R-S-T-环的顶点三元组（u,u,v）。

类比于 合取查询，合取正则路径查询，或称 *CRPQ*，允许使用形如 $x \xrightarrow{L} y$的原子，其中 L 是关系字母表上的一个正则表达式，其语义是：存在一条从 x 到 y 的路径，该路径读取 L [6]中的关系名称组成的词。该语言通常被视为任何合理图查询语言的骨干 [2]。而 合取双向正则路径查询，或称 *C2RPQ* 允许 L 在扩展后的字母表上构造，该字母表加入了关系的逆。

正则查询 [21], 也称为 嵌套正的 *2RPQ* [5], 可以被视为在克莱尼星、并集和逆运算下的闭包C2RPQ。其包含问题已被证明可以在初等时间内判定，这与对 C2RPQ [5, 22]的一些更具野心的推广相反。

或许在精神上与这项工作最接近的语言是 *CQPDL* [3], ，表示 合取查询，其原子关系可以由 CPDL 程序定义。它在 C2RPQs 的表达能力方面更强，但与 正则查询不可比较。

2 预备知识

我们使用标准记法 $\bar{u}$ 表示来自某个集合 X 的元素元组，并用 $\bar{u}[i]$ 表示其第 i 个分量中的元素。如果 $\bar{q} = (q_1, \ldots, q_k)$ 是一个元组且 $1 \le i \le k$，则我们用 $\bar{q}[i \mapsto r]$ 表示元组 $(q_1, \ldots, q_{i-1}, r, q_{i+1}, \ldots, q_k)$。对于关系 R, $S \subseteq X \times X$，我们定义复合运算 $R \circ S := \{(u,v) \mid \exists w. uRw, wSv\}$。对于函数 $f\colon X \to Y$，我们通常写作 $f(\bar{u})$ 来表示 $(f(\bar{u}[1]) \ldots, f(\bar{u}[k]))$，其中 k 是 $\bar{u}$ 的维数。我们用 $\wp(X)$ 表示 X 的子集所构成的集合。

（简单无向）图是一个元组 $G = (V, E)$，其中 V是顶点集， E是边集；每条边都是一个大小至多为二的顶点集。我们分别用 V和 E表示 $V(G)$ 和$E(G)$。除非另有说明，以下均假设图的顶点集是有限集。树是没有环的连通图（特别地，不存在形如 $\{v\}$ 的边）。

图 G的树分解是一个二元组 (T, v)，其中 T是一棵树，而 $\mathrm{v}: V(T) \to \wp(V(G))$ 是一个函数，它将 T的每个节点（称为袋）关联到 G的一个顶点集。当 $x \in \mathrm{v}(b)$ 时，我们称袋 $b \in V(T)$ 包含顶点 x。此外，它还必须满足以下三个性质：A. G的每个顶点 x都至少包含在 T的一个袋中；B. 对于 G中连接x和 y的每条边， T中至少有一个袋同时包含 x和 y；C. 对于 G的每个顶点 x， T中包含 x的袋构成 $V(T)$ 的一棵连通子树。

无限图的树分解是一个无限树以及满足上述条件 A、B、C 的映射。(T, v) 的宽度是 $|\mathrm{v}(b)| - 1$ 的最大值（如果 T 是无限的，则为上确界），其中 b 遍历 $V(T)$。 G 的树宽，记作 $\mathrm{tw}(G)$，是 G 的所有树分解的宽度中的最小值。我们将所有树宽至多为 k 的（有限）图的集合记为 TW_k。不同树宽的图见图 2（B 和 C），树分解的一个示例见图 4。

设 $\mathbb{P}$， $\mathbb{A}$和 $\mathbb{V}$ 分别为可数无限且两两不相交的原子命题、原子程序和变量集合。一个 *Kripke* 结构是一个元组

$$K = (X, \{\to_a \mid a \in \mathbb{A}\}, \{X_p \mid p \in \mathbb{P}\}) \tag{1}$$

其中 X是一个“世界”集合， $\to_a \subseteq X \times X$是每个 $a \in \mathbb{A}$对应的转移关系，而 $X_p \subseteq X$是每个 $p \in \mathbb{P}$ 对应的一元关系。两个 世界之间的距离是 K 的

底层无向图中的最短路径长度。若 K 的每个世界都只有有限个距离 1 的“邻居”，则称 K 具有有限度。我们用 $W(K)$ 表示 K 的世界集合 X。从结构 $K=(X,\{\to_a \mid a\in\mathbb{A}\},\{X_p \mid p\in\mathbb{P}\})$ 到另一个结构 $K'=(X',\{\to'_a \mid a\in\mathbb{A}\},\{X'_p \mid p\in\mathbb{P}\})$ 的同态是一个函数 $f\colon X\to X'$，满足：对于每个 $w,w'\in X$，若 $w\in X_p$，则 $f(w)\in X'_p$；若 $(w,w')\in\to_a$，则 $(f(w)\ f(w'))\in\to'_a$；以上对每个 $p\in\mathbb{P}$ 和 $a\in\mathbb{A}$ 均成立。若函数 $f\colon X\to X'$ 满足 $X\subseteq X$，则称其为部分同态，条件是它是从 $\hat{}K$ 到 K' 的同态，其中 $\hat{}K$ 是由 $\hat{}X$ 诱导的 K 的子结构。令 $\mathrm{Hom}_k(K,K')$ 为所有满足 $\{\bar{u}[i]\mapsto\bar{v}[i]\mid 1\le i\le k\}$ 是从 K 到 K' 的部分同态的 $(\bar{u},\bar{v})\in W(K)^k\times W(K')^k$ 的集合。我们常会滥用记号：对于维度为 $k'<k$ 的 $\bar{u}',\bar{v}'$，写作 $(\bar{u}',\bar{v}')\in\mathrm{Hom}_k(K,K')$，表示 $(\bar{u}',\bar{v}')\in\mathrm{Hom}_{k'}$。Kripke 结构的树宽是其底层（可能无限）图的树宽。

3 命题动态逻辑及其扩展

CPDL。*CPDL 表达式* 可以是 公式 或 φ 程序 π，由以下文法定义，其中 p 遍历 $\mathbb{P}$，而 a 遍历 $\mathbb{A}$：

$$\varphi ::= p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \langle\pi\rangle$$
$$\pi ::= \varepsilon \mid a \mid \bar{a} \mid \pi\cup\pi \mid \pi\circ\pi \mid \pi^* \mid \varphi?$$

我们将 程序 $[\![\pi]\!]_K$ 和 公式 $[\![\varphi]\!]_K$ 在 Kripke 结构 K （1）中的语义定义为，其中 $[\![\pi]\!]_K\subseteq X\times X$ 和 $[\![\varphi]\!]_K\subseteq X$ 如下：

$$[\![p]\!]_K := X_p \text{ for } p\in\mathbb{P},\qquad [\![\neg\varphi]\!]_K := X\setminus[\![\varphi]\!]_K,$$
$$[\![\varphi_1\wedge\varphi_2]\!]_K := [\![\varphi_1]\!]_K\cap[\![\varphi_2]\!]_K,$$
$$[\![\langle\pi\rangle]\!]_K := \{u\in X\mid\exists v\in X.(u,v)\in[\![\pi]\!]_K\},$$
$$[\![\varepsilon]\!]_K := \{(u,u)\mid u\in X\},\qquad [\![a]\!]_K := \to_a,$$
$$[\![\bar{a}]\!]_K := \{(v,u)\in X^2\mid(u\to_a v)\},$$
$$[\![\pi_1\star\pi_2]\!]_K := [\![\pi_1]\!]_K\star[\![\pi_2]\!]_K \text{ for } \star\in\{\cup,\circ\},$$
$$[\![\pi^*]\!]_K := \text{the reflexive transitive closure of } [\![\pi]\!]_K,$$
$$[\![\varphi?]\!]_K := \{(u,u)\mid u\in X, u\in[\![\varphi]\!]_K\}.$$

我们用 $K,u\models\varphi$ 表示 $u\in[\![\varphi]\!]_K$，用 $K,u,v\models\pi$ 表示 $(u,v)\in[\![\pi]\!]_K$；如果对于每个结构 K，$[\![\varphi_1]\!]_K=[\![\varphi_2]\!]_K$（分别为 $[\![\pi_1]\!]_K=[\![\pi_2]\!]_K$），则我们记作 $\varphi_1\equiv\varphi_2$（分别为 $\pi_1\equiv\pi_2$），在这种情况下，我们称 φ_1,φ_2（分别为 π_1,π_2）等价。命题动态逻辑是这样一个 CPDL 表达式片段：不使用形如 $\bar{a}$ 的原子程序。

ICPDL。ICPDL 定义为 CPDL 加上 程序交 $\pi ::= \pi\cap\pi$ 其语义由 $[\![\pi_1\cap\pi_2]\!]_K$：$=[\![\pi_1]\!]_K\cap[\![\pi_2]\!]_K$.

loop-CPDL。loop-CPDL 由 CPDL 加上 $\varphi ::= \mathrm{loop}(\pi)$ 定义，其语义由$[\![\mathrm{loop}(\pi)]\!]_K$：$=\{x\mid(x,x)\in[\![\pi]\!]_K\}$。

添加强化功能。“原子”是形如 $\pi(x,x')$ 的表达式，其中 π 是一个 CPDL$^+$ 程序，且 $x,x'\in\mathbb{V}$。对于原子 $\pi(x,x')$，我们定义 $\mathrm{vars}(\pi(x,x')):=\{x,x'\}$；对于原子集 C，我们定义 $\mathrm{vars}(C):=\bigcup_{A\in C}\mathrm{vars}(A)$。

<escape>我们将 CPDL$^+$ 定义为 CPDL 的一种扩展，它还允许使用 形如以下形式的程序： </escape>

$$\pi ::= C[x_s,x_t]$$

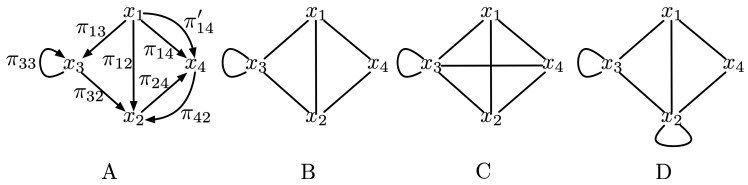

图 2. A. 集合 $C = \{\pi_{13}(x_1, x_3)\ \pi_{12}(x_1, x_2)\ \pi_{14}(x_1, x_4), \pi'14(x_1, x_4)\ \pi_{33}(x_3, x_3)\ \pi_{32}(x_3, x_2)\ \pi_{24}(x_2, x_4)\ \pi_{42}(x_4, x_2)\}$ 中所含原子的图形表示；B. $G_{C[x_1,x_2]}$；C. $G_{C[x_3,x_4]}$；D. $G_{C[x_2,x_2]}$。注意，$G_{C[x_1,x_2]}$、$G_{C[x_2,x_2]} \in$ 属于 TW2，但 $G_{C[x_3,x_4]} \in$ 属于 TW3 \ TW2。

其中：(1) C 是原子的有限集；(2) $x_s, x_t \in$ 变量集(C)[2]；(3) C 的底层图 G_C 是连通的，其中 G_C 定义为 $V(G_C)$ =变量集(C)，而 $E(G_C)$ ={变量集(A) | $A \in C$}。[3]我们称这些程序为合取程序。注意，$\{x_s, x_t\} \subseteq$ 变量集(C)，因此我们也定义变量集($C[x_s, x_t]$) := 变量集(C)。

函数 f: vars(C) $\to$ (K) 若满足以下条件，则称为一个 C-满足赋值：对于每个原子 $\pi'(x, x') \in C$，均有 $(f(x)\ f(x')) \in [\![\pi']\!]_K$。Kripke 结构$K$上合取程序的语义 $[\![C[x_s, x_t]]\!]_K$ 是所有满足以下条件的有序对 $(w_s, w_t) \in$ (K) $\times$ (K) 的集合：存在某个C-满足赋值 f，使得 $f(x_s) = w_s$ 且 $f(x_t) = w_t$。

对于一类图 G，我们将 CPDL$^+$(G) 定义为 CPDL$^+$ 的一个片段，其中的程序具有 G 所允许的某种形状。形式化地，对于任意合取程序 $\pi = C[x_s, x_t]$，我们将 $C[x_s, x_t]$ 的底层图 $G_{C[x_s,x_t]}$ 视为具有 $V(G_{C[x_s,x_t]}) = V(G_C)$ 和 $E(G_{C[x_s,x_t]})$ ={$\{x_s, x_t\}$} $\cup E(G_C)$。注意，在 $G_{C[x_s,x_t]}$ 中，x_s 和 x_t 总是通过一条边相连，但在 G_C 中则不一定。CPDL$^+$(G) 是 CPDL$^+$ 的一个片段，其中唯一允许的合取程序具有 $C[x_s, x_t]$ 这种形式，且 $G_{C[x_s,x_t]} \in$ G。底层图的示例见图 2。

我们称 CPDL$^+$($\mathcal{G}$) 的一个表达式是正的，如果它不包含任何形如 $\neg\psi$ 的子公式。

有时，我们需要使用扩展了程序交（$\cap$）的 CPDL$^+$，并将其记为 ICPDL$^+$。$\pi_1 \cap \pi_2$ 的语义与 ICPDL 的语义相同。注意，ICPDL 中的任何表达式也是 ICPDL$^+$（不包含合取程序）中的表达式，且二者具有相同的语义；因此，ICPDL$^+$ 扩展了 ICPDL。类似地，ICPDL$^+$(G) 定义为在 CPDL$^+$(G) 的基础上扩展程序交所得的逻辑。

命题 1。 ICPDL$^+$ 和 CPDL$^+$ 在表达能力方面是等价的。此外，对于任何包含由两个不同节点之间的一条边构成的图的图类 $\mathcal{G}$，我们有：ICPDL$^+$($\mathcal{G}$) 和 CPDL$^+$($\mathcal{G}$) 在表达能力方面也等价。

证明。 $\pi_1 \cap \pi_2$ 是等价于$\{\pi_1(x,\ y)\ \pi_2(x,\ y)\}[x, y]$。 □

对于 e 一个 ICPDL$^+$ 表达式，令 sub(e) 表示所有子表达式 的集合，其中 e 采用通常意义上的定义（定义见 §A.1）。

4 与 loop-CPDL和ICPDL

对于表达式由公式和程序构成的逻辑 L_1 和 L_2，如果存在一个翻译映射，将 L_1-公式映射为 L_2- 公式，并将 L_1- 程序映射为 L_2- 程序，且保持等价性，我们就称 L_2 至少与 L_1，具有同等的表达能力，记作 $L_1 \leqq L_2$, 。我们用 $L_1 \lneqq L_2$ 表示 $L_1 \leqq L_2$ 且 $L_2 \not\leqq L_1$。我们用 $L_1 \equiv L_2$ 表示 $L_1 \leqq L_2$ 且 $L_2 \leqq L_1$，；此时称 L_1 和 L_2 表达能力相同。

[2]请注意，x_s 和 x_t 可能是相同的变量，也可能是不同的变量变量。[3]我们将注意力限制在“连通的”合取程序这一事实对于我们的任何结果都不是必要的，但它将使我们能够轻松地将其与先前定义的逻辑联系起来，例如ICPDL 和loop-CPDL。

设 $\mathsf{G}_{\circlearrowleft}:=\{G\}$ 是仅包含一个图 $G=(\{v\},\{\{v\}\})$ 的类，该图仅有一条自环边；设 $\mathsf{G}_{\cap}:=\{G\}$ 是仅包含一个图 $G=(\{v,v'\},\{\{v,v'\}\})$ 的类，该图包含一条边。注意，$\mathsf{G}_{\circlearrowleft}$ 和 $\mathsf{G}_{\cap}$ 都包含于 TW_1。

接下来我们将展示，loop-CPDL $\equiv$ CPDL$^+(\mathcal{G}_{\circlearrowleft})$、以及 ICPDL $\equiv$ CPDL$^+(\mathcal{G}_{\cap})$ $\equiv$ CPDL$^+(\mathsf{TW}_1)$ $\equiv$ CPDL$^+(\mathsf{TW}_2)$ 之间存在多项式时间翻译。

4.1 loop-CPDL和CPDL$^+(\mathcal{G}_{\circlearrowleft})$ 具有相同的表达能力

从loop-CPDL 到CPDL$^+(\mathcal{G}_{\circlearrowleft})$ 的翻译，只需递归地将每个loop(π)的出现替换为 π不使用 loop 的$\{\pi(x,x)\}[x,x]$（详情见附录 B.1）。

反方向的翻译，即从 CPDL$^+(\mathsf{G}_{\circlearrowleft})$ 到 loop-CPDL，首先注意到每个合取程序都具有 $\{\pi_1(x,x)\ldots,\pi_n(x,x)\}[x,x]$ 的形式。如果该程序满足其中没有 π_i 包含合取程序，那么它就等价于 $(\text{loop}(\pi_1)\wedge\cdots\wedge\text{loop}(\pi_n))?$。因此，递归替换即可得到一个等价的 loop-CPDL 表达式（详情见附录 B.2）。

由这些翻译，我们进而得到：

命题 关于 2. loop-CPDL和 CPDL$^+(\mathcal{G}_{\circlearrowleft})$ 是 表达能力相同，通过多项式时间翻译实现。

4.2 ICPDL、CPDL$^+(\mathsf{TW}_1)$，以及CPDL$^+(\mathsf{TW}_2)$ 具有相同的表达能力

从 ICPDL 到 CPDL$^+(\mathcal{G}_{\cap})$。该翻译通过递归地将每个 $\pi_1\cap\pi_2$ 满足 π_1,π_2 不包含 $\cap$子表达式的出现替换为等价表达式 $\{\pi_1(x,y)\ \pi_2(x,y)\}[x,y]$。详情见附录 B.3。

<escape>命题3. ICPDL $\leqq$ CPDL$^+(\mathcal{G}_{\cap})$。</escape>

从 CPDL$^+(\mathsf{TW}_2)$ 到 ICPDL 的转换。我们先给出一个技术性结果。对于图 G 和$B\subseteq V(G)$，如果 B是 G中的团，我们称之为团，当且仅当 $\{\{x,y\}\mid x,y\in B,x\neq y\}\subseteq E(G)$。以下引理（其证明见附录）指出，在处理 CPDL$^+(\mathsf{TW}_n)$ 中的 $C'[x,y]$时，可以假设 $\mathsf{G}_{C[x_1,x_2]}$ 具有一个树宽为 $\leq n$ 的树分解，并且满足额外性质：每个包都是 $\mathsf{G}_{C[x_1,x_2]}$中的团。

引理 *1.* 给定一个 *CPDL$^+$(TW$_n$)*-合取程序 $C'[x,y]$，可以在多项式时间内计算出一个 *CPDL$^+$(TW$_n$)*-合取程序 $C[x,y]\equiv C'[x,y]$，使得 $vars(C)=vars(C')$，并得到 $G_{C[x,y]}$的一个树宽为 $\leq n$的树分解 (T,v)，满足对于任意 $b\in V(T)$，$v(b)$ 都是 $G_{C[x,y]}$中的团。

下一个引理是从CPDL$^+(\mathsf{TW}_2)$）到ICPDL的翻译的关键组成部分：

引理 2。 设 C是由以下形式的原子组成的有限集：$\pi(z_1,z_2)$，其中 π是一个ICPDL-程序，且$z_1,z_2\in\mathbb{V}$，G_C 是一个团，并且 $|\text{变量集}(C)|\leq 3$。则：

(1) 对于任意 x，$y\in$变量集(C)，$x\neq y$，存在一个 ICPDL-程序 $\pi_{C[x,y]}$使得 $\pi_{C[x,y]}\equiv C[x,y]$。

(2) 对于任意 $x\in$变量集(C)，存在一个 ICPDL-程序 $\pi_{C[x,x]}$使得 $\pi_{C[x,x]}\equiv C[x,x]$。

此外，这些转换可以在多项式时间内完成。

图 3 展示了引理 2 中 $\pi_{C[x,y]}$ 和 $\pi_{C[x,x]}$ 的一个示例。一般情况则更为复杂，因为 C 可能包含——如图 2.A 所示——针对相同变量 z_1、z_2 的多个原子 $\pi_1(z_1,z_2)\ldots\pi_n(z_1,z_2)$，也可能包含示例中未展示的 $\pi(z_1,z_1)$ 形式的原子。此外，有时还需要将程序的方向“反转”，以获得所需的 ICPDL 程序。引理 2 的证明推迟至附录。

利用引理2 可以给出这种翻译。出于技术上的考虑，为了简化ICPDL$^+(\mathsf{TW}_2)$ 表达式（而非CPDL$^+(\mathsf{TW}_2)$）翻译为ICPDL。

命题 4. ICPDL$^+(\mathsf{TW}_2)$ $\leqq$ 通过多项式时间翻译得到的 ICPDL。

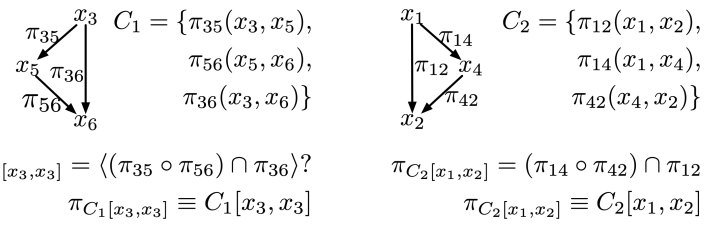

图 3。一个将仅由 ICPDL 原子组成的合取程序和 3 个变量翻译为ICPDL 的示例。 ICPDL-程序 $\pi_{C[x_3,x_3]}$和$\pi_{C[x_1,x_2]}$的名称即为引理2中使用的名称。

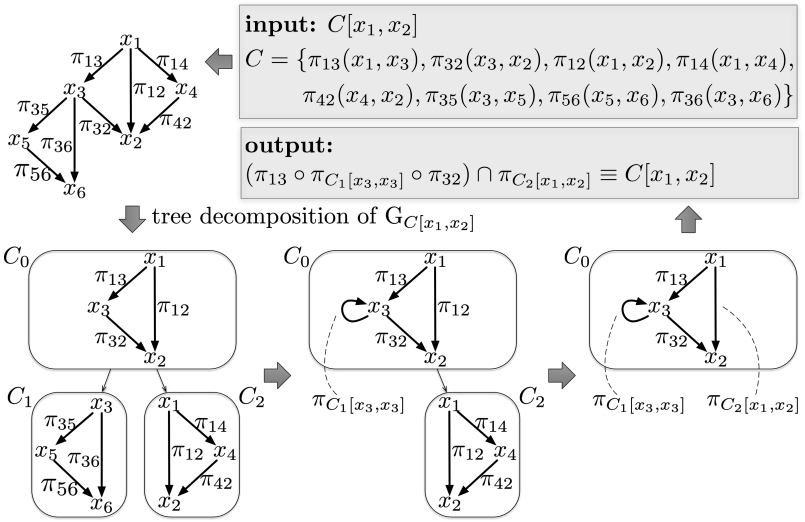

图 4。一个合取程序 $C[x_1, x_2]$，它仅由ICPDL 程序组成，以及从一个树宽为2 的树分解中构造语义等价的ICPDL公式的算法步骤，该树分解属于$G_{C[x_1,x_2]}$。圆角框表示此类树分解中的包，其中变量 x_1、……、 x_6 被分组（边和标签等附加信息不属于此树分解）。关于的定义，参见图$\pi_{C[x_3,x_3]}$3；。

图 4 展示了一个合取程序 $C[x_1, x_2]$ 的示例，其中 C 仅包含形如 $\pi(z_1, z_2)$ 的原子，其中 π 是一个 ICPDL 程序。给定 $G_{C[x_1,x_2]}$ 的一个树分解 (T, v)，其树宽至多为 2，且所有包都是 $G_{C[x_1,x_2]}$ 中的团（根据引理 1，不失一般性，可以假设这一点），可以逐步移除其叶子，直到只剩下一个包。每次移除一个叶子 B 时，所有仅使用 B 中变量的 C 中的原子都会被移除，但得益于引理 2，会添加一个包含 B 信息的 ICPDL 程序作为替代。一般情形则更加复杂，因为 C 可能包含一些并非 ICPDL 程序的程序 π（即合取程序）。在这种情况下，我们需要递归地消除这些 π，首先从仅包含 ICPDL 程序的那些开始。命题 4 的证明推迟至附录。

根据命题1、3 和4，以及$\mathcal{C}_\cap \subseteq \mathrm{TW}_1$, 这一事实，我们得到以下推论：

定理 1. ICPDL、CPDL$^+$（TW_1）和CPDL$^+$（TW_2）具有相同的表达能力，且可通过多项式时间转换实现。

5 不可区分性

在本节中，我们通过一种使用卵石的受限（双向）模拟形式，刻画 CPDL$^+$(TW$_k$) 的表达能力；这种形式类似于用于刻画有限变量一阶片段的 k-卵石博弈。

5.1 模拟关系

我们将通过一个双人零和图博弈来定义 k-模拟，其作用于成对的$(K,\ K')$ 个Kripke 结构之间G$[\rightharpoonup_k]$。该博弈的博弈场包含一个位置集合 $S \cup D$，其中

$$S = \{s\} \times Hom_k(K, K')$$

$$D = \{d_1, \ldots, d_k\} \times (W(K)^k \times W(K')^k)$$

其中，Spoiler 拥有从 S 出发的所有位置，而 Duplicator 拥有从 D出发的所有位置。G$[\rightharpoonup_k]$ 的移动集合是满足以下条件的最小集合：[4]

(1) 如果 $\bar{u}' = \bar{u}[i \mapsto w]$，则存在一条从 $(s, \bar{u}, \bar{v})$ 到 $(d_i, \bar{u}', \bar{v})$ 的移动，其中 w 是来自 K 的一个世界，与 $\bar{u}[j]$的距离为 ≤ 1，对于某个满足 $i \neq j$ 的 $1 \leq j \leq k$；以及 (2) 如果 $\bar{v}' = \bar{v}[i \mapsto w]$，则存在一条从 $(d_i, \bar{u}', \bar{v})$ 到 $(s, \bar{u}', \bar{v}')$ 的移动，其中 w 是来自 K' 的一个世界，与 $\bar{v}[j]$的距离为 ≤ 1，对于某个满足 $i \neq j$ 的 $1 \leq j \leq k$。

Duplicator 的获胜条件就是任意一场无限博弈，这是一种“安全条件”，意味着该博弈具有（位置）确定性。[5] 关于这类博弈的更多信息，请参阅 [14, 第 2]章。

给定两个 Kripke 结构 K, K'，以及元组 $\bar{v} \in W(K)^k$和 $\bar{v}' \in W(K')^k$，若 $K'\bar{v}'k$ 模拟 $K, \bar{v}$，记作 $K, \bar{v} \rightharpoonup_k K', \bar{v}'$，则满足以下条件：1) $(s, \bar{v}, \bar{v}')$ 是 G$[\rightharpoonup_k]$ 在 (K, K') 上的有效位置（即，$\bar{v}, \bar{v}'$ 诱导一个部分同态）；2) $\bar{v}$ 中的所有世界都位于 K 的同一个连通分量中；3) Duplicator 从 $(s, \bar{v}, \bar{v}')$ 出发有获胜策略。示例见图 5。

备注 1. 注意，G$[\rightharpoonup_k]$ 的任何移动都保持上述性质 *2*）。

定理 2. 设 $k \geq 2$。给定 *Kripke* 结构 K, K'，其中 K，具有有限度，且世界$u, v \in W(K)$ 和 $u', v' \in W(K')$，以下各项等价

(1) 对每个正的 CPDL$^+$(TW$_k$)-公式 φ，若 $K, v \models \varphi$，则 $K', v' \models \varphi$；以及 *(2)* $K, v \rightharpoonup_{k+1} K', v'$；

并且以下各项等价

(1) 对每个正的 CPDL$^+$(TW$_k$)-程序 π，若 $K, u, v \models \pi$，则 $K', u', v' \models \pi$；以及 *(2)* $K, u, v \rightharpoonup_{k+1} K', u', v'$。

此外，有限度的假设仅在1到2 的蕴含中需要。

证明。我们首先证明 1 到 2 的蕴含关系。首先指出，k-模拟关系与“存在型 k-鹅卵石博弈”密切相关，[17, 18] 该博弈作用于具有二元和一元签名的一阶结构上。已知 [1, 定理 4] 对于每一对 K, K' 有限 结构，如果对于每个布尔 合取查询 q，其 树宽 $< k$ 满足 $K \models q$ 则蕴含

[4]为便于记号，我们用 $(s, \bar{u}, \bar{v})$ 代替 $(s, (\bar{u}, \bar{v}))$，$(d_i, \bar{u}, \bar{v})$ 也同理。[5]这些博弈通常被描述为持续无限长。请注意，Spoiler 永远不会陷入“无路可走”的境地，因为他总能在分量 1 中放入分量 2 所包含的世界，等等。然而，Duplicator 可能会陷入无路可走的境地。无限时长的变体是添加一个由 Spoiler 控制的新位置“(*s*, Win)”，并添加从 (*s*, Win) 到 (*s*, Win) 的移动，以及从 D 中每个位置到 (*s*, Win) 的移动。这样，Duplicator 的获胜条件就是所有不经过位置 (*s*, Win) 的对局。由于这是一个安全条件（因此也是奇偶条件），该博弈的位置策略决定性得证 [8]。

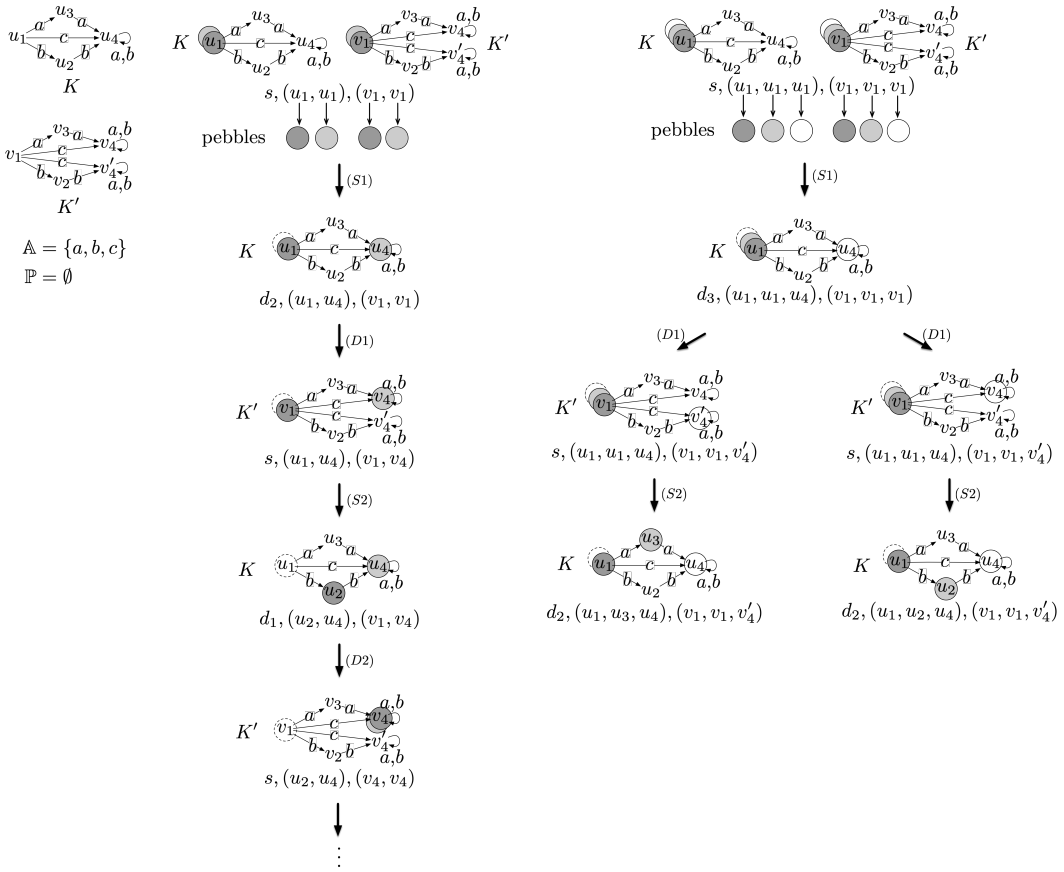

图 5。左：Kripke 结构 K和 K'。中：在 K和 K'上进行的博弈 G[$\rightarrow_2$]示例。移动顺序为：(S1)、(D1)、(S2)、(D2)。事实上，Duplicator 总能在 G[$\rightarrow_2$]中回应 Spoiler 的任意移动，因此 $K, u_1 \rightarrow_2 K'\, v_1$。右：Spoiler 在 K和 K'上进行的博弈 G[$\rightarrow_3$]中的获胜策略的树表示。图中展示了 Duplicator 所有可能的回答。在 (S2) 之后，Duplicator 无法回答，因此 $K, u_1 \not\rightarrow_3 K'\, v_1$。注意，对于 $\pi = \{c(x_s, x_4)\ a(x_s, x_3)\ a(x_3, x_4)\ b(x_s, x_2)\ b(x_2, x_4)\}[x_s, x_s]$，$K, u_1, u_1 \models \pi$以及 $K, v_1, v_1 \not\models \pi$。从图形上看，各位置中的向量可以理解为 Kripke 结构上移动的彩色卵石。图中仅显示相对于上一次移动改变了卵石位置的 Kripke 结构。

$K, \models q$，那么 Duplicator 就赢得了 k 上的存在型 K-棋子博弈，K'。由于 合取程序本质上就是合取查询，因此可以将这一结果调整到我们的研究设定中，这并不令人意外。

根据逆否命题，假设 $K, u, v \not\rightarrow_{k+1} K'\, u', v'$。为简化记号，我们用 (w_1, w_2) 代替 k-元组 $(w_1, w_2, \ldots, w_2)$。$((u, v), (u', v')) \notin \mathrm{Hom}_k(K, K')$ 或者 u和v 属于 K 的不同连通分量的情形都是平凡的，因此我们假设相反的情况。这意味着，Spoiler 在从 $(s, (u, v), (u', v'))$ 出发的 G[$\rightarrow_{k+1}$] 中，拥有一种能在有限轮数内获胜的策略。考虑 Spoiler 的获胜策略，该策略由一棵有限高度的树描述，其顶点标记为来自 G[$\rightarrow_{k+1}$] 的位置。具体而言：(i) 根节点必须标记为 $(s, (u, v), (u', v'))$；(ii) 对于任何标记为 $(d_i, \bar{u}, \bar{v})$ 的顶点，对于 G[$\rightarrow_{k+1}$] 的每一种可能移动，都存在一个标记为 $(s, \bar{u}, \bar{v}')$ 的子节点；(iii) 任何标记为 $(s, \bar{u}, \bar{v})$ 的顶点都恰好有一个子节点 $(s, \bar{u}', \bar{v})$，其标签与 G[$\rightarrow_{k+1}$] 中的一次移动一致；(iv) 对于每个叶节点（标记为 $(d_i, \bar{u}, \bar{v})$），在 G[$\rightarrow_{k+1}$] 中都不存在从该位置出发的可能移动。注意，由于 K' 具有有限度，因此存在有限个

从$G[\rightarrow_{k+1}]$中由 Duplicator 拥有的任意位置出发的移动，因此树的分支数是有限的。因此，策略树是有限的。

考虑 Spoiler 的这样一种获胜策略，由此得到树 T及其标记 $\lambda\colon V(T) \to (W_K)^{k+1}$，其构造过程如下：(1) 删除所有标记为 Spoiler 位置的顶点 y，但保留根节点（即由 Spoiler 所拥有的初始配置），并在 y 的父节点与 y的（唯一）子节点之间添加一条边；(2) 将每个顶点的标记投影到其$(W_K)^{k+1}$ 分量上（也就是说，如果策略树中 x 的标签为$(\bar{d}_i, \bar{u}, \bar{v})$，则为 $\lambda(x) = \bar{u}$）。根据这棵树，可以构造一个树宽为 k 的合取程序 $C[x_s, x_t]$，其中包含这些元组的所有同态类型。令 Q 为所有形如(y, i)的数对所构成的集合，其中 y 是来自 T和 $1 \le i \le k+1$的一个顶点，并令 $\approx$ 为 Q 上的最细等价关系，满足：(a) 若 $\lambda(y)[i] = \lambda(y)[i']$，则$(y, i) \approx (y, i')$；(b) 若$y'$ 是 y 的子节点，且 $\lambda(y)[i] = \lambda(y')[i]$，则$(y, i) \approx (y', i)$。合取程序 C将为 $\approx$的每个等价类使用一个变量，我们将其记为 $[y, i]_\approx$，并包含以下内容：(i) 若 $y = y'$且$(\lambda(y)[i], \lambda(y)[j]) \in [\![a]\!]_K$，则包含一个原子 $a([y, i]_\approx, [y', j]_\approx)$；(ii) 若 $\lambda(y)[i] \in [\![p]\!]_K$，则包含一个原子 $p?([y, i]_\approx, [y, i]_\approx)$。变量 x_s（分别为 x_t）是等价类 $[y, i]_\approx$，其中 y是 T的根节点，且 $\lambda(y)[i] = u$（分别为 $\lambda(y)[i] = v$）。示意图见图 6。

命题 1. $C[x_s, x_t] \in \mathrm{CPDL}^+(\mathrm{TW}_k)$。

证明。根据备注 1，以及 u和 v属于 K的同一个连通分量这一事实，G_C是连通的。另一方面，$G_{C[x_s,x_t]}$的树宽为 k，因为（T，λ）给出了$G_{C[x_s,x_t]}$的一个宽度为 k的树分解（T，λ'），其中对于每个 $y \in V(T)$，$\lambda'(y) = \{[y, i]_\approx : 1 \le i \le k+1\}$。□

命题 2. $K, u, v \models C[x_s, x_t]$和 $K, u', v' \not\models C[x_s, x_t]$。

证明。K、u、$v \models C[x_s, x_t]$这一事实是由（y，i）$\approx$（y'，i'）推出 $\lambda(y)[i] = \lambda(y')[i']$这一结论所导出的，依据定义。事实上，$f = \{[y, i]_\approx \mapsto \lambda(y)[i] : y \in V(T)\ 1 \le i \le k+1\}$是一个满足 C的赋值。

我们通过反证法证明 $K', u', v' \not\models C[x_s, x_t]$。若 $K', u', v' \models C[x_s, x_t]$，考虑相应的 C-满足赋值 $f'\colon \mathrm{vars}(C) \to (W_{K'})$。利用 f'，我们可以如下选取 Spoiler 的策略中的一条分支：对于策略树中标记为 $(\bar{d}_i, \bar{u}, \bar{v})$ 的每个顶点，选取 $(s, \bar{u}, \bar{v}')$，其中 $\bar{v}'[j] = f'([y', j]_\approx)$，而 $y' \in V(T)$ 是策略树中 y 的（唯一）子节点。还需证明，在 $G[\rightarrow_{k+1}]$中存在从 $(\bar{d}_i, \bar{u}, \bar{v})$ 到 $(s, \bar{u}, \bar{v}')$ 的一次移动。根据 C 的定义，它包含由 K 中由 $\bar{u}$ 诱导的子结构，使用变量 $[y', 1]_\approx, \ldots, [y', k+1]_\approx$ 所导出的所有可能原子。因此，$\{\bar{u}[t] \mapsto f'([y', t]_\approx)\}_t = \{\bar{u}[t] \mapsto \bar{v}'[t]\}_t$ 是从 K 到 K' 的部分同态见证，因此 $(\bar{d}_i, \bar{u}, \bar{v}) \to (s, \bar{u}, \bar{v}')$ 是 $G[\rightarrow_{k+1}]$的合法移动。这意味着 Spoiler 的策略树不可能是有限的，因为沿此方式向下遍历树，最终会到达策略树的一个叶节点 $(\bar{d}_i, \bar{u}, \bar{v})$，而该节点在 $G[\rightarrow_{k+1}]$中仍有一条出边移动，这与该策略对 Spoiler 来说是必胜策略的事实矛盾（更准确地说，与第 (iv) 点矛盾）。□

请注意，严格来说，C 可能包含无限多个原子（这些原子取自有限的变量集(C)，即变量的集合）。这是因为，在有限度的Kripke 结构中，两个世界之间可能存在无限多个转移。然而，根据 $K, u', v' \not\models C[x_s, x_t]$以及 K具有有限度这一事实，可以提取一个由有限多个原子 $C' \subseteq C$构成的集合，使得 $K, u', v' \not\models C'[x_s, x_t]$具体如下所述。

考虑映射集合 $F\colon \mathrm{vars}(C) \to (W_{K'})$，它们满足 (i) $f(x_s, x_t) = (u', v')$，以及 (ii) $f(x)$ 到 u'或 v' 的距离至多为 $|\mathrm{vars}(C)|$。注意，由于 K' 的度有限，F 是有限的。对于每个 $f \in F$，选择 C 中的一个原子 $a_f(x_f, y_f)$，使其不会映射到 K'，也就是说，对于 K' 的转移关系 $\rightarrow_{a_f}$，满足 $(f(x_f)\ f(y_f)) \notin \rightarrow_{a_f}$。令 C' 为所有此类原子的集合，再加上来自 C 的一些（有限个）原子，以确保 (a) $\mathrm{vars}(C') = \mathrm{vars}(C)$，以及 (b) $G_{C'}$是连通的。注意，此时 C' 是有限的。我们声称 $K', u', v' \not\models C'[x_s, x_t]$。事实上，如果存在一个满足 C' 的

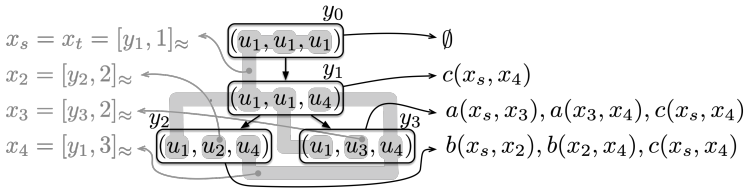

图 6. 树的一个示例 T，其节点为 y_0、y_1、y_2、y_3，出现在定理 2 1-to-2 的证明中，（下方）中 Spoiler 的策略构造。涉及的变量以及 C 中的原子 满足 K、u_1、$u_1 \models \pi$ 和K, v_1，$v_1 \not\models \pi$对于 $\pi = C[x_s, x_t]$。灰色区域表示由 $\approx$ 诱导的划分。

赋值 f：变量集$(C') \to$ (K,)，那么 $f \in F$ 和 C' 应包含原子 (, y_f)，它不会被 f 映射到 K,。因此，不存在满足 C' 的赋值，并且 K, u'，$v' \not\models C'[x_s, x_t]$。

的情形 K、$u \not\rightharpoonup_{k+1} K, u'$是类似的。在这种情况下，$C[x_s, x_t]$的构造与前面相同，但我们令$x_t$等于 x_s。最终的公式于是为$\langle C[x_s, x_t]\rangle$。

2 到 1 我们现在通过对表达式进行结构归纳法证明 2 到 1 的蕴含。对于任意 $(\bar{u}, \bar{v}) \in \{(v, v), ((u, v), (u', v'))\}$, 假设 $K, \bar{v} \rightharpoonup_{k+1} K, \bar{v}$,并且 $K, \bar{v} \models e$对于一个正的 CPDL$^+$(GTW$_k$) 表达式 e成立，我们证明 $K, \bar{v}' \models e$。

唯一有趣的情况是 e 是一个合取程序；其余情况都很简单，可参见附录C.1。

▷ $e = C[x_s, x_t]$实际上，我们将使用更一般的程序版本 $C[\bar{z}]$，其中 $\bar{z}$可以包含来自 vars(C) 的任意数量的变量 $1 \le t \le k+1$，而 $C[\bar{z}]$ 的底层图 $\mathsf{G}_{C[\bar{z}]}$ 是 G_C 加上每一对来自 $\bar{z}$ 的不同变量 y、y' 之间的边 $\{y, y'\}$。因此，$\mathsf{G}_{C[\bar{z}]} \in \mathsf{TW}_k$ 表示所有变量 $\bar{z}$ 都出现在其树分解的某个袋中。Kripke 结构 K 上的语义 $[\![C[\bar{z}]]\!]_K$（˜ ˜）符合预期：由来自 K 的世界组成的 t 元组 $\bar{u}$ 的集合，其中 $f(\bar{z}) = \bar{u}$ 对某个满足 C 的赋值 f 成立。

ˆ 假设我们有 $e = C[x_s, x_t]$。我们证明如下：

命题 3. 对于每个 $C \subseteq \hat{C}$以及来自 变量集(C) 的每个变量元组 $x_1, \ldots, x_n$，满足 $n \le k+1$，使得 $\mathsf{G}_{\hat{C}[x_1,\ldots,x_n]} \in \mathsf{TW}_k$，对于所有 $\bar{v} \in$ (K)^{n}和 $\bar{v}' \in$ (K,)^n：如果 $(K, \bar{v}) \rightharpoonup_{k+1} (K, \bar{v}')$且 $\bar{v} \in [\![C[x_1, \ldots, x_n]]\!]_K$，那么 $\bar{v}' \in [\![C[x_1, \ldots, x_n]]\!]_{K'}$。

ˆ 首先注意到，上述命题蕴含 $K' u', v' \models C[x_s, x_t]$。现在我们对 C 的大小 $|C|$ 使用归纳法证明该命题。设 (T, v) 是 G_π 的一个宽度为 k 的树分解，其中 $\pi = C[x_1, \ldots, x_n]$。不失一般性地，设 T 的根节点袋 b 满足 $\mathrm{v}(b) = \{x_1, \ldots, x_n\}$。

假设 $(K, \bar{v}) \rightharpoonup_{k+1} (K, \bar{v})$ 且 $\bar{v} \in [\![\pi]\!]_K$。我们将证明 $\bar{v}' \in [\![\pi]\!]_{K'}$。设 h是 K上的一个 C-满足赋值，满足 $\bar{v} = (h(x_1) \ldots, h(x_n))$。我们需要给出 K,上的一个 C-满足赋值 h'，使得 $\bar{v}' = (h'(x_1) \ldots, h'(x_n))$。

假设 T 的根节点 b 有 ℓ 个子节点 $b_1, \ldots, b_\ell$，并令 $T_1, \ldots, T_\ell$ 为对应的子树。令 $\bar{y}_i$ 为来自 $\mathrm{v}(b_i) \cap \mathrm{v}(b)$ 的（两两不同的）变量构成的任意向量，也就是说，这些变量同时出现在 T 的根节点包中，以及其第 i 个子节点的包中。我们可以假设 $\bar{y}_i$ 的维度为 $\le k$（否则，父节点和子节点的包将完全相同）。令 $\bar{v}_0 = \bar{v}$，并对每个 $1 \le i \le \ell$ 令 $\bar{v}_i = h(\bar{y}_i)$。令 C_0 为 C 在根节点包中的所有原子的集合（即两个变量都在 $\mathrm{v}(b)$ 中的原子）。令 $C_i \subseteq (C \setminus C_0)$ 为包含在 T_i 的各个包中的 $C \setminus C_0$ 的所有原子的集合（即两个变量都在 T_i 的某个包中的原子）。

于是可得 $K, \bar{v} \models C[x_1, \ldots, x_n]$当且仅当 $K, \bar{v}_i \models C_i[\bar{y}_i]$对每个 $0 \le i \le \ell$ 都成立。现在有两种可能：要么是 $C_0 = C$，要么是 $C_0 \subsetneq C$。

情形 $C_0 = C$：若 $C_0 = C$，这意味着 C 的所有原子都使用 $\{x_1, \dots, x_n\}$ 中的变量。考虑函数 $h' \colon \mathrm{vars}(C) \to K'$，使得若 $h(x_i) = u[j]$，则 $h'(x_i) = u'[j]$。下面证明 h' 是 K' 上一个 C-满足赋值。对于每个原子 $\pi'(x_i, x_j) \in C = C_0$，根据关于 h 的假设，有 $(h(x_i)\ h(x_j)) = (u[i], u[j])$ 属于 $[\![\pi']\!]_K$。另一方面，我们还有 $K, u[i], u[j] \to_{k+1} K', u'[i], u'[j]$。因此，根据关于 π' 的归纳假设，$(u'[i], u'[j]) \in [\![\pi']\!]_{K'}$。这意味着 h' 是 K' 上一个 C-满足赋值。

情况 $C_0 \subsetneq C$：若 $C_0 \subsetneq C$，则对每个 i 都有 $|C_i| < |C|$。[6] 于是，由于我们有 K、$u_i \models C_i[\bar{y}_i]$ 以及 $K, u_i \to_{k+1} K', u'_i$，可以对每个 C_i 应用归纳假设，从而得到某个满足 C 的赋值 h'_i，使得在 K' 上有 $K', u'_i \models C_i[\bar{y}_i]$，并将 $\bar{y}_i$ 映射到 u'_i。这意味着 $h' := h'_0 \cup \cdots \cup h'_\ell$ 是 K' 上满足 C 的赋值，因此 $K', h'(x_1) \dots, h'(x_n) \models C[x_1, \dots, x_n]$。□

5.2 双模拟关系

这里我们像之前一样，通过双人游戏 (K, K')来定义 $\mathbf{G}[\rightleftharpoons_k]$上的双模拟概念，这与定义模拟游戏 $\mathbf{G}[\to_k]$时的做法相同。不失一般性，假设 K, K' 的世界集合互不相交。这次，$\mathbf{G}[\rightleftharpoons_k]$ 的博弈场包含一组位置 $S \cup D$，其中

$$S = \{s\} \times (\mathit{Hom}_k(K, K') \dot\cup \mathit{Hom}_k(K', K)),$$

$$D = \{d_1, \dots, d_k\} \times \big((W(K)^k \times W(K')^k) \dot\cup (W(K')^k \times W(K)^k)\big),$$

Spoiler 拥有 S中的所有位置，而 Duplicator 拥有 D中的所有位置。$\mathbf{G}[\rightleftharpoons_k]$的移动集合是满足以下条件的最小集合：

(1) 若 $u' = u[i \mapsto w]$，则存在一项移动，从 (s, u, v) 到 (d_i, u', v)；其中 w 是与 $u[j]$ 的距离为 ≤ 1 的一个世界，对某个满足 $i \neq j$ 的 $1 \leq j \leq k$ 而言；(2) 若 $u[1] = \cdots = u[k]$（因此 $v[1] = \cdots = v[k]$），则存在一项移动，从 (s, u, v) 到 (s, v, u)。(3) 若 $v' = v[i \mapsto w]$，则存在一项移动，从 (d_i, u', v) 到 (s, u', v')；其中 w 是与 $v[j]$ 的距离为≤ 1的一个世界，对某个满足 $i \neq j$ 的 $1 \leq j \leq k$ 而言。

同样，Duplicator 的获胜条件就是任意无限对局。

我们现在定义两个相关的概念。k-半双模拟的概念 $\to\!\!\!\!\!\leftrightarrow_k$ 定义如下。给定两个 Kripke 结构 K, K'，其世界集合互不相交（不失一般性），以及给定元组 $u \in W(K)^k$和 $u' \in W(K')^k$，若满足以下条件，我们称存在一个从 K, u到 K', u'的 k-半双模拟，记作 $K, u \to\!\!\!\!\!\leftrightarrow_k K', u'$：1) (s, u, u') 是 $\mathbf{G}[\rightleftharpoons_k]$在 (K, K') 上的有效位置（即它们诱导出部分同态）；2) u 中的所有世界都处于K的同一个连通分量中；3) Duplicator 从 (s, u, u') 出发拥有获胜策略。若 $K, u \to\!\!\!\!\!\leftrightarrow_k K', u'$且 $K', u' \to\!\!\!\!\!\leftrightarrow_k K, u$，我们称 K, u与 K', u'之间存在一个 k-双模拟，并记作 $K, u \rightleftharpoons_k K', u'$。

Remark 2. *If* $K_1, u_1 \to\!\!\!\!\!\leftrightarrow_k K_2, u_2$ *then* $K_1, u_1 \to_k K_2, u_2$.

与定理 2 类似，对于 k-半双模拟也可以得到如下刻画结果：也就是说，(1) $K, u \to\!\!\!\!\!\leftrightarrow_{k+1} K', u'$当且仅当(2) 对于任意 $\mathrm{CPDL}^+(\mathrm{TW}_k)$ 表达式 e（相应地为公式或程序），都有 $K, u \models e$蕴含 $K', u' \models e$。证明遵循相同的总体思路。在从 1 到 2 的情形中，Spoiler 的策略可能更复杂，因为他可以改变结构。不过，每次切换结构时使用否定，仍可将其转化为 $\mathrm{CPDL}^+(\mathrm{TW}_{k-1})$ 的公式。对于从 2 到 1 的情形，证明与定理 2 完全相同，按表达式的结构进行归纳。由于备注 2，基础情形与定理 2 中的情形相同。若 $\varphi = \neg\psi$，则根据条目 2，有 $K', v' \to\!\!\!\!\!\leftrightarrow_{k+1} K, v$，因此由归纳假设可得 $K', v' \models \psi$蕴含 $K, v \models e$。后者等同于 $K, v \models \varphi$蕴含 $K', v' \models \varphi$。形式化表述及证明细节见附录 C.2。

[6] 存在一种退化情形，使得 $C_0 = \emptyset$，但由于 图 $\mathbf{G}_C$ 是 C 连通的，我们也可以假设 树分解 满足至少存在一个 原子 位于 C_0 中。

注意，由于 公式 对否定封闭，根据前述刻画（引理 5），我们可得 K、$v \leftrightarrows_{k+1} K', v'$ 当且仅当对于每个 CPDL$^+$(TW$_k$) 公式 φ，都有 K、$v \models \varphi$ 当且仅当 $K', v' \models \varphi$。要得到关于 程序 的类似结果，我们只需取该概念的对称闭包，即 k-双模拟：

定理 3。 设 $k \geq 2$。给定有限度的 *Kripke*结构 K, K'，以及世界 $u, v \in V(K)$ 和 $u', v' \in V(K')$，以下各项等价

(1) 对每个 CPDL$^+$(TW$_k$) 公式 φ，都有 $K, v \models \varphi$ 当且仅当 $K', v' \models \varphi$；以及 *(2)* $K, v \rightleftharpoons_{k+1} K', v'$；

并且以下各项等价

(1) 对每个 CPDL$^+$($\mathcal{G}$TW$_k$) 程序 π，都有 $K, u, v \models \pi$ 当且仅当 $K', u', v' \models \pi$；以及 *(2)* $K, u, v \rightleftharpoons_{k+1} K', u', v'$。

此外，有限度这一假设仅在 1-到-2 蕴含中需要。

5.3 计算（双）模拟

注意，当 K、K' 是有限结构时，博弈G$[\rightarrow_k]$ 和G$[\rightleftharpoons_k]$ 也是有限的，共有 $O(|V(K)|^k \cdot |V(K')|^k)$ 个位置。两种情况下，Duplicator 的获胜条件都是“安全条件”。因此，可以在线性时间内，根据博弈的移动数和位置数计算获胜位置的集合，例如通过经典的 McNaughton-Zielonka吸引子分解计算。这产生了一个多项式算法，该算法运行时间为$(|V(K)| \cdot |V(K')|)^{O(k)}$，用于计算 k-模拟关系和 k-双模拟关系 $\rightarrow_k$以及 $\rightleftharpoons_k$。

命题 5. 对于每个 k，关系 $\rightarrow_k$和 $\rightleftharpoons_k$在有限个 结构 K上，K'都可以在多项式时间内计算。

也可以（而且很有用）将结构 K, K' 之间针对 CPDL$^+$(TW$_{k-1}$) 的模拟 $\rightarrow_k$ 概念，约化为模态逻辑（ML）在 k-幂 Kripke 结构上的经典模拟关系，即 $S_k(K)$ 和 $S_k(K')$，其中 $S_k(K)$ 的世界是来自 K 的世界所组成的 k 元组，而原子程序则反映了改变某个分量的不同方式（通过距离为 ≤ 1 的见证，在 K 中），如 G$[\rightarrow_k]$所示。对于 Kripke 结构之间的双模拟 $\rightleftharpoons_k$，以及两个方向上的（单向）模拟，也可以采用相同的方法，其中每个结构都包含 K 和 K' 的信息。这将为证明这些（双）模拟关系可以在多项式时间内计算提供另一种证明。关于这种与 ML 模拟之间联系的完整细节，请参见附录 C.3 节。

6 分离

在本节中我们将展示，尽管 CPDL$^+$(TW$_1$) 和 CPDL$^+$(TW$_2$) 是 表达能力相同的，我们仍有一个严格的层次结构 CPDL$^+$(TW$_2$) $\subsetneq$ CPDL$^+$(TW$_3$) $\subsetneq \cdots$ 来自 TW$_2$。

Theorem 4. *For every* $k \geq 2$, CPDL$^+$(TW$_k$) $\subsetneq$ CPDL$^+$(TW$_{k+1}$).

<escape>本质上，我们证明了 $(k+1)$-团的存在性可以在 CPDL$^+$(TW$_k$) 中表达，但无法在 CPDL$^+$(TW$_{k-1}$)中表达，对任意 $k \geq 3$均成立。考虑以下 $(k+1)$-团公式</escape>

$$\xi_{k+1} := \langle C[x_1, x_{k+1}] \rangle \wedge \neg \langle C'[x_1, y] \rangle$$

CPDL$^+$(TW$_k$) 的表达能力，其中 $C = \{a(x_i, x_j) \mid 1 \leq i < j \leq k+1\}$ 且 $C' = \{a(x_1, y)\ a(y, y)\}$，对任意固定的 $a \in \mathbb{A}$。我们将证明，对于每个 $k \geq 3$，ξ_{k+1} 都不能用 CPDL$^+$(TW$_{k-1}$) 表达。这一不可表达性结果将直接源自以下具有独立研究价值的“树状模型性质”。

命题 6. 对于每个 $k \geq 2$、*Kripke* 结构 K 和世界 $u \in W_K$，都存在一个树宽为 $\leq k-1$ 的 *Kripke* 结构 $\hat{K}$ 以及世界 $\hat{u} \in W_{\hat{K}}$，使得 $K, u \rightleftharpoons_k \hat{K}, \hat{u}$。此外，如果 K 是 $\hat{}$ 可数的，则 K 有一个宽度为 $\leq k-1$ 的可数树分解。

证明。 考虑集合 $\mathcal{S}$ 所有由至多 k 个世界组成的非空集合，取自 K。

考虑具有 $V(T) = \{u\} \cdot \mathsf{S}^*$ 的无限树 T（即，顶点是以 $\{u\}$ 开头的 S 上的有限序列），并且当 u 是 v 的长度为 $|v|-1$ 的前缀时，存在一条边 $\{u, v\}$。将 T 的根顶点 r 视为 $r = \{u\}$。我们将 $v \in V(T)$ 的最后一个集合记为 $\lambda(v)$（即，对于每个 $S \in \mathsf{S}$ 和 $v \in \mathsf{S}^*$，$\lambda(v \cdot S) = S$）。

对于一对世界 u，u'（属于 K）以及一对顶点 v，v'（属于树 T），记 $\approx$ 为 $\{((u, v), (u', v')) \mid v$，是 v 在 $T, u = u'$ 和 $u \in \lambda(v) \cap \lambda(v')$ 中的子节点$\}$的自反、对称和传递闭包。令 $[u, v]_\approx$ 表示相对于 $\approx$ 的 (u, v) 的等价类。

现在我们定义 $\hat{K}$ 具有

(1) $W_{\hat{K}} = \{[u, v]_\approx \mid u \in W_K, v \in V(T), u \in \lambda(v)\}$; (2) 对于每个原子程序 $a \in \mathbb{A}$，都有 $[\![a]\!]_{\hat{K}}$ 由所有满足 $([u, v]_\approx, [u', v]_\approx) \in [\![a]\!]_K$ 的世界对 $([u, v]_\approx, [u', v]_\approx)$ 组成；以及 (3) 对于每个原子命题 $p \in \mathbb{P}$，都有 $[\![p]\!]_{\hat{K}}$ 由所有满足 $u \in [\![p]\!]_K$ 的世界 $[u, v]_\approx$ 组成。

最后，考虑标记 $\mathrm{v} : V(T) \to \wp(W_{\hat{K}})$，它将每个顶点 v 映射到 $\{[u, v]_\approx \mid u \in \lambda(v)\}$。$\hat{}$ 由此可知，(T, v) 是 K 的一个树分解，其树宽至多为 $k-1$。由此可知，(T, v) 是一个树分解。

断言 4. (T, v) 是一个 树分解，它属于 $\hat{K}$，其 树宽 至多为 $k-1$。

$\hat{}$我们证明，通过构造，K 是 k-双模拟的于 K。

$\hat{}$命题 5. $K, u \rightleftharpoons_k K, \hat{u}$, 其中 $\hat{u} = [u, r]_\approx$ 和 r 是 T 的根节点。

证明。我们将证明一个更一般的结论：对于任意 k 元组 $\bar{u}$，其中的世界来自 K，以及 T 的任意顶点 $\bar{}$ $\hat{}$ 具有以下形式 $v = w \cdot \{\bar{u}[1], \ldots, \bar{u}[k]\}$，我们有 $K, \bar{u} \rightleftharpoons_k K, \hat{u}$，其中 $\hat{u}[i] = [\bar{u}[i], v]_\approx$ 对每一个 i 均成立。

$\bar{}$ $\bar{}$ $\bar{}$注意，$\{\bar{u}[i] \mapsto \hat{u}[i]\}_{i \leq k}$ 是一个函数，而且还是一个双射：如果 $\hat{u}[i] = \hat{u}[j]$，那么 $(\bar{u}[i], v) \approx (\bar{u}[j], v)$，这意味着 $\bar{u}[i] = \bar{u}[j]$。根据 $[\![\to_a]\!]_{\hat{K}}$ 的定义，实际上，$\{\bar{u}[i] \mapsto \bar{}$ $\hat{u}[i]\}_i$ 和 $\bar{}$ $\hat{}$ $\hat{}$ $\{\hat{u}[i] \mapsto \bar{u}[i]\}_i$ 分别是从 $\mathrm{Hom}_k(K, K)$ 和 $\mathrm{Hom}_k(K, K)$ 出发的部分同态。通过这种方式，我们验证了 $K, \bar{u} \leftrightarrows_k K, \hat{u}$ 和 $K, \hat{u} \leftrightarrows_k K, \bar{u}$ 的第一个条件。

$\hat{}$ $\bar{}$令 u_i'in K与某个 $\bar{u}[j]$相距 距离 ≤ 1。于是我们可以证明 $K, \bar{u}[i \mapsto u_i'] \rightleftharpoons_k K, \hat{u}[i \mapsto \hat{u}_i']$对于 $\hat{}$ $u_i' = [u_i', v_\prime]_\approx$ 和 $v_\prime = v \cdot (\{\bar{u}[i'] \mid i' \neq i\} \cup \{u_i'\}) \in V(T)$。注意，对于每个 $i' \neq i$，我们都有 $\bar{}$ $\bar{}$ $\hat{}$ $\hat{u}[i'] = [\bar{u}[i'], v]_\approx = [\bar{u}[i'], v']_\approx$，因此 $\hat{u}[i \mapsto \hat{u}_i']$ 具有所需的形式。

$\hat{}$ $\bar{}$ $\hat{}$ 对于 K 中与某个 $\hat{u}[j]$ 距离为 ≤ 1 的任意 $\hat{u}_i'$，我们以对称方式进行：根据 $K^{\bar{}}$ 的定义，必定存在某个 v 和 u_i'，使得 $\hat{u}_i' = [u_i', v]_\approx$，$\hat{u}[j] = [\bar{u}[j], v]_\approx$ 以及 u_i'、$\bar{u}[j]$ 在 K 中的距离为 $\hat{}$ $\bar{}$ ≤ 1。然后，我们继续使用 K、$\bar{u}[i \mapsto u_i']$ 和 K、$\hat{u}[i \mapsto \hat{u}_i']$，它们具有所需的形式。

$\hat{}$ $\bar{}$然后，我们可以永远重复同样的策略，从而证明 $K, \bar{u} \rightleftharpoons_k K, \hat{u}$。 □

最后，注意到 (T, v) 和 $\hat{K}$ 在 K 可数时也是可数的。 □

与ICPDL的情况相同，CPDL$^+$ 可以用最小不动点逻辑定义，因此它也继承了勒文海姆–斯科伦可数模型性质：如果一个CPDL$^+$ 公式是可满足的，那么它在一个可数的结构中也是可满足的。因此，我们可以由命题6 和定理3得到以下推论。

推论5。 对于每个 $k \geq 2$，CPDL$^+$(TW$_k$) 具有“TW$_k$-模型性质”：若公式 $\varphi \in$CPDL$^+$(TW$_k$) 可满足，则它在一个树宽至多为的可数 *Kripke* 结构中 k中可满足。

现在我们可以开始证明定理4。

定理4的证明。令 $k \geq 2$。注意，$(k+1)$-团公式 $\xi_{k+1} \in \mathrm{CPDL}^{+}(\mathrm{TW}_k)$ 蕴含存在一个以当前世界为起点的有向 $(k+1)$-团，其中包含 a个元素；并且它显然在一个有向 $(k+1)$-团结构中可满足。然而，树宽为 $\leq k-1$ 的结构中不可能存在任何 $(k+1)$-团。因此，根据推论5，ξ_k 无法用 $\mathrm{CPDL}^{+}(\mathrm{TW}_{k-1})$ 表达。□

注*3.* 尽管命题 *6*适用于每个 $k \geq 2$，但这并不意味着 $CPDL^{+}(TW_1)$具有 TW_1-模型性质，因为定理 *3*中 $\rightleftharpoons_k$ 与 $CPDL^{+}(TW_{k-1})$之间的刻画仅对 $k \geq 3$成立。事实上，根据定理*1*，$CPDL^{+}(TW_1) \equiv CPDL^{+}(TW_2)$，因此，*3*-团公式实际上可在 $CPDL^{+}(TW_1)$中表达。

7 可满足性

对于 CPDL 的片段 C，其可满足性问题是：给定一个公式$^{+}$ C，判断是否存在某个 Kripke 结构 $\varphi \in$ 及其中的一个 世界 K，使得 w K成立， $w \models \varphi$。

注意，任何可数树分解都可以转换为底层树为二叉树的树分解，同时保持树宽不变。根据前述推论5、$\mathrm{CPDL}^{+}(\mathrm{TW}_k)$的可满足性问题（更一般地，$\mathrm{CPDL}^{+}$ ）的可判定性可直接得出：CPDL^{+} 表达式可以有效地表示为单子二阶逻辑（MSO）表达式，并且已知对于树宽有界的结构，MSO 的可满足性可通过将其 MSO 解释映射到树结构上（参见例如 [23,事实 2]）以及 Rabin 定理来判定。由于每个 CPDL^{+} 公式都属于某个 k对应的 $\mathrm{CPDL}^{+}(\mathrm{TW}_k)$，因此 CPDL^{+} 也是可判定的。

命题 7。 CPDL^{+}-可满足性 是可判定的。

复杂性。 接下来，我们将确定可满足性问题的复杂性。我们紧密沿用 [12,定理48] 中关于 ICPDL 的可满足性证明，并将其调整到更一般的 CPDL^{+}情形。为明确材料的复用情况，所有定义和记号要么逐字采用，要么紧密遵循 [12]中的表述。我们将明确指出证明中有所不同的部分。

证明通过一系列归约完成：

- 从任意 结构 上的 $\mathrm{CPDL}^{+}(\mathrm{TW}_k)$ 可满足性问题 到树宽为 结构 k （由推论 5可直接得到）。
- 从 树宽为 结构 $^{+}(\mathrm{TW}_k)$ 上的 CPDL k 可满足性问题 到“正则树”上的 $\mathrm{CPDL}^{+}(\mathrm{TW}_k)$ 可满足性问题。
- 从正则树上的 $\mathrm{CPDL}^{+}(\mathrm{TW}_k)$ 可满足性问题 到 树上的自动机模型的空性问题。

7.1 ω-正则树可满足性

该 ω-正则树可满足性是一个判定问题：给定一个CPDL^{+} 公式 φ 和一个双向交替奇偶树自动机 (TWAPTA) $\mathcal{T}$，判断是否存在一棵无限树属于 $L(\mathcal{T})$，当将其视为一个Kripke 结构时，它是 φ的模型。

令 Σ_N 为有限节点字母表， Σ_E 为有限边字母表。 Σ_N-标记的 Σ_E-树是一个偏函数 T: ${\Sigma_E}^{*} \to \Sigma_N$，其定义域记为 $dom(T)$，且关于前缀封闭。 $dom(T)$ 中的元素是 T 的节点。若 $dom(T) = {\Sigma_E}^{*}$，则称 T 完备。本节其余部分中，我们讨论完备树。我们用 $\mathrm{tree}(\Sigma_N, \Sigma_E)$ 表示所有完备 Σ_N-标记的 Σ_E-树组成的集合。若 Σ_E 不重要，我们就简称为 Σ_N-标记树。

由 TWAPTA 接受的完整 $\wp(P)$-标记 A-树，其中 $P \subseteq \mathbb{P}$ 和 $A \subseteq \mathbb{A}$ 分别是由 原子命题和原子程序构成的有限集。这样的树 T 可以被视为

使用 Kripke 结构（A^*，$\{\rightarrow_a \mid a \in A\}$，$\{T_p \mid p \in P\}$），其中对于所有 $a \in A$ 和 $T_p =$，$\rightarrow_a = \{(u, ua) \mid u \in A^*\}$，且对于 $p \in P$，$\{u \in A^* \mid p \in T(u)\}$。请注意，以这种方式导出的 Kripke结构 相对于 A 是确定且全定义的，即对于所有 $a \in A$，转移关系 $\rightarrow_a$ 是一个全函数。

TWAPTA 是行走自动机，在每次转移时，可以从 Σ_N 读取当前节点标签，并移动到子节点，检查来自 Σ_E的边标签；也可以通过检查来自 Σ_E的边标签移动到父节点，或停留在同一节点。此外，自动机还支持交替，因此转移是这些类型移动的 正的布尔组合 。接受基于 奇偶条件。由于篇幅限制，且我们的论证无需处理其定义的相关细节，我们将其定义推迟到附录 E.1。

对于 TWAPTA T 及其状态 s，我们将 $[\![\mathrm{T}\ s]\!]$ 定义为所有满足以下条件的有序对 (T, u) 的集合：$T \in \mathrm{tree}(\Sigma_N, \Sigma_E)$，且 $u \in {\Sigma_E}^*$ 存在 T 在 T 上、从状态 s 和顶点 u 开始的成功运行（详情见附录 E.1）。T 接受的语言 $L(\mathrm{T})$ 定义为 $\{T \in \mathrm{tree}(\Sigma_N, \Sigma_E) \mid (T, \varepsilon) \in [\![\mathrm{T}\ s_0]\!]\}$，其中 s_0 是 T 的初始状态。我们使用以下关于 TWAPTA 空性问题的上界，其中指标 $i(\mathrm{T})$ 指的是奇偶条件中使用的指标数量。

定理 6。 *([24], [12,* 定理 *3。1])* 对于给定的 *TWAPTA* $\mathcal{T}$ 及其转移函数 δ，可以在 $exp(|\mathcal{T}| + i(\mathcal{T})) \cdot |\delta|^{O(1)}$ 的时间内检验 $L(\mathcal{T}) \neq \emptyset$是否成立。其中 $exp(n)$ 表示 $2^{n^{O(1)}}$。

现在我们可以正式定义 ω-正则树可满足性。设 φ 是一个 CPDL$^+$ 公式，设 $A = \{a \in \mathbb{A} \mid a$在 φ 中出现$\}$，并且 $P = \{p \in \mathbb{P} \mid p$在 φ 中出现$\}$。如果存在 $T \in L(\mathrm{T})$ 使得 $\varepsilon \in [\![\varphi]\!]_T$，则公式 φ 关于在 $\wp(P)$-标记的 A-树上的 TWAPTA T 是可满足的。最后，ω-正则树可满足性是这样一个判定问题：给定这样的 φ 和 T，判断 φ 是否关于 T 可满足。

7.2 求解 ω-正则树可满足性

我们证明 ω-正则树可满足性问题属于2指数时间，方法是将其指数时间归约为TWAPTA的非空性问题。实际上，我们将使用ICPDL$^+$ 而不是CPDL$^+$，以简化对以下构造的重用 [12]用于ICPDL。

该归约的主要组成部分是对以下两类对象进行相互归纳翻译：(i) 将ICPDL$^+$ 公式翻译为TWAPTA，以及 (ii) 将ICPDL$^+$ 程序翻译为某类非确定性自动机（非确定有限自动机），该类自动机在 [12]中被引入。后者是作用于单词的标准非确定有限自动机，它们沿着一个完全的 Σ_N-标记树移动，并从$\{a, \bar{a} \mid a \in \Sigma_E\}$中读取符号。它们还可以进行条件 ε-转移，只有当当前树节点被给定的TWAPTA接受时，这些转移才可执行。关于非确定有限自动机在TWAPTA上的定义见附录E.2。

对于基于 TWAPTA T 的 NFA A，我们将 $[\![\mathrm{A}]\!]$ 定义为所有三元组 (T, u, v) 的集合，使得$T \in \mathrm{tree}(\Sigma_N, \Sigma_E)$、$u$、$v \in {\Sigma_E}^*$，并且 A 在 T 上使用来自 T 的 $+T$个测试，从 u到 v存在一条接受运行（详见附录 E.2）。

从 ICPDL$^+$ 到自动机。固定一个有限集，其中包含 原子命题 $P \subseteq \mathbb{P}$ 和 原子程序$A \subseteq \mathbb{A}$，基于它们构建 ICPDL$^+$ 公式 和 程序。在树这一语境下，对于 ICPDL$^+$公式 ψ和 程序 π，我们定义 $[\![\psi]\!]$ 和 $[\![\pi]\!]$ 如下：

$$[\![\psi]\!] := \{(T, u) \mid T \in tree(\wp(P), A), u \in A^*, u \in [\![\psi]\!]_T\},$$
$$[\![\pi]\!] := \{(T, u, v) \mid T \in tree(\wp(P), A), u, v \in A^*, (u, v) \in [\![\pi]\!]_T\}.$$

本节的目的是展示如何转换

- 每个 公式 ψ转换为一个 TWAPTA $\mathcal{T}(\psi)$ 使得 $[\![\mathcal{T}(\psi)]\!] = [\![\psi]\!]$ 并且
- 每个 程序 π转换为一个 TWAPTA$\mathcal{T}(\pi)$和一个 非确定有限自动机 $\mathcal{A}(\pi)$其字母表为 $\mathcal{T}(\pi)$，使得 $[\![\mathcal{A}(\pi)]\!] = [\![\pi]\!]$。

所有自动机都在 $\wp(P)$-标记的 A-树上运行。该构造对 ψ 和 π 的结构进行归纳。每个公式 ψ 对应的 TWAPTA $\mathrm{T}(\psi)$ 的定义与 [12, §3.2] 中相同。我们只需描述 ICPDL$^+$ 程序 π 对应的 $\mathrm{A}(\pi)$ 和 $\mathrm{T}(\pi)$ 的归纳构造。如果 π 的形式为 $a, \bar{a}, \psi?$, $\pi_1 \cup \pi_2$, $\pi_1 \cap \pi_2$, $\pi_1 \circ \pi_2$ 或 χ^*，则按 [12, §3.2] 中的方法进行构造。

7.2.1 合取程序的情形。剩下的是形如 $\pi = C[x_s, x_t]$ 的合取程序情形。根据归纳假设，对于每个原子 $\pi'(x, x') \in C$，都存在一个 TWAPTA $\mathrm{T}(\pi')$ 和一个定义在 $\mathrm{T}(\pi')$ 上的非确定有限自动机 $\mathrm{A}(\pi')$，使得 $[\![\mathrm{A}(\pi')]\!] = [\![\pi']\!]$。不失一般性，我们假设各原子对应自动机的状态集两两不相交。因此，我们将它们分别记作 $\mathrm{T}(\pi'(x, x'))$ 和 $\mathrm{A}(\pi'(x, x'))$。令 Q 为对每个 $\pi'(x, x') \in C$，状态集 $\mathrm{A}(\pi'(x, x'))$ 的（不相交）并集。

对于任意 $\pi_\iota(x, x_\iota) \in C$ 以及状态 q, q' 属于 $\mathcal{A}(\pi_\iota(x, x_\iota))$，我们将考虑一种新型的 程序，其形式为

$$p \dashrightarrow q$$

其语义为 $[\![p \dashrightarrow q]\!] := [\![\mathrm{A}_{q,q'}(\pi'(x, x'))]\!]$，其中 $\mathrm{A}_{q,q'}(\pi'(x, x'))$ 是在 TWAPTA $\mathrm{T}(\pi'(x, x'))$ 上的非确定有限自动机 $\mathrm{A}(\pi'(x, x'))$ 将 q 和 q' 分别设为初始状态和终止状态后所得的结果。请注意，严格来说，$p \dashrightarrow q$ 也取决于 $\mathrm{A}(\pi'(x, x'))$ 和 $\mathrm{T}(\pi'(x, x'))$；但由于我们已假设状态互不相交，因此可以唯一确定这些自动机，故它们在定义中是隐含的。Q 上的 ICPDL 程序是任何由以下语法描述的程序

$$\pi ::= \pi \star \pi \quad \text{for } \star \in \{\cap, \cup, \circ\}$$
$$\pi ::= p \dashrightarrow q \quad \text{for } p, q \in Q$$

具有预期的语义。

接下来，我们将展示如何把 $C[x_s, x_t]$ 转换为一个在 Q 上的等价 ICPDL 程序。此后，只需应用前面几种情形（$\cap$、$\circ$、$\cup$ [12, §3.2] 中的转换，将其转换为一个合适的非确定有限自动机即可。

考虑任意一个以 树 T_C 为结构、以变量集(C)为顶点的二叉树。对于任意两个变量 x，$y \in$ 变量集(C)，我们记 $x \rightsquigarrow_{T_C} y$ 为从 x 到 y 在 T_C 中的唯一简单路径。令 $f\colon$ 变量集$(C) \to \wp(Q)$ 为一个函数，它将每个变量 y 映射到状态 $Q_y \subseteq Q$ 的一个集合，使得对于每个原子 $\pi_\iota(x, x') \in C$：

- 如果路径 $x \rightsquigarrow_{T_C} x_\iota$ 包含 y，那么 Q_y 恰好包含 $\mathrm{A}(\pi_\iota(x, x_\iota))$ 中的一个状态，
- 否则，Q_y 不包含来自 $\mathrm{A}(\pi_\iota(x, x_\iota))$ 的任何状态。

其直观理解是，借助 T_C 我们猜测 C 映射到树中的“形状”，并借助 f 我们猜测见证每个原子的路径中的中间状态。令 $\mathcal{S}$ 为满足上述条件的所有对(T_C, f)所组成的集合。

对于 C 的变量集中的任意两个变量 y、$y' \in$，若 y' 是 T_C 中 y 的子节点，且 $q \in f(y)$ 和 $q' \in f(y')$ 是某个公共非确定有限自动机 $\mathrm{A}(\pi'(x, x'))$ 中的两个状态，则路径 $x \rightsquigarrow_{T_C} x'$ 要么 (a) 先经过 y 再经过 y'，要么 (b) 先经过 y' 再经过 y。于是，我们将原子 $\rho(y, y')$ 与 $\rho = q \dashrightarrow q'$（若 (a) 成立），或将原子 $\rho(y', y)$ 与 $\rho = q' \dashrightarrow q$（若 (b) 成立），关联到这样的元组 (y, y', q, q')。

对于给定的 T_C 和 f，考虑 $C_{T_C,\, f}$ 作为所有以这种方式描述的原子所构成的集合。注意到

(1) $\mathrm{vars}(C_{T_C,f}) = \mathrm{vars}(C)$，(2) $C_{T_C,f}$ 的原子数量至多是 C 的原子数量的多项式级（事实上是二次），并且 (3) $\bigcup_{(T_C,f) \in \mathcal{S}} [\![C_{T_C,f}[x_s, x_t]]\!] = [\![C[x_s, x_t]]\!]$，(4) $C_{T_C,f}$ 的底层图的树宽为 1。

由于对于每个 $\hat{C} = C_{T_C,f}$ 满足 $(T_C, f) \in \mathcal{S}$ 的情形，我们有 $G_{\hat{C}}$ 的树宽为1，因此$G_{\hat{C}[x_s,x_t]} \in \mathrm{TW}_2$。因此，我们可以在多项式时间内构造一个在 Q上等价的 ICPDL 程序。

将其称为 $\hat{}\pi_{T_C,f}$，其大小为多项式级（参见定理 1）。因此，$C[x_s,x_t]$ 等价于 Q 上由指数多个多项式大小的 ICPDL 程序构成的并，每个 $(T_C,f)\in S$ 对应一个 $\hat{}\pi_{T_C,f}$。随后，我们可以应用 §3.2] 中用于 ICPDL [12, 的构造，得到所需的 TWAPTA $T(C[x_s,x_t])$ 和一个定义在 $T(C[x_s,x_t])$ 上的非确定有限自动机 $A(C[x_s,x_t])$，使得 $[\![A(C[x_s,x_t])]\!]=[\![C[x_s,x_t]]\!]$。

7.2.2 自动机大小。 ICPDL 程序 交宽度 $\mathrm{iw}(\pi)$ 定义为 ICPDL 程序 π 中的 [12] 如下：

$$\mathrm{iw}(a)=\mathrm{iw}(\bar{a}):=1 \text{ for all } a\in\mathbb{A},\qquad \mathrm{iw}(\theta?):=1,$$
$$\mathrm{iw}(\pi_1\cup\pi_2)=\mathrm{iw}(\pi_1\circ\pi_2):=\max\{\mathrm{iw}(\pi_1),\mathrm{iw}(\pi_2)\},$$
$$\mathrm{iw}(\pi^*):=\mathrm{iw}(\pi),\qquad \mathrm{iw}(\pi_1\cap\pi_2):=\mathrm{iw}(\pi_1)+\mathrm{iw}(\pi_2).$$

这一概念可推广到任意 程序 或 公式 α 定义 $\mathrm{IW}(\alpha)$ 为其中某个 程序 的最大交宽度 （如果不包含 程序，则为 1）。IW 之所以重要，是因为如 [12], 所示，对于每个 ICPDL 表达式 α，所构造的 TWAPTA 仅在 $\mathrm{IW}(\alpha)$上呈指数增长：

引理 3。 *([12,引理 3。7])* 对于每个 ICPDL 表达式 e，$|\mathcal{T}(e)|\le 9\cdot|e|^{2\cdot\mathrm{IW}(e)+1}$ 以及 $i(\mathcal{T}(e))\le|e|$。

注 4。 通过简单检查 *[12],* 的构造，可以将前述界改进为$i(\mathcal{T}(e))\le$ 否定深度(e)，其中 否定深度(e) 表示 表达式 e 中嵌套否定的最大层数（加一）。

在针对 ICPDL$^+$ 表达式的构造中，我们得到的交宽度是多项式有界的，因为在每一步中，我们都会将一个合取程序转换为已有自动机上的一组多项式规模的 ICPDL 程序。具体而言，每个 $\hat{}\pi_{T_C,f}$above 的交宽度至多为 $C_{T_C,f}$ 的原子数量，因此关于 C 的原子数量至多为二次。此外，该构造不会改变嵌套否定的数量。因此，根据引理 3，对于每个 ICPDL$^+$ 表达式 α，我们得到一个规模为单指数的 TWAPTA $T(\alpha)$。

让我们将其记为 由合取宽度$\mathrm{cw}(\pi)$ 的CPDL$^+$ 程序 π 定义如下：
Denote

$$\mathrm{cw}(a)=\mathrm{cw}(\bar{a}):=1 \text{ for all } a\in\mathbb{A},\qquad \mathrm{cw}(\theta?):=1,$$
$$\mathrm{cw}(\pi_1\cup\pi_2)=\mathrm{cw}(\pi_1\circ\pi_2):=\max\{\mathrm{cw}(\pi_1),\mathrm{cw}(\pi_2)\},$$
$$\mathrm{cw}(\pi^*):=\mathrm{cw}(\pi),\qquad \mathrm{cw}(C[x_s,x_t]):=\sum_{\pi(x,y)\in C}\mathrm{cw}(\pi).$$

我们将 $\mathrm{CW}(\alpha)$ 定义为任意 表达式 α 中最大的 合取宽度，即其中某个 程序 的合取宽度（如果不包含任何程序，则为 1）。

在前一个将合取程序归约为定义在$\mathcal{Q}$上的ICPDL 程序的并的过程中，注意到每个定义在上的ICPDL 程序的规模都是多项式级的，因此关于原始公式的合取宽度也是多项式级的。此外，交宽度是合取宽度的多项式。将这一观察与引理3的规模界、备注4以及定理6中 TWAPTA 的复杂性结合起来，我们得到以下结果（详情见附录 E.3）。

命题 8。 对于一个 *TWAPTA* $\mathcal{T}$，其转移函数为 δ，以及一个 CPDL$^+$ 公式 φ，我们可以在以下时间内判定

$$exp\big(|\mathcal{T}|+i(\mathcal{T})+|\varphi|^{\mathrm{CW}(\varphi)}\big)\cdot|\delta|^{O(1)}$$

是否存在某个$T\in L(\mathcal{T})$使得$\varepsilon\in[\![\varphi]\!]_T$。因此，$\omega$-正则树可满足性问题对于CPDL$^+$ 属于2指数时间，并且 ω-正则树可满足性问题对于$\{\varphi\in\mathrm{CPDL}^+\mid\mathrm{CW}(\varphi)\le n\}$属于指数时间，对于任意 $n\in\mathbb{N}$。

7.3 化归为 ω-正则树可满足性

这本质上就是第 [12, §4.2]节所展示的归约。与 [12, 引理4。7]中完全相同的归约（只需进行简单调整，将树宽2 改为树宽 k）表明，对于每个CPDL$^+$(TW$_k$)公式 φ，都存在一个可在多项式时间内计算的CPDL$^+$(TW$_k$) 公式 φ' 以及TWAPTA $\mathcal{T}$，使得下述结论成立。

引理4。 φ 在一个树宽为 k 的结构上可满足，当且仅当 φ' 相对于T可满足。此外，CW(φ) = CW(φ')。

<escape>只有在将 树宽 视为固定值时，该归约才是多项式的；否则，该归约将是指数级的。</escape>

作为引理 4、命题 8和推论 5 的结果，我们得到本节的主要结果：

定理7. 对于每个 $k \geq 1$,
(1) CPDL$^+$-可满足性 可判定，复杂度为 3指数时间，*(2)* CPDL$^+$(TW$_k$)-可满足性是 2指数时间完全的，*(3)* $\{\varphi \in \text{CPDL}^+ \mid \text{CW}(\varphi) \leq k\}$-可满足性 是 指数时间完全的。

定理 7 中第2项的下界源于已知的2指数时间 下界ICPDL 可满足性 [20, 定理 2] 结合定理1。第3项的下界源于指数时间 下界命题动态逻辑 [10]。

根据下面的备注，定理 7 的最后一项可视为结果 [12, 定理4。8]，该结果指出，ICPDL 公式的交宽度有界时，其可满足性问题属于指数时间。

备注5。 存在一个多项式时间翻译，可将ICPDL 公式（具有交宽度 k）翻译为CPDL$^+$(TW$_2$) 公式，其合取宽度 k 有限（例如，翻译Tr$_3$ 出自§*B.3*）。

8 模型检测

我们考虑（有限）模型检测问题：给定一个有限 Kripke 结构K、一个世界 $w \in |K|$ 和一个 CPDL$^+$ 公式 φ，判断是否 $K, w \models \varphi$。已知对于 PDL、ICPDL 及许多其他变体，该问题是 PTime-完全的 [10, 19]。在不受限制的情形下，容易看出，CPDL$^+$ 的模型检测问题对于复杂度类 DP $= \{A \setminus B \colon A, B \in \text{NP}\}$ 是困难的。

命题 9。 *CPDL$^+$* 的模型检测问题是 *DP*-困难的，且属于 P^{NP}。

然而，如果 CPDL$^+$ 公式的树宽有界，模型检测仍然属于多项式时间。

定理 8。 对于任意连通图类 G:
(1) 若 $G \subseteq TW_k$ 对于某个 k成立，则 *CPDL$^+$(G)* 的模型检测问题是 *PTime*-完全的；*(2)* 否则，在假设 G 可递归枚举且 $W[1] \neq FPT$ 的条件下，*CPDL$^+$(G)* 的模型检测问题不属于 *PTime*。即使对于 *CPDL$^+$(G)* 的正片段，这一结论仍然成立。

证明概要。 1 为了证明 CPDL$^+$(TW$_k$) 模型检测 属于 多项式时间，所采用的过程是一种经典的动态规划算法。给定一个 公式 φ 和一个有限的 Kripke 结构 k，我们基于 世界 中 K 的 子表达式 ψ 的 φ 标记，迭代地对 子表达式的 ψ中的所有世界进行标记。

为了处理 程序 $\pi \in sub(\varphi)$ 我们利用了以下事实： 合取查询 的树宽 $\leq k$ 的求值可在多项式时间内完成 [7, 定理 3] (更准确地说LOGCFL-完全

[13, 定理 6。12]）。这得出了一种算法，其运行时间关于 φ 呈线性时间，关于 K 呈多项式时间，其中多项式的次数为 $k+1$，如果 $\varphi \in \mathrm{CPDL}^+(\mathrm{TW}_k)$。

下界来源于多项式时间困难性，即模型检测模态逻辑的（参见，例如，[19,命题 5]）。

2. 这是因为，关于合取查询已有类似结论：对于底层图属于 G 的二元关系上的布尔合取查询类，其模型检测问题不属于多项式时间，除非 FPT = W[1] [15,推论 19]。因此，由布尔合取查询的求值到 $\mathrm{CPDL}^+(\mathrm{G})$ 的模型检测存在多项式时间归约，可得该结果：当且仅当公式 $C[x,x]$ 在结构 K 的某个世界中成立时，布尔合取查询 q 在（有限）结构 K 中为真；其中 C 是 q 的原子集合，x 是 q 的任意变量。□

9 结论

我们引入了一种表达能力强的逻辑 CPDL^+，它涵盖了在Kripke 结构 和 图数据库中已知的若干形式体系。我们尚不清楚 CPDL^+ 的可满足性复杂性是否为 3指数时间完全，这个问题留待未来研究。尽管 可满足性 对CPDL^+ 而言是可判定的，但有限可满足性甚至对于 loop-CPDL [12, §7]仍是一个值得注意的开放问题。另请参阅附录 G中的讨论。

致谢

本研究部分获得了法国-阿根廷国际研究项目（IRP）SINFIN的资助。Diego Figueira 部分获得了 ANR QUID 项目（资助编号：ANR-18-CE400031）的资助。Santiago Figueira 部分获得了布宜诺斯艾利斯大学（UBA）项目（资助编号：UBACyT 20020190100021BA）的资助。

参考文献

[1] Albert Atserias、Phokion G. Kolaitis 和 Moshe Y. Vardi。2004。作为证明系统的约束传播。载于《约束编程原理与实践 - CP 2004》，第10届国际会议 CP 2004，2004年9月27日至10月1日，加拿大多伦多，论文集（《计算机科学讲义》，第3258卷）。Springer，77–91。https://doi.org/10.1007/978- 3-540-30201-8 9_[2] Pablo Barceló Baeza。2013。图数据库查询。载于 ACM 数据库系统原理研讨会（PODS）。ACM，
175 188 10 1145 2463664 2465216 – 。https://doi.org/ 。 / 。Michael Benedikt、Pierre [3] Bourhis 和 Michael Vanden Boom。2016。可判定不动点逻辑表达力的提升。载于计算机科学逻辑年会（LICS）。ACM，817–826。https://doi.org/10.1145/2933575. 2933592[4] Hans L Bodlaender。1993。一种寻找小树宽树分解的线性时间算法。载于第25届 ACM 计算理论年度研讨会论文集。226–234。[5] Pierre Bourhis、Markus Krötzsch 和 Sebastian Rudolph。2014。如何最佳嵌套正则路径查询。载于第 27 届描述逻辑国际研讨会非正式论文集，奥地利维也纳，2014年7月17日至20日（CEUR 研讨会论文集，第1193卷）。CEUR-WS.org，404–415。http://ceur-ws.org/Vol-1193/paper_80.pdf[6] Diego Calvanese、Giuseppe De Giacomo、Maurizio Lenzerini 和 Moshe Y. Vardi。2000。带逆运算的合取正则路径查询包含关系。载于知识表示与推理原理（KR）。Morgan Kaufmann，176–185。[7] Chandra Chekuri 和 Anand Rajaraman。2000。重新审视合取查询包含关系。《理论计算机科学》 239, 2 2000 211 229 10 1016 0304 3975 99 00220 0 ( ), – 。https://doi.org/ 。 /S - ( ) - E. [8] Allen Emerson 和 Charanjit S. Jutla。 。树自动机、μ演算与确定性（扩展摘要）。载于第32届计算机科学基础年会，波多黎各圣胡安，1991年10月1日至4日。IEEE 计算机学会，
368 377 10 1109 1991 185392 – 。https://doi.org/ 。 /SFCS. 。Diego Figueira、 [9] Santiago Figueira 和 Edwin Pin。 。强化版 PDL：关于带交与逆运算的 PDL 表达力扩展。载于计算机科学逻辑年会（LICS）。ACM 出版社。[10] Michael J. Fischer 和 Richard E. Ladner。1979。正则程序的命题动态逻辑。《计算机与系统科学杂志》（JCSS）18，2（1979），194–211。https://doi.org/10.1016/0022-0000(79)90046-1

[11] Stefan Göller 和 Markus Lohrey。2006。命题动态逻辑的无限状态模型检测。载于 EACSL 计算机科学逻辑年会（CSL）（《计算机科学讲义》，第4207卷）。Springer，349–364。https://doi.org/ 10 1007 11874683_23 . / Stefan Göller、Markus Lohrey 和 Carsten Lutz。2009。带交与逆运算的 PDL：可满足性与无限状态模型检测。《符号逻辑杂志》 74, 1 2009 279 314 10 2178 1231082313 ( )，–。https://doi.org/ 。[12] /jsl/ Georg Gottlob 、 Nicola Leone 和 Francesco Scarcello。2001。。无环合取查询的复杂性。《 ACM 学报》 48，3 2001 431 498 10 1145 382780 ( )，–。https://doi.org/ 。 / 。[13] [14] Erich Grädel、Wolfgang Thomas 和 Thomas Wilke。2003。自动机、逻辑与无限博弈：当前研究指南。第2500卷。Springer。[15] Martin Grohe、Thomas Schwentick 和 Luc Segoufin。2001。何时合取查询求值是易处理的？载于第33届ACM计算理论年度研讨会论文集，2001年7月6日至8日，希腊克里特岛伊拉克利翁。ACM， 657 666 10 1145 380752 380867 – 。https://doi.org/ 。 / 。[16] David Harel、Dexter Kozen 和 Jerzy Tiuryn。2001。动态逻辑。载于《哲学逻辑手册》。Springer，99–217。[17] Phokion G. Kolaitis 和 Moshe Y. Vardi。1995。论 Datalog 的表达能力：工具与案例研究。《计算机与系统科学杂志》（JCSS） 51, 1 1995 110 134 10 1006 1995 1055 ( )，–。https://doi.org/ 。 /jcss. 。[18] Phokion G. Kolaitis 和 Moshe Y. Vardi。2000。。合取查询包含与约束满足。《计算机与系统科学杂志》（JCSS） 61, 2 2000 302 332 10 1006 2000 1713 ( )，–。https://doi.org/ 。 /jcss. 。[19] Martin Lange。2006。。使用所有扩展特性的命题动态逻辑模型检测。《应用逻辑杂志》4，1（2006），39–49。https://doi.org/ 10 1016 2005 08 002 . /j.jal. . . Martin Lange 和 Carsten Lutz。。带交运算的命题动态逻辑的指数时间下界。《符号逻辑杂志》 70, 4 2005 1072 1086 10 2178 1129642115 ( )，–。https://doi.org/ 。 /jsl/ 2005 2[20] Juan L. Reutter、Miguel Romero 和 Moshe Y. Vardi。2017。。图数据库上的正则查询。《理论计算机科学》 61, 1 2017 31 83 10 1007 00224 016 9676 2 ( )，–。https://doi.org/ 。 /s - - - Sebastian Rudolph 和 Markus Krötzsch。2013。。标记与检查：使用单子定义查询进行数据访问。载于 ACM 数据库系统原理研讨会（PODS）。ACM， 151 162 10 1145 2463664 2465227 – 。https://doi.org/ 。 / 。[21] [22] Detlef Seese。1991。。可判定图单子理论的模型结构。《纯粹与应用逻辑年刊》（1991），169–195。 https://doi.org/10.1016/0168-0072(91)90054-P [24] Moshe Vardi。1989。关于将双向自动机归约为单向自动机的注记。《信息处理快报》（IPL）30，5（1989），261–264。 53, 2 [23]

A 部分缺失内容3

A.1 子表达式的定义

对于 e 一个ICPDL$^+$-程序或一个ICPDL$^+$-公式，令sub(e) 为所有子表达式的集合 e，即满足以下条件的最小集合：

- $e \in$下标(e),
- 若 $\psi \wedge \psi' \in \mathrm{sub}(e)$ 则$\{\psi,\ \psi'\} \subseteq \mathrm{sub}(e)$
- 如果 $\neg\psi \in \mathrm{sub}(e)$ 那么 $\psi \in \mathrm{sub}(e)$
- 如果$\langle\pi\rangle \in \mathrm{sub}(e)$，那么 $\pi \in \mathrm{sub}(e)$
- 如果 $\pi\star\ \pi' \in \mathrm{sub}(e)$，那么$\{\pi, \pi'\} \subseteq \mathrm{sub}(e)$对每个$\star \in \{\circ\ \cup \cap\}$,
- 如果 $\pi^* \in \mathrm{sub}(e)$，那么 $\pi \in \mathrm{sub}(e)$,
- 如果 $\psi? \in \mathrm{sub}(e)$，那么 $\psi \in \mathrm{sub}(e)$,
- 如果 $C[x_s, x_t] \in \mathrm{sub}(e)$并且 $\pi(x,y) \in C$，那么 $\pi \in \mathrm{sub}(e)$。

B 缺失的章节详情4

B.1 从 loop-CPDL到CPDL$^+$($\mathcal{G}_{\circlearrowleft}$)

我们定义如下翻译 Tr_1，将 loop-CPDL-公式 映射到 CPDL$^+$($\mathcal{G}_{\circlearrowleft}$)-公式，并将 loop-CPDL-程序 映射到CPDL$^+$($\mathcal{G}_{\circlearrowleft}$)-程序：

$$
\begin{aligned}
\mathrm{Tr}_1(\star) &= \star && \star \in \mathbb{P} \cup \{\varepsilon\}\\
\mathrm{Tr}_1(\varphi \wedge \psi) &= \mathrm{Tr}_1(\varphi) \wedge \mathrm{Tr}_1(\psi)\\
\mathrm{Tr}_1(\neg\varphi) &= \neg\mathrm{Tr}_1(\varphi)\\
\mathrm{Tr}_1(\langle\pi\rangle) &= \langle\mathrm{Tr}_1(\pi)\rangle\\
\mathrm{Tr}_1(\varphi?) &= \mathrm{Tr}_1(\varphi)?\\
\mathrm{Tr}_1(\pi_1 \star \pi_2) &= \mathrm{Tr}_1(\pi_1) \star \mathrm{Tr}_1(\pi_2) && \star \in \{\cup \circ\}\\
\mathrm{Tr}_1(\pi^*) &= \mathrm{Tr}_1(\pi)^*\\
\mathrm{Tr}_1(\mathrm{loop}(\pi)) &= \{\mathrm{Tr}_1(\pi)(x,x)\}[x,x] && (2)
\end{aligned}
$$

命题 10. loop-CPDL $\leqq$ CPDL$^+$($\mathcal{G}_{\circlearrowleft}$) 通过$\mathrm{Tr}_1$。

B.2 从 CPDL$^+$($\mathcal{G}_{\circlearrowleft}$) 到loop-CPDL

在 CPDL$^+$($\mathcal{G}_{\circlearrowleft}$)中，唯一允许的 合取程序 $C[x,y]$满足 $x = y$，且所有 原子 在C的形式均为 $\pi(x,x)$，其中 π一个 CPDL$^+$($\mathcal{G}_{\circlearrowleft}$)-程序。

考虑以下从 CPDL$_2$ (G$_{\circlearrowleft}$)-公式到 loop-CPDL-公式，以及从 CPDL$^+$(G$_{\circlearrowleft}$)-程序到 loop-CPDL-程序的翻译 Tr$^+$：采用 §B.1 中 Tr_1 的所有子句，但不包括 (2)，并将 Tr_1 替换为 Tr_2，然后添加

$$
\mathrm{Tr}_2(C[x,x]) = \left(\bigwedge_{\pi(x,x)\in C} \mathrm{loop}(\mathrm{Tr}_2(\pi))\right)?
$$

命题 11。 CPDL$^+$($\mathcal{G}_{\circlearrowleft}$) $\leqq$ loop-CPDL 通过Tr_2。

B.3 从 ICPDL到CPDL$^+$($\mathcal{G}_{\cap}$的翻译

考虑如下从 ICPDL 公式到 CPDL$^+$(G$_{\cap}$) 公式、以及从 ICPDL 程序到 CPDL$^+$(G$_{\cap}$) 程序的翻译 Tr_3：取 §B.1 中 Tr_1 的所有子句，将 Tr_1 替换为 Tr_3，但不包括 (2)，并添加

$$
\mathrm{Tr}_3(\pi_1 \cap \pi_2) = \{\mathrm{Tr}_3(\pi_1)(x,y), \mathrm{Tr}_3(\pi_2)(x,y)\}[x,y].
$$

注意，任何 底层图 都属于 $C[x,y] \in \mathrm{sub}(\mathrm{Tr}_3(\pi))$ 中的 $\mathcal{G}_\cap$，因此 $\mathrm{Tr}_3(\pi)$ 属于 $\mathrm{CPDL}^+(\mathcal{G}_\cap)$。

<escape>命题3. ICPDL $\leqq$ $\mathrm{CPDL}^+(\mathcal{G}_\cap)$。</escape>

引理 *1.* 给定一个 *CPDL⁺(TWₙ)*-合取程序 $C'[x,y]$，可以在多项式时间内计算出一个 *CPDL⁺(TWₙ)*-合取程序 $C[x,y] \equiv C'[x,y]$，其变量集满足 $vars(C) = vars(C')$，以及一个树宽为 $\leq n$ 的 $G^{C[x,y]}$ 的树分解 (T, v)，使得对于任意 $b \in V(T)$，$v(b)$ 都是 $G^{C[x,y]}$ 中的一个团。

证明。 设 $(T, \mathbf{v})$ 是一个 树宽 至多为 n 的 $\mathrm{G}^{C'[x_1,x_2]}$ 的树分解。设 $A = \mathbb{A} \cap \mathrm{sub}(C'[x,y])$ 为 原子程序 在 C' 中或其 子表达式的集合。定义 C 为 $C' \cup D$，其中

$$D = \{((\cup_{a\in A} a) \cup (\cup_{a\in A} \bar{a}))^*(z_1, z_2) \mid b \in V(T), z_1, z_2 \in \mathbf{v}(b), z_1 \neq z_2\}.$$

注意到（T，v）是 G_D 的一个树分解，其中每个包都是 G_D 中的一个团。由于 $\mathrm{G}_{C'}$ 是连通的，并且 CPDL^+ 的语义具有组合性，因此对于表达式中未出现的原子无关，于是 $C[x,y] \equiv C'[x,y]$，显然，$\mathrm{vars}(D) \subseteq \mathrm{vars}(C')$。此外，（$T$，v）也是 $\mathrm{G}^{C[x,y]}$ 的一个树分解。由于 D 至多有 $|\text{个 vars}(C')|^2$ 原子，而计算树宽至多为 n 的树分解可以在线性时间内完成 [4], ，因此整个构造仍可在多项式时间内完成。□

对于一个 ICPDL-程序 π，我们定义 π^{-1} 如下：

$$\begin{aligned}
\star^{-1} &= \star & \bigstar &\in \{\varepsilon, \varphi?\} \\
a^{-1} &= \bar{a} & a &\in \mathbb{A} \\
\bar{a}^{-1} &= a & a &\in \mathbb{A} \\
(\pi_1 \star \pi_2)^{-1} &= \pi_2^{-1} \star \pi_1^{-1} & \star &\in \{\cap, \cup, \circ\} \\
\pi^{*-1} &= \pi^{-1*}
\end{aligned}$$

命题 12。 π 是一个 ICPDL-程序 当且仅当 π^{-1} 是一个 ICPDL-程序，并且 $(u, v) \in [\![\pi]\!]_K$ 当且仅当 $(v, u) \in [\![\pi^{-1}]\!]_K$。

引理 2. 设 C 是一个由以下形式的原子构成的有限集：$\pi(z_1, z_2)$，其中 π 是一个 ICPDL-程序，且 z_1，$z_2 \in \mathbb{V}$，G_C 是一个团，并且 $|\text{变量集}(C)| \leq 3$。则：

(1) 对于任意 x，$y \in$ 变量集(C)，$x \neq y$，存在一个 ICPDL-程序 $\pi_{C[x,y]}$ 使得 $\pi_{C[x,y]} \equiv C[x,y]$。

(2) 对于任意 $x \in$ 变量集(C)，存在一个 *ICPDL*-程序 $\pi_{C[x,x]}$ 使得 $\pi_{C[x,x]} \equiv C[x,x]$。

此外，这些翻译均可在多项式时间内完成。

证明。 对于 z_1，$z_2 \in$ 变量集(C)，$z_1 \neq z_2$，令

$$\Pi_{z_1 z_2} = \left(\bigcap_{\pi(z_1,z_2)\in C} \pi \right) \cap \left(\bigcap_{\pi(z_2,z_1)\in C} \pi^{-1} \right)$$

$$\Pi_{z_1} = \left(\bigwedge_{\pi(z_1,z_1)\in C} \langle \pi \cap \varepsilon \rangle \right)?$$

其中，空范围的合取定义为 ε。

对于 $x \neq y$，定义如下的ICPDL-程序$\pi_{C[x,y]}$和$\pi_{C[x,x]}$：

- 若变量集$(C) = \{x, y, z\}$与$x \neq z \neq y$，则

$$\pi_{C[x,y]} = \Pi_x \circ (\Pi_{xy} \cap (\Pi_{xz} \circ \Pi_z \circ \Pi_{zy})) \circ \Pi_y$$
$$\pi_{C[x,x]} = \langle \pi_{C[x,y]} \rangle ?$$

- 如果变量集$(C) = \{x, y\}$那么

$$\pi_{C[x,y]} = \Pi_x \circ \Pi_{xy} \circ \Pi_y$$
$$\pi_{C[x,x]} = \langle \pi_{C[x,y]} \rangle ?$$

- 如果 变量集$(C) = \{x\}$那么

$$\pi_{C[x,x]} = \Pi_x$$

可以证明，$\pi_{C[x,y]} \equiv C[x,y]$和 $\pi_{C[x,x]} \equiv C[x,x]$。 □

命题 4. ICPDL$^+$(TW$_2$) $\leqq$ ICPDL 通过多项式时间翻译。

证明。令 $c_1(e) = \bigcup_{C[x,y] \in \mathrm{sub}(e)} \mathrm{vars}(C)$， $c_2(e)$ 表示 sub(e) 中不是 ICPDL 表达式的表达式集合，并令 $c_3(e)$ 表示 e 的语法复杂度。最后，令 $c(e) = (|c_1(e)|, |c_2(e)|, c_3(e))$。我们按照 c 的字典序递归定义从 ICPDL$^+$(TW$_2$) 公式到 ICPDL 公式，以及从 ICPDL$^+$(TW$_2$) 程序到 ICPDL 程序的翻译 Tr(e)。

对于合取程序以外的情况，Tr 的定义与 §B.1 中的 Tr$_1$ 相同，只需将 Tr$_1$ 替换为 Tr，并将 $\bigstar \in \{\cup, \circ\}$ 替换为 $\bigstar \in \{\cup, \circ, \cap\}$。$c$ 的第二个分量确保：如果根据这些规则之一有 Tr$(e) = \ldots \mathrm{Tr}(e') \ldots$，那么 $c(e') < c(e)$。

对于 合取程序 $C[x,y]$的情况，设

$$C = \{\pi_1[x_1, y_1], \ldots, \pi_n[x_n, y_n]\}.$$

使得 $\mathsf{G}_{C[x,y]} \in \mathrm{TW}_2$。如果所有 π_i 都是 ICPDL 程序，且 $|\mathrm{vars}(C)| \leq 3$，那么 C 满足引理 2 的假设。在 $x = y$的情况下，存在一个 ICPDL 程序 $\pi_{C[x,x]} \equiv C[x,x]$，并定义 Tr$(C[x,y]) = \pi_{C[x,x]}$。在 $x \neq y$的情况下，存在一个 ICPDL 程序 $\pi_{C[x,y]} \equiv C[x,y]$，并定义 Tr$(C[x,y]) = \pi_{C[x,y]}$。

否则存在某个 π_i不是一个 ICPDL-程序 或 $|$变量集$(C)| > 3$。我们定义 Tr$(C[x,y])$ =Tr$(D[x,y])$对于一组 原子 D，具体如下。

(1) 如果存在 i，使得 π_i不是 ICPDL 程序，那么 $D = C \setminus \{\pi_i[x_i, y_i]\} \cup \{\mathrm{Tr}(\pi_i)\ [x_i, y_i]\}$。显然，Tr$(C[x,y])$ $\equiv$Tr$(D[x,y])$。注意，在这种情况下，(1) $c_1(D[x,y]) \subseteq c_1(C[x,y])$，因为根据构造，我们不会向 D添加新变量；并且 (2) $c_2(D[x,y]) \subsetneq c_2(C[x,y])$，因为在 D中，我们移除了至少一个不是 ICPDL 程序的表达式，且没有添加新的表达式。因此， $c(D[x,y]) < cC[x,y]$。

(2) 否则，所有 π_i都已经是 ICPDL 程序，并且 $|\mathrm{vars}(C)| > 3$。设 (T, v) 是 C 的底层图的一个以包含 x 和 y 的根节点为根的树分解（注意，由于 G$_{C[x,y]}$包含边 $\{x, y\}$，这是可能的），其宽度为 ≤ 2，且 T 的任意叶节点都是大小至少为 2 （至多为 3 ）的袋，并且不包含在其父袋中。若 T 是单个袋 B，则结论成立：在 $x = y$ 的情况下，定义 Tr$(C[x,y]) = \pi_{C[x,x]}$；否则定义 Tr$(C[x,y])$ $= \pi_{C[x,y]}$。若 T 至少有两个袋，则从 T 中任选一个叶节点 B，并令 B' 为 T 中 B 的父节点。令 C_B 为所有满足 vars$(A) \subseteq B$ 的原子 $A \in C$ 的集合。注意， C_B满足引理 2 的假设。

- 假设 $B' \cap B = \{z_1\}$ 且 $B \supseteq \{z_1, z_2\}$ $(z_1 \neq z_2)$。我们将 D 定义为 C 中除 C_B 中元素之外的所有原子的集合，再加上引理 2 中的 ICPDL 程序 $\pi_{C_B[z_1,z_1]} \equiv C_B[z_1, z_1]$。注意， $z_2 \in \mathrm{vars}(C) \setminus \mathrm{vars}(D)$。

- 假设 $B' \cap B = \{z_1, z_2\}$ $(z_1 \neq z_2)$ 且 $B = \{z_1, z_2, z_3\}$ $(z_1 \neq z_3 \neq z_2)$。我们将 D 定义为 C 中除 C_B 中元素之外的所有原子的集合，再加上引理 2 中的 ICPDL 程序 $\pi_{C_B[z_1,z_2]} \equiv C_B[z_1, z_2]$。注意，$z_3 \in \mathrm{vars}(C) \setminus \mathrm{vars}(D)$。显然，$\mathrm{Tr}(C[x,y]) \equiv \mathrm{Tr}(D[x,y])$。此外，注意 $c_1(D[x,y]) \subsetneq c_1(C[x,y])$，且 $c_2(D[x,y]) \subseteq c_2(C[x,y])$。因此 $c(D[x,y]) < c(C[x,y])$。

至此，证明完毕。我们只需证明该翻译可在多项式时间内完成，这一点将在下文的命题13中证明。□

命题 13。从ICPDL$^+$(TW$_2$) 到ICPDL 的翻译可在多项式时间内计算。

证明。我们首先分析以下算法 1。该算法源自命题 4 证明中的情形 2，它将形如 $C[x,y]$ 的 CPDL$^+$(TW$_2$) 合取程序翻译为语义等价的 ICPDL 程序，其中 C 是形如 $\pi(z_1, z_2)$ 的原子集，π 是 ICPDL 程序，且 z_1、$z_2 \in \mathbb{V}$。

输入：一个 CPDL$^+$(TW$_2$) 程序 $C'[x,y]$，其中 C' 是形如 $\pi(z_1, z_2)$ 的原子集，π 是 ICPDL 程序，且 z_1、$z_2 \in \mathbb{V}$输出：一个 ICPDL 程序 π，使得 $\pi \equiv C'[x,y]$令 $C[x,y] \equiv C'[x,y]$ 满足 $\mathrm{vars}(C) = \mathrm{vars}(C')$，并令 (T, v) 为 $\mathsf{G}^{C[x,y]}$ 的一个树宽至多为 2 的树分解，使得任意 $b \in V(T)$、$\mathrm{v}(b)$ 都是 $\mathsf{G}^{C[x,y]}$ 中的团（引理 1 ）。当 T 不是单个袋时，令 B 为 T 的一个叶节点；令 B' 为 B 在 T 中的父节点；$C_B := \{\pi(z_1, z_2) \in C \mid \{z_1, z_2\} \subseteq B\}$；如果 $B' \cap B = \{z_1\}$ 且 $B \setminus B' \neq \emptyset$，则$C := C \setminus C_B \cup \{\pi_{C_B[z_1,z_1]}\}$；结束如果 $B' \cap B = \{z_1, z_2\}$ $(z_1 \neq z_2)$ 且 $B = \{z_1, z_2, z_3\}$ $(z_1 \neq z_3 \neq z_2)$，则$C := C \setminus C_B \cup \{\pi_{C_B[z_1,z_2]}\}$；从 T中移除 B 结束如果 $x \neq y$，则返回 $\pi_{C[y,y]}$，否则返回 $\pi_{C[x,x}$结束算法 1：特殊情况下计算从 ICPDL$^+$(TW$_2$) 到 ICPDL 的翻译

根据引理 1，第一行可以在多项式时间内完成，并且 $|C| = |C'| + O(|\text{变量集}(C')^2|)$。

对于 π一个 ICPDL-程序，令 $c(\pi)$ 表示 π的语法复杂度；对于由形如 $\pi(z,\ z')$ 的 原子 组成的集合 D，其中 π是一个 ICPDL-程序，令 $c(D) = \sum_{\pi(z,\ z') \in D} c(\pi)$。

构造ICPDL-程序$\pi_{C_B[z_1,z_1]}$和$\pi_{C_B[z_1,z_2]}$自引理 2 得出，可以在 $O(c(C_B))$ 时间内完成。注意，上述算法从 C中的原子集合中 C_B移除了元素，但新增了一个ICPDL-程序。我们检查得出，该程序不会

在循环的迭代过程中增长过多。通过检查引理 2 中 $\pi_{C_B[z_1,z_1]}$ 和$\pi_{C_B[z_1,z_2]}$ 的定义，可以看出存在一个常数 k，使得在循环的任意一次迭代中，对于任意 B 和 C_B，都有 $c(\pi_{C_B[z_1,z_1]})$ $c(\pi_{C_B[z_1,z_2]}) \leq k|C_B| + c(C_B)$。因此，计算 $\pi_{C_B[z_1,z_1]}$和 $\pi_{C_B[z_1,z_2]}$ 可以在 $O(k|C| + c(C))$ 时间内完成；此外，算法 1 的输出 π 满足 $c(\pi) \leq k|C| + c(C)$。

如引理 1 的证明中所述，树分解 T的树宽至多为2，可以在多项式时间内计算，因此 T 中有多项式多个袋。由于每轮循环的每一步都会移除 T 的一个袋，因此迭代次数是多项式级的。因此，算法 1的运行时间是多项式时间。

接下来，我们将说明如何使用算法 1，将任意 CPDL$^+$(TW$_2$)-程序转换为语义等价的 ICPDL 程序。给定一个 CPDL$^+$(TW$_2$)-程序 π，我们按如下步骤进行。如果不存在 $C[x,y] \in \mathrm{sub}(\pi)$，那么 π已经是一个 ICPDL 程序，处理完毕。否则，选取 $C[x,y] \in \mathrm{sub}(\pi)$，其中 C是一个 ICPDL 程序集合。应用算法 1，得到一个与 $C[x,y]$语义等价的 ICPDL 程序 π_*。在 π中将 $C[x,y]$ 替换为 π'，然后重复该过程。显然，此算法是多项式时间的，并能计算出所需的 ICPDL 程序。□

C 第 5 节缺失的细节5

C.1 定理 2 的 2 到 1 证明中遗漏的情形2

▷ $e = p$对于 $p \in \mathbb{P}$ 若 K， $v \models p$那么，由于存在一个将v映射到 v,的部分同态，可知 K,$v' \models p$。

▷ $e = a$对于 $a \in \mathbb{A}$若 K， u， $v \models a$那么，由于存在一个将$u \mapsto u'$和 $v \mapsto v$,映射到 K,u'， $v' \models a$的部分同态，可知。

▷ $e = \bar{a}$ 对于 $a \in \mathbb{A}$ 与情形 $e = a$类似。

▷ $e = \varphi_1 \wedge \varphi_2$若 K， $v \models \varphi_1 \wedge \varphi_2$那么对于 $i = 1$， 2，有 K， $v \models \varphi_i$；根据归纳假设，对于 $i = 1$， 2，有 K,$v' \models \varphi_i$，这意味着 K,$v' \models \varphi_1 \wedge \varphi_2$。

▷ $e = \pi_1 \cup \pi_2$ 与前一种情况类似。

▷ $e = \varphi?$ 如果 K， u， $v \models \varphi?$ 那么 $u = v$和 K， $u \models \varphi$。根据归纳假设， K,$u' \models \varphi$，因此K,u'， $v' \models \varphi$。

▷ $e = \pi$ 如果 $K, v \models \pi$，那么存在某个 $\tilde{v}$属于 K，使得 $(v, \tilde{v}) \in [\![\pi]\!]_K$。由于：(1) v和 $\tilde{v}$属于 K 中的同一个连通分量；(2) $(K, v) \rightarrow_{k+1} (K'v')$；以及 (3) $k+1 \geq 3$，因此存在某个 $\tilde{v}'$，使得 $(K, v, \tilde{v}) \rightarrow_{k+1} (K'v', \tilde{v}')$。根据 $\rightarrow_{k+1}$ 的定义，可以通过在 K 中从v “导航”到 $\tilde{v}$ 来得到这一结果。现在可以应用归纳假设，得到 $K'v', \tilde{v}' \models \pi$，从而得到 $K'v' \models \pi$。

▷ $e = \pi^*$假设 $K, u, v \models \pi^*$和 $K, u, v \rightarrow_{k+1} K'u', v'$。我们按迭代次数进行归纳。基例 0 对应于 $K, u, v \models \pi^0$， 当且仅当 $u = v$时成立；在这种情况下，根据 $\rightarrow_{k+1}$的定义， $u' = v'$，因此 K, u'， $v' \models \pi^0$。对于归纳情形， $K, u, v \models \pi^n$。注意，对于 $C = \{\pi(x_i, x_{i+1}) \mid 0 \leq i < n\}$， π^n 等价于 $\tilde{\pi} = C[x_0, x_n]$。由于 $\tilde{\pi}$ 的子表达式与 π^* 的子表达式相同（即 π及其内部的子表达式），我们可以用 $\tilde{\pi}$替换 π^n，并应用前一情形，从而得到 $K'u', v' \models \pi^n$。

▷ $\pi \circ \pi'$ 此情形与情形 π^* 类似。 □

C.2 以 k-半双模拟刻画

引理 5。 设 $k \geq 2$。给定 *Kripke*结构 K, K'， 它们具有有限度， 其世界 为 $u, v \in V(K)$ 以及$u', v' \in V(K')$， 以下各项等价

(1) 对于每个 CPDL$^+$(TW$_k$) 公式 φ， 我们有 $K, v \models \varphi$蕴含 K,$v' \models \varphi$；并且 *(2)* $K, v \rightarrow^{\leftrightarrow}_{k+1} K$,$v'$；以下各项等价

(1) 对于每个 CPDL$^+$($\mathcal{G}_{\mathrm{TW}_k}$)程序 π，我们有 $K, u, v \models \pi$蕴含 $K, u', v' \models \pi$；并且
(2) $K, u, v \rightarrow^{\leftrightarrow}_{k+1} K, u', v'$。

此外，有限度这一假设仅在1到2 的蕴含关系中才是必需的。

证明。我们按照定理 2 的证明，通过逆否命题进行推导。假设 $K, u, v \not\rightarrow^{\leftrightarrow}_{k+1} K' u', v'$。$((u, v), (u', v')) \notin \mathrm{Hom}_k(K, K')$ 的情况是显然的，因此我们假设相反的情况。这意味着 Spoiler 在从 $(s, ((u, v), (u', v')))$ 开始的 $\mathrm{G}[\rightleftharpoons_{k+1}]$ 中，拥有一种能够在有限轮数内获胜的策略。

Spoiler 的获胜策略仍然是一棵有限树，其顶点标记为 $\mathrm{G}[\rightleftharpoons_{k+1}]$ 中的局面。具体而言：(i) 根节点必须标记为 $(s, ((u, v), (u', v')))$；(ii) 每个标记为 $(d_i, \bar{u}, \bar{v})$ 的顶点，对于 $\mathrm{G}[\rightleftharpoons_{k+1}]$ 中的每一种可能移动，都有一个标记为 $(s, \bar{u}', \bar{v})$ 的子节点；(iii) 每个标记为 $(s, \bar{u}, \bar{v})$ 的顶点恰有一个子节点，其标签与 $\mathrm{G}[\rightleftharpoons_{k+1}]$ 中的一步移动相符（请记住，这次 Spoiler 有两种类型的移动）；(iv) 对每个叶节点（标记为 $(d_i, \bar{u}, \bar{v})$），在 $\mathrm{G}[\rightleftharpoons_{k+1}]$ 中都不存在从该局面出发的可能移动。由于 K, K' 都是有限度的，因此从 $\mathrm{G}[\rightleftharpoons_{k+1}]$ 中的任意局面出发的移动数量有限，树的分支数也有限。因此，策略树是有限的。

根据 Spoiler 的这种获胜策略，考虑由此得到的树 T 和标记 $\lambda : V(T) \to (\mathcal{V}(K)^{k+1} \cup \mathcal{V}(K')^{k+1})$，其构造过程如下：(1) 删除所有标记为 Duplicator 位置的顶点 v，并在 v 的父节点与 v 的所有子节点之间添加边；(2) 将每个顶点的标记投影到其第一个结构的各个分量上（即，如果策略树中 x 的标签为 $(s, \bar{u}, \bar{v})$，则为 $\lambda(x) = \bar{u}$）。如果对于某些 $\bar{u}, \bar{v}$，父节点的标签为 $(s, \bar{u}, \bar{v})$，而子节点的标签为 $(s, \bar{v}, \bar{u})$，我们就称这条边为“切换边”（注意，根据 $\mathrm{G}[\rightleftharpoons_{k+1}]$ 的定义，这意味着对于某些 u, v，有 $\bar{u} = (u, \ldots, u)$ 和 $\bar{v} = (v, \ldots, v)$）。因此，T 可以看作一棵“树的树”，其中每棵树都是删除切换边后 T 的一个连通分量。考虑从 T 中删除切换边后得到的所有树 $T_1, \ldots, T_n$。现在我们自底向上进行处理：对于每个满足仅其根节点可能存在一条来自另一棵树（的叶节点）的切换边的 T_i，我们按照定理 2 的证明中的方法进行处理，并构造一个合取程序 $C_i[x, x]$。对于归纳情形，假设存在一棵树 T_j，其叶节点可能存在指向其他树的切换边。然后，我们像之前一样为 T_j 构造合取程序，但额外添加一些原子：对于形如 $[v, \ell]_{\approx}$ 的 C_j 中的变量 z，其中 v 是与连接 T_j 和 T_i 的切换边相邻的 T_j 的叶节点，我们添加原子 $\neg C_i[x, x]?(z, z)$。通过与定理 2 的证明中类似的论证，我们可以假设每个 C_i 都只有有限个原子（这次利用的是“当前”结构的有限度）。

最后，可以证明，由处理所有合取程序 $C[x_s, x_t]$ 得到的T_i能够忠实地描述 Spoiler 的获胜策略，因此不可能出现K， u， $v \models C[x_s, x_t]$。

2 对 1 假设 $K, v \rightarrow^{\leftrightarrow}_{k+1} K' v'$。证明按照公式的结构归纳法进行，与定理 2 类似。根据备注 2，基础情形与定理 2 相同。若 $\varphi = \neg\psi$，则根据条目 2，我们有 $K' v' \rightarrow^{\leftrightarrow}_{k+1} K, v$，因此根据归纳假设，我们有 $K' v' \models \psi$蕴含 $K, v \models \psi$。后者等同于 $K, v \models \varphi$蕴含 $K' v' \models \varphi$。其余情形与定理 2 的证明完全相同，唯一的区别是，我们在公式可能包含否定的情况下使用归纳假设。□

[7]请注意，在此情形下，$[v, \ell]_{\approx} = [v, \ell']_{\approx}$ 对于所有 ℓ， ℓ'，因为$\mathrm{G}[\rightleftharpoons_{k+1}]$将这些切换移动限制为仅能应用于形如$(s, (u, \cdots u), (v, \cdots v))$ 的位置。

C.3 与基本模态逻辑模拟的联系令

$$K = (X, \{\to_a \mid a \in \mathbb{A}\}, \{X_p \mid p \in \mathbb{P}\}) \tag{3}$$
$$K' = (X', \{\to'_a \mid a \in \mathbb{A}\}, \{X'_p \mid p \in \mathbb{P}\})$$

是 Kripke结构。对于 $v \in X$和 $v' \in X'$我们称 K', v' 由机器学习模拟 K， v（记作 K， $v \rightharpoonup K', v'$），当且仅当存在 $Z \subseteq X \times X'$使得 vZv'，且对所有 uZu'，我们都有

(1) $u \in X_p$蕴含 $u' \in X'_p$对于所有 $p \in \mathbb{P}$；以及 (2) 如果 $u \to_a w$那么存在 $w' \in X'$使得 $u' \to'_a w'$且 wZw'。

众所周知，如果 K'具有有限度，那么 K， $v \rightharpoonup K', v'$当且仅当：在 v 中为真的任意正模态公式，在 v' 中也为真。

下面我们将证明，我们的 k-模拟 $\rightharpoonup_k$可以归约为 ML-模拟 $\rightharpoonup$。对于一个Kripke 结构 K 如（3）中所示，并且 $k > 0$ 定义如下的 Kripke 结构：

$$S_k(K) = (S_{K,k}, \{\to_i^{K,k} \mid i \in \mathbb{A}_k\}, \{S_r^{K,k} \mid r \in \mathbb{P}_{K,k}\})$$

其中

- $S_{K,k} = X^k$
- $\mathbb{A}_k = \{1, \ldots, k\}$
- $\mathbb{P}_{K,k} = (\mathbb{P} \times \{1, \ldots, k\}) \cup (\mathbb{A} \times \{1, \ldots, k\}^2) \cup \{1, \ldots, k\}^2$
- $\bar{u} \to_i^K \bar{v}$当且仅当存在$1 \le j \le k$使得$i \ne j$，且$w \in X$位于距离≤ 1处$u[j]$满足$\bar{v} = \bar{u}[i \mapsto w]$
- 对于 $p \in \mathbb{P}, \bar{v} \in S^{K,k}(p, i)$当且仅当 $\bar{v}[i] \in X_p$；对于 $a \in \mathbb{A}, \bar{v} \in S^{K,k}(a, i, j)$当且仅当 $\bar{v}[i] \to_a \bar{v}[j]$；并且 $\bar{v} \in S^{K,k}(i, j)$当且仅当$\bar{v}[i] = \bar{v}[j]$

定理 9。对于*Kripke*结构 K, K'， $\bar{w} \in W(K)^k$, 以及 $\bar{w}' \in W(K')^k$: $K, \bar{w} \rightharpoonup_k K', \bar{w}'$当且仅当$S_k(K)$ $\bar{w} \rightharpoonup S_k(K')$ $\bar{w}'$。

证明。可以证明，对于任意 $\bar{u} \in X^k$和 $\bar{u}' \in X'^k$项（1）中的 $\rightharpoonup$ 在 $S_k(K)$ 以及 $S_k(K')$等价于$(\bar{u}, \bar{u}') \in Hom_k(K, K')$。

假设 $K, \bar{v} \rightharpoonup_k K' \bar{v}'$通过 Z。首先注意，$(s, \bar{w}, \bar{w}')$ 是 G$[\rightharpoonup_k]$在 (K, K') 上的一个有效位置。我们证明，Duplicator 在 G$[\rightharpoonup_k]$中从 $(s, \bar{w}, \bar{w}')$ 出发有一个获胜策略；该位置是通过“复制” Z得到的：假设 $\bar{u}Z\bar{v}$，且 Spoiler 从 $(s, \bar{u}, \bar{v})$ 移动到 $(d_i, \bar{u}', \bar{v})$，其中 $\bar{u}' = \bar{u}[i \mapsto w]$，而 w 是 K中距 $\bar{u}[j]$为 ≤ 1 的一个世界，对某个满足 $i \ne j$ 的 $1 \le j \le k$。由于 $\bar{u} \to_i^K \bar{u}'$ 和 $\bar{u}Z\bar{v}$，根据 $\rightharpoonup$ 在 $S_k(K)$ 和 $S_k(K')$ 上的第 (2) 项，存在 $v' \in X'^k$ 使得 $\bar{u}' \to_i^{K'} \bar{v}'$ 和 $\bar{u}'Z\bar{v}'$。根据定义，$(s, \bar{u}', \bar{v}')$ 是一个有效位置，而 Duplicator 从 $(d_i, \bar{u}', \bar{v})$ 移动到 $(s, \bar{u}', \bar{v}')$。由于 Duplicator 总能移动，我们得出他有一个获胜策略。

对于另一个方向，假设 Duplicator 在 G$[\rightharpoonup_k]$ 中从 $(s, \bar{w}, \bar{w}')$ 出发有一个获胜策略，该策略由一棵没有叶节点的树 T 描述，其顶点标记为 G$[\rightharpoonup_{k+1}]$中的局面。具体而言：(i) 根节点必须标记为 $(s, \bar{w}, \bar{w}')$；(ii) 任何标记为 $(d_i, \bar{u}, \bar{v})$ 的顶点，都恰有一个子节点，其标签为 $(s, \bar{u}, \bar{v}')$，且与 G$[\rightharpoonup_{k+1}]$中的一步移动一致；(iii) 对于 G$[\rightharpoonup_{k+1}]$中每一种可能的移动，任何标记为 $(s, \bar{u}, \bar{v})$ 的顶点都有一个对应的子节点。令 $Z \subseteq X^k \times X'^k$ 满足： $\bar{u}Z\bar{v}$ 当且仅当 $(s, \bar{u}, \bar{v})$ 是 T 中某个节点的标签。显然， Z 满足条件 (1) 和 (2)，并且 $S_k(K)$ $\bar{v} \rightharpoonup S_k(K')$ $\bar{v}'$ 通过 Z 成立。□

The k-半双模拟关系也可以归约为ML-模拟。给定（3）中的Kripke 结构 K和 K'， 我们可以在多项式时间内构造Kripke 结构 $Q_k(K, K')$和$Q'_k(K, K')$， 使得

<escape>定理 10. 对于 Kripke 结构 $K, K\prime$, $\bar{w} \in V(K)^k$和 $\bar{w}\prime \in V(K\prime)^k$: $K, \bar{w} \rightarrow_k^{\leftrightarrow} K\prime, \bar{w}\prime$ 当且仅当$Q_k(K, K\prime)\ \bar{w} \rightharpoonup Q'_k(K, K\prime)\ \bar{w}\prime$。</escape>

因此我们得到

推论 11。对于 Kripke 结构 K、K', $\bar{w} \in V(K)^k$和 $\bar{w}' \in V(K')^k$: K、$\bar{w} \rightleftharpoons_k K'\bar{w}'$ 当且仅当 $Q_k(K$、$K')\ \bar{w} \rightharpoonup Q'_k(K$、$K')\ \bar{w}'$ 且 $Q_k(K'K)\ \bar{w}' \rightharpoonup Q'_k(K'K)\ \bar{w}$。

定义 $Q_k(K$、$K\prime)$ 和 $Q'_k(K$、$K\prime)$。设 K和 $K\prime$是如（3）中所述的 Kripke 结构，并且还满足 X和 X' 不相交（不失一般性）。Kripke 结构$Q_k(K$、$K\prime)$ 和 $Q'_k(K$、$K\prime)$ 定义如下：

$$Q_k(K, K') = (Q, \{\rightarrow_i \mid i \in \mathbb{A}_Q\}, \{Q_q \mid q \in \mathbb{P}_Q\})$$

其中

- $Q = X^k \cup X'^k$
- $\mathbb{A}_Q = \{1, \ldots, k\} \cup (X \times X') \cup (X' \times X)$
- $\mathbb{P}_Q = (\mathbb{P} \times \{1, \ldots, k\}) \cup (\mathbb{A} \times \{1, \ldots, k\}^2) \cup \{1, \ldots, k\}^2$
- 对于每个 $1 \le i \le k$, $\bar{u} \rightarrow_i \bar{v}$当且仅当存在 $Y \in \{X, X\prime\}$, 使得 $\bar{u}, \bar{w} \in Y^k$并且存在 $1 \le j \le k$满足 $i \ne j$, 且 $w \in Y$距 距离 ≤ 1处于 $\bar{u}[j]$且 $\bar{v} = \bar{u}[i \mapsto w]$
- 对于每个 $(v_1, v_2) \in (X \times X\prime) \cup (X\prime \times X), \bar{u}_1 \rightarrow_{(v_1, v_2)} \bar{u}_2$当且仅当 $\bar{u}_1[i] = v_1$且 $\bar{u}_2[i] = v_2$对于所有$1 \le i \le k$（注意，若 $v_1 \in X$，则 $\bar{u}_1 \in X^k$且 $\bar{u}_2 \in X\prime^k$；若 $v_1 \in X\prime$，则$\bar{u}_1 \in X\prime^k$且 $\bar{u}_2 \in X^k$）

- 对于 $(\tilde{X}, \tilde{\rightarrow}) \in \{(X, \rightarrow), (X\prime \rightarrow\prime)\}$, 若 $\bar{v} \in \tilde{X}^k$, 则对于 $p \in \mathbb{P}, \bar{v} \in Q_{(p,i)}$当且仅当 $\bar{v}[i] \in \tilde{X}_p$; 对于 $a \in \mathbb{A}, \bar{v} \in Q_{(a,i,j)}$当且仅当 $\bar{v}[i] \tilde{\rightarrow}_a \bar{v}[j]$; 并且 $\bar{v} \in Q_{(i,j)}$当且仅当 $\bar{v}[i] = \bar{v}[j]$

and

$$Q'_k(K, K') = (Q', \{\rightarrow'_i \mid i \in \mathbb{A}_Q\}, \{Q'_q \mid q \in \mathbb{P}_Q\})$$

其中

- $Q\prime = Q$
- 对于每个 $1 \le i \le k, \bar{u} \rightarrow\prime_i \bar{w}$, 其定义同 $Q_k(K, K\prime)$, 其中以 $\rightarrow\prime i$ 替换 $\rightarrow_i$
- 对于每个 $(v_1, v_2) \in (X \times X') \cup (X' \times X)$, $\bar{u}_2 \rightarrow' (v_1, v_2)\ \bar{u}_1$当且仅当 $\bar{u}_1 \rightarrow (v_1, v_2)\ \bar{u}_2$属于 $Q_k(K, K')$
- $Q'_{(p,i)}$、$Q'_{(a,i,j)}$ 和 $Q'_{(i,j)}$ 的定义与 $Q_k(K, K')$ 中的定义相同，只是将 $Q_{(p,i)}$ 替换为 $Q'_{(p,i)}$，将 $Q_{(a,i,j)}$ 替换为 $Q'_{(a,i,j)}$，并将 $Q_{(i,j)}$ 替换为 $Q'_{(i,j)}$

第 6 节缺失的详细信息

断言 4。 $(T, \mathbf{v})$ 是 $\hat{}K$ 的 树宽 至多为 $k-1$的 树分解。

证明。 $(T, \mathbf{v})$ 必须满足 A、B、C 三个条件，而这些条件是 树分解的条件 .

$\hat{}$条件 A 成立，因为每个 世界 $[u, v]_\approx$ 都属于 K，并且包含在 包 中，该包属于 $v_\circ$

条件 B 对任意 原子程序 $a \in \mathbb{A}$均成立，理由相同：对于满足 $([u, v]_\approx, [u', v']_\approx)$ 的 世界 对 $v = v'$, $[u, v]_\approx$ 和 $[u', v']_\approx$均包含在 包 中，该包属于 $v_\circ$

对于“连通性”条件 C，考虑 $\hat{}K$ 的任意世界 $[u, v]_\approx$，并注意到，对于来自 V_u 的任意顶点 v', $[u, v]_\approx \in \mathbf{v}(v') \upsilon = \{v' \in V(T) \mid (u, v) \approx (u, v')\}$。此外，$V_{u,v}$ 构成 T 的一个连通子树，而 T 的任何顶点 v' 若其包包含 $[u, v]_\approx$，则必须满足 $v' \in V_{u,v}$。因此，$(T, \mathbf{v})$ 验证了条件 C。

最后 $\underset{\text{sinc}}{\text{y,}}$ 每个 包 至多包含 k个元素，树宽 为 $(T, \mathbf{v})$ 至多为 $k-1$。 □

E 缺失的章节细节7

在一个完全树 T的语境中，节点 $va \in dom(T)$若满足 $a \in \Sigma_E$，则称为 a-后继，属于 v；而 v则是 a-前驱，属于 va。

E.1 双向交替奇偶树自动机的定义

要定义 TWAPTA，我们需要先介绍一些预备知识。对于有限集 X，我们用 $B^+(X)$ 表示所有正的布尔公式组成的集合，其中 X 的元素用作变量。允许使用常量 true 和 false，即对于任意集合 X，都有 $\text{true}, \text{false} \in B^+(X)$。子集 $Y \subseteq X$ 可以通过显然的方式视为一种赋值：将 Y 的所有元素赋值为 true，将 $X \setminus Y$ 的所有元素赋值为 false。对于边字母表 Σ_E，令 $\Sigma_E = \{\bar{a} \mid a \in \Sigma_E\}$ 为 Σ_E 的一个不相交副本。对于 $u \in \Sigma^*$ 和 $d \in \Sigma_E \cup \Sigma_E \cup \{\varepsilon\}$，定义如下：

$$u \cdot d = \begin{cases} ud & \text{if } d \in \Sigma_E \\ u & \text{if } d = \varepsilon \\ v & \text{if there is } a \in \Sigma_E \text{ with } d = \bar{a} \text{ and } u = va \\ \mathit{undefined} & \text{otherwise} \end{cases}$$

一个双向交替奇偶树自动机（*TWAPTA*）在完全 Σ_N-标记的 Σ_E-树上是一个元组 $T = (S, \delta, s_0, Acc)$，其中

- S 是一个有限的非空状态集，
- $\delta: S \times \Sigma_N \to B^+(mov(\Sigma_E))$ 是转移函数，其中 $mov(\Sigma_E) = S \times (\Sigma_E \cup \Sigma_E \cup \{\varepsilon\})$ 是移动的集合，
- $s_0 \in S$ 是初始状态，并且
- $Acc \mid S \to \mathbb{N}$是优先级函数。

对于 $s \in S$和 $d \in \Sigma_E \cup \overline{\Sigma_E} \cup \{\varepsilon\}$，我们将对应的移动记作 s, d。直观地说，移动 s, a，其中 $a \in \Sigma_E$，表示自动机将自身的一个副本以状态 s发送到当前树节点的 a-后继节点。类似地，$s, \bar{a}$ 表示将一个副本发送到 a-前驱节点（如果存在），而 s, ε 表示停留在当前节点。形式上，TWAPTA 的行为通过运行来定义。设 T 为上述 TWAPTA，$T \in$ 树(Σ_N, Σ_E)，$u \in \Sigma_E{}^*$为一个节点，$s \in S$为 T 的一个状态。T 在 T上的一个 (s, u)-运行是一个（不一定完整的）$(S \times \Sigma_E{}^*)$-标记树 T_R，满足

- $T_R(\varepsilon) = (s, u)$，并且
- 对于所有 $\alpha \in dom(T_R)$，如果 $T_R(\alpha) = (p, v)$ 且 $\delta(p, T(v)) = \theta$，那么存在一个子集 $Y \subseteq mov(\Sigma_E)$，它满足 θ，并且对于所有 $(p, d) \in Y$，$v \cdot d$都有定义，且存在 α在 T_R中的一个后继 β，使得 $T_R(\beta) = (p, v \cdot d)$。

我们称一个 (s, u)-运行 T_R是 成功的，如果对于其中的每条无限路径 $\alpha_1\alpha_2\cdots$，T_R以下数值都是偶数（假设该路径从根节点开始）：

$$\min\{Acc(s) \mid s \in S \text{ with } T_R(\alpha_i) \in \{s\} \times \Sigma_E{}^* \text{ for infinitely many } i\}$$

对于 $s \in S$定义 $[\![\mathrm{T}\ s]\!] := \{(T, u) \mid T \in$树$(\Sigma_N, \Sigma_E)\ u \in \Sigma_E{}^*$，且 T 在 T上存在一条成功的 (s, u)-运行$\}$，并且 $[\![\mathrm{T}]\!] := [\![\mathrm{T}\ s_0]\!]$。

现在，T 所接受的语言 L定义为

$$L(\mathcal{T}) := \{T \in tree(\Sigma_N, \Sigma_E) \mid (T, \varepsilon) \in [\![\mathcal{T}]\!]\}.$$

对于一个 TWAPTAT $= (S, \delta, s_0, \mathrm{Acc})$，我们将其大小 $|\mathrm{T}| := |S|$定义为其状态数量，并将其指标 i(T)定义为$\max\{\mathrm{Acc}(s) \mid s \in S\}$。转移函数 $|\delta|$的大小是出现在 δ值域中的所有正的布尔函数长度之和。

E.2 定义 基于 TWAPTA 的非确定性有限自动机 TWAPTA

形式上，一个非确定有限自动机(非确定有限自动机) $\mathcal{A}$ 定义在TWAPTA $\mathcal{T}$ 上，状态空间为 S，是一个元组$(Q, p_0, q_0, \to_{\mathcal{A}})$，其中 Q是一个状态有限集，p_0而 q_0是两个选定的状态，称为

初始 和 最终 分别，并且 $\to_{\mathcal{A}}$ 是以下形式的标签转移集合，其中 $q, q' \in Q$ 和 $a \in \Sigma_E$: (i) $q \xrightarrow{a}_{\mathcal{A}} q'$, (ii) $q \xrightarrow{\bar{a}}_{\mathcal{A}} q'$ 或 (iii) $q \,\mathcal{T}\!\longrightarrow^{s} q'$ 其中 $s \in S$。

第三类转移称为测试转移。NFA 在完整的 Σ_N 标记 Σ_E 树的节点集合上定义二元关系。为明确这一点，设 $T \in \mathrm{tree}(\Sigma_N, \Sigma_E)$，并将定义 $\Rightarrow_{\mathcal{A},T} \subseteq (\Sigma_E{}^* \times Q) \times (\Sigma_E{}^* \times Q)$ 为满足以下条件的最小关系：对于所有 $u \in \Sigma_E{}^*$、$a \in \Sigma_E$、p、$q \in Q$ 和 $s \in S$，有：(i) $(u,p) \Rightarrow_{\mathcal{A},T} (ua, q)$，若 $p \xrightarrow{a}_{\mathcal{A}} q$；(ii) $(ua,p) \Rightarrow_{\mathcal{A},T} (u,q)$，若 $p \xrightarrow{\bar{a}}_{\mathcal{A}} q$；(iii) $(u,p) \Rightarrow_{\mathcal{A},T} (u,q)$，若 $p\,\mathrm{T}\!\longrightarrow^{s}_{\mathcal{A}} q$ 且 $(T,u) \in [\![\mathrm{T}\ s]\!]$。

定义 $[\![\mathcal{A}]\!] := \{(T,\ u,\ v) \mid T \in \text{树}(\Sigma_N,\ \Sigma_E),\ u,\ v \in \Sigma_E{}^*,$ 以及 $(u,\ p_0) \Rightarrow^*_{\mathcal{A},\ T} (v,\ q_0)\}$。

E.3 自动机界

首先，我们证明，从一个合取程序到基于$\mathcal{Q}$的 ICPDL 程序并集的归约，可以得到以下界限。

断言 6. 设 $\pi' = \bigcup_{T_C,f} \hat{}\pi_{T_C,f}$ 是由 *ICPDL*$^+$ 合取程序 π 生成的、基于先前构造的自动机的 *ICPDL* 程序。那么：*(1)* π' 的大小[8] 为 $exp(cw(\pi))$；*(2)* $iw(\pi') \le cw(\pi)^{O(1)}$；*(3)* $neg\text{-}depth(\pi') = neg\text{-}depth(\pi)$。

证明。注意，如果 $\pi = C[x_s, x_t]$，每个 $\hat{}\pi_{T_C,f}$的大小均为关于 $\mathrm{cw}(\pi)^{O(1)}$ 的多项式，其中记住$\mathrm{cw}(\pi)$ 个 C 的原子，这是将定理1 的多项式时间翻译应用于大小为二次的合取程序$C_{T_C,f}$的结果。于是，至多有 $2^{\mathrm{cw}(\pi)^{O(1)}}$ 个这样的程序 $\hat{}\pi_{T_C,f}$，每个程序的大小均为关于 $\mathrm{cw}(\pi)^O(1)$ 的多项式。这给出一个关于$\mathrm{cw}(\pi)^{\mathrm{cw}(\pi)^{O(1)}}$ 的界，该界属于 $\exp(\mathrm{cw}(\pi))$。$\mathrm{iw}(\pi')$关于 π 为多项式这一事实，源于每个 $\hat{}\pi_{T_C,f}$的大小均为多项式，并且 $\mathrm{iw}(\bigcup_{T_C,f} \hat{}\pi_{T_C,f})$ 等于 $\max_{T_C,f}\{\mathrm{iw}(\hat{}\pi_{T_C,f})\}$。□

结合引理 3 的大小界与备注 4、断言 6 以及定理 TWAPTA的复杂性6 ，我们得到：

命题 8。 对于一个 *TWAPTA* $\mathcal{T}$ 及其转移函数 δ 以及一个 CPDL$^+$ 公式 φ，我们可以在以下时间内判定

$$exp\big(|\mathcal{T}| + i(\mathcal{T}) + |\varphi|^{\mathrm{CW}(\varphi)}\big) \cdot |\delta|^{O(1)}$$

是否存在某个 $T \in L(\mathcal{T})$ 使得 $\varepsilon \in [\![\varphi]\!]_T$。因此，$\omega$正则树可满足性问题对于 CPDL$^+$ 属于 2指数时间，而 ω正则树可满足性问题对于 $\{\varphi \in \mathrm{CPDL}^+ \mid \mathrm{CW}(\varphi) \le n\}$属于 指数时间，对任意 $n \in \mathbb{N}$。

证明。 我们首先通过对 φ归纳，证明所构造的 TWAPTA满足以下界：

断言 7。 $|\mathcal{T}(e)| \le 9 \cdot |e|^{2 \cdot \mathrm{CW}(e)+1}$，且 $i(\mathcal{T}(e)) \le$否定深度(e)。

基例 对任意 表达式 e，若其中没有 合取程序，则 e 是一个 CPDL 表达式。此时，引理 TWAPTA3 中等价的 $\mathrm{IW}(e) = 1$的大小界

我们得到 $|\mathcal{T}(e)| \le 9 \cdot |e|^3$，以及 $i(\mathcal{T}(e)) \le$否定深度(e)。

I归纳情形：若 e是一个 合取程序 $C[x_s, x_t]$，对于 $C = \{\pi_i(x_i, y_i) \mid 1 \le i \le n\}$。

根据归纳假设，我们有 $|\mathrm{T}(\pi_i)| \le 9 \cdot |\pi_i|^{2 \cdot \mathrm{CW}(\pi_i)+1}$，且 $i(\mathrm{T}(\pi_i)) \le$否定深度(π_i)，并且$|\mathrm{T}(e)| \le (\sum_i |\mathrm{T}(\pi_i)|) + |\mathrm{T}(\hat{}e)|$，其中 $\hat{}e$ 是一个以 $\mathrm{T}(\pi_i)$ 为元素、与 e 等价的 ICPDL 程序，定义见第 7.2.1 节。根据断言 6 的界，$|\hat{}e| \le 2^{\mathrm{cw}(e)^c}$，其中某个 $c \in \mathbb{N}$。因此，应用引理 3 的自动机界，我们有 $|\mathrm{T}(\hat{}e)| \le 9 \cdot 2^{2 \cdot \mathrm{cw}(e)^c+1} \le 9 \cdot 2^{2 \cdot \mathrm{CW}(e)^c+1}$。

[8]“大小”指不计其所基于的 非确定有限自动机/TWAPTA的大小。

因此，

$$\begin{aligned}|\mathcal{T}(e)| &\le \Big(\sum_i |\mathcal{T}(\pi_i)|\Big) + 9\cdot 2^{2\cdot \mathrm{CW}(e)^c+1}\\ &\le \Big(\sum_i 9\cdot |\pi_i|^{2\cdot \mathrm{CW}(\pi_i)+1}\Big) + 9\cdot 2^{2\cdot \mathrm{CW}(e)^c+1}\\ &\le 9\cdot \Big(\sum_i \cdot|\pi_i|^{2\cdot \mathrm{CW}(e)+1} + \cdot 2^{2\cdot \mathrm{CW}(e)^c+1}\Big)\\ &\le 9\cdot \Big(\sum_i |\pi_i|\Big)^{2\cdot \mathrm{CW}(e)+1}\\ &= 9\cdot |e|^{2\cdot \mathrm{CW}(e)+1}\end{aligned}$$

这就完成了命题7的证明。现在，结合定理6中TWAPTA的复杂性与命题7 的界限，我们得到，对于$\mathcal{T}(\varphi)$，其空性可在以下时间内检测

$$\begin{aligned}exp(|\mathcal{T}(\varphi)| + i(\mathcal{T}(\varphi)))\cdot|\delta|^{O(1)} &\le exp(9\cdot|\varphi|^{2\cdot \mathrm{CW}(\varphi)+1} + neg\text{-}depth(\varphi))\cdot|\mathcal{T}(\varphi)|^{O(1)}\\ &\le exp(|\varphi|^{\mathrm{CW}(\varphi)})\end{aligned}$$

由此可知（同样根据定理6），对于转移函数为 δ的TWAPTA T，以及一个CPDL$^+$ 公式 φ，判定 T $\cap$T(φ) 是否为空可在$exp\big(|\mathrm{T}| + i(\mathrm{T}) + |\varphi|^{\mathrm{CW}(\varphi)}\big)\cdot|\delta|^{O(1)}$内完成。□

F 第 8 节缺失的细节8

命题9。 命题动态逻辑的模型检测问题CPDL$^+$ 是DP-困难的，且属于P^{NP}。

证明。下界可通过从模型检测布尔组合问题合取查询的模型检测问题进行直接归约得到。对于上界，只需注意到定理8 （第1项）的证明中的算法运行于 P^{NP}。□

定理 8。 对于任意由连通$\mathcal{G}$ 组成的图：
(1) 若 $G \subseteq TW_k$ 对某个 k 成立，则 *CPDL$^+$(G)* 的模型检测问题是 *PTime*-完全；*(2)* 否则，在假设 G 是递归可枚举的且 $W[1] \neq FPT$ 的条件下，*CPDL$^+$(G)* 的模型检测问题不属于多项式时间。即使对于 *CPDL$^+$(G)* 的正片段，这一结论仍然成立。

证明。用于证明 CPDL$^+$(TW$_k$) 模型检测属于多项式时间的过程，是一种经典的动态规划算法。给定一个公式 φ 和一个有限 Kripke 结构 k，我们根据 ψ 的子表达式标记，迭代地用 φ 的子表达式 ψ 标记 K 的所有世界。对于每个公式 $\psi \in sub$（φ），我们使用一个一元关系（即一个集合） $U_\psi \subseteq W$（K）；对于每个程序 $\pi \in sub$（φ），我们使用一个二元关系$B_\pi \subseteq W(K) \times W(K)$，所有关系都初始化为空关系。我们首先处理所有原子命题 $p \in \mathrm{sub}(\varphi)$ 和原子程序 $a \in \mathrm{sub}(\varphi)$：我们设置$U_p = X_p$和 $B_a = \to_a$。对于原子程序 $\bar{a} \in sub$（φ）的逆，我们设置 $B_{\bar{a}} =$（$\to_a$）$^{-1}$。现在，取任意一个公式 $\psi \in sub$（φ），使得它在 sub（ψ）中的所有子表达式（除 ψ 外）都已处理。我们按如下方式处理 ψ：

- 如果 $\psi = \psi_1 \wedge \psi_2$，则设置 $U_\psi = U_{\psi_1} \cap U_{\psi_2}$
- 如果 $\psi = \neg\psi'$，则设置 $U_\psi = W(K) \setminus U_{\psi'}$，
- 如果 $\psi = \pi$，则设置 $U_\psi = \{w \in W(K) : \exists w' s.t. (w, w') \in B_\pi\}$。

很容易看出，借助合适的数据结构，这些操作中的每一个都可以在线性时间内完成（甚至可以达到多项式时间）。现在，任取一个程序 $\pi \in sub(\varphi)$，使得它的所有子表达式 均来自$sub(\pi)\setminus\{\pi\}$且已完成处理。我们按照以下方式处理 π：

- 如果 $\pi = \pi_1 \star \pi_2$ 对于 $\star \in \{\circ, \cup\}$ 成立，则令 $B_\pi = B_{\pi_1} \star B_{\pi_2}$，
- 若 $\pi = (\pi')^*$，则设置 $B_\pi = B^*_{\pi'}$，
- 若 $\pi = C[x_s, x_t]$，我们将 $C[x_s, x_t]$ 视为在已处理的关系上执行的 合取查询，以填充 B_π。

显然，前两项可以在二次时间内完成。对于最后一项，众所周知，树宽 $\leq k$ 的合取查询 的求值可在多项式时间内完成，见 [7,定理 3]（更准确地说是 LOGCFL-完全的，见 [13, 定理 6。12]），方法是对查询的树分解 进行自底向上的处理。

总而言之，这得到了一种关于 φ 呈线性时间且关于 K 呈多项式时间的算法，其中多项式的次数为 $k+1$，当 $\varphi \in \mathrm{CPDL}^+$（$\mathrm{TW}_k$）时。

下界来自PTIME-困难的 模态逻辑 模型检测（例如参见 [19,命题 5]）。

2. 这是因为，对于合取查询，已有类似结论：除非 FPT = W[1] [15, 推论 19].[9] 我们知道，当且仅当公式 $C[x, x]$ 在某个世界中成立时，布尔合取查询 q 才在一个结构中为真；其中，C 是 q 的原子集合，而 x 是 q 的任意变量。

这个 多项式时间归约将 合取查询的模型检测 归约为 命题动态逻辑的模型检测，即 $\mathrm{CPDL}^+(\mathcal{G})$，这意味着CPDL的模型检测在 $\mathrm{CPDL}^+(\mathcal{G})$ 不可能属于 多项式时间，前提是假设W[1] ≠ FPT。[10] □

G 扩展讨论

有限可满足性。 请注意，在此处研究的可满足性问题中，Kripke 结构可能是无限的。即使对于 loop-CPDL，有限可满足性问题是否可判定仍是一个众所周知的公开问题 [12,§7]。

无限状态模型检测。无限状态模型检测（即，针对某些有限表示的无限Kripke结构的模型检测）已针对CPDL [11] 和ICPDL [12]进行了研究。尽管我们认为，上界结果可以扩展到CPDL+，但我们将这一问题留待未来工作。

连通性。 我们对合取程序 所假设的连通性限制，正如前面所提到的，并非必要条件。所有上界均适用于非连通的 程序，而且可以很容易地调整该刻画，以涵盖这些更一般类型的查询。我们采用连通性条件的原因在于，这样便于与 命题动态逻辑 逻辑系列进行比较，因为该系列只能在一个结构的单个连通分量内进行移动。

任意元数程序。在我们的研究过程中，我们已经证明，所考虑的对象不仅可以是二元程序 $C[x, y]$，实际上也可以是任意元数的程序 $C[\bar{x}]$。这可以被视为一种查询语言，能够涵盖整个 CRPQ 语言，甚至是正则查询语言，并支持任意元数的查询（而不仅仅是元数为 ≤ 2 的查询，如图1所示）。我们还可以考虑以显而易见的方式，将组合、并集和 Kleene 星算子扩展到 $2n$元关系上。这将产生一种表达能力更强的形式体系。目前，我们尚不清楚可满足性问题是否可判定，因此将这一研究方向留待未来工作。另一个研究方向是放弃 Kleene 星，转而采用一种更一般的传递闭包形式，允许使用“参数”，其方式与

[9]这里我们采用连通合取查询这一事实并不起作用，而 [15] 的结果对于连通合取查询[10]对于参数化模型检测也可以作出类似的类比，其中参数是公式的大小。在我们的情形中，我们会得到参数化模型检测问题 $\mathrm{CPDL}^+(\mathcal{G})$ 是W[1]-困难的。

与一阶逻辑中的传递闭包相同的方式（这一想法在 [5] 另一种语境下进行了探讨）。

常量。 由于本研究旨在涵盖某些查询语言，例如 合取查询 和合取正则路径查询，因此能够访问 常量是合理的。在此语境下，常量通常被建模为名词 (即，一种 原子命题 ，其为真时至多只能在一个 世界中成立。我们认为，在 CPDL$^+$ 中可以轻松处理名词，但代价是增加树状模型性质的 树宽。这些细节留待未来工作探讨。