arXiv:2603.24597v1 [cs.CC] 9 Mar 2026

自适应数据结构选择中检测和修复结构性过度规定的算法障碍

Faruk Alpay* Levent Sarioglu†

摘要

我们研究了在自适应数据结构选择中检测和修复一种系统性结构性过度规定形式的算法障碍。输入实例会导出一个隐含的工作负载特征——排序、稀疏性、动态性、局部性或子串结构；即使测量得到的工作负载证据仅支持其中一个严格子集，候选实现仍可能因符合这一完整特征而受到青睐。在这样一种模型下，成对评估器偏好能够实现隐含特征的实现，我们证明这种偏好会同时传递到基准测试聚合和 Bradley– Terry–Luce 拟合中。随后，我们确立了两个结果。第一，可判定性存在一道明确的边界：在无界输入域上，判断表示选择流水线是否表现出超出测量证据支持度的结构承诺是不可判定的（可由停机问题归约而得）；而在有限域上，该问题可判定，但枚举的代价为指数级。第二，不动点障碍：在一项保守的程序变换约束下——修复算子必须保持已与证据相符的流水线不变——任何全可计算修复算子都存在一个过度指定的不动点，该不动点可通过克莱尼递归定理构造。这些障碍在性质上不同于数据结构设计中的经典下界：单元探测下界和动态图库下界约束的是有限工作负载上操作的回答效率，而我们的结果约束的是跨流水线族统一检测和修复过度规定的可能性。

关键词：数据结构；图算法；字符串算法；动态数据结构；算法选择；Bradley–Terry–Luce；可判定性；Rice 定理；Kleene 递归定理；保守程序变换

1 引言

为某种工作负载选择一种实现或表示——在邻接表与邻接矩阵、平衡搜索树与哈希、后缀数组与后缀树，或静态与动态图原语之间做出选择——是数据结构与算法设计中的核心任务 [1, 3, 5, 9]。现代系统越来越多地根据执行轨迹、基准测试结果或学习型成本模型做出这类选择，通常通过比较候选实现的成对结果，再将这些结果汇总为单一分数或排名 [2, 13, 10, 11]。

数据结构设计的复杂性已得到充分研究：动态数据结构存在更新/查询权衡 [12], ；动态图算法的复杂性前沿已得到精细刻画 [5, 6], ；字符串索引结构则实现了不同的预处理/查询模式 [7, 8, 9]。这些结果构成了复杂性壁垒：它们界定了在固定的有限实例或操作序列上支持给定操作集的效率上限。

本文指出了另一类障碍——可计算性障碍——当表示选择过程受到系统性过度指定时，就会出现这种障碍。具体而言，我们形式化定义了一种结构偏好通道：当输入实例呈现出某种工作负载特征时，评估者

*巴赫切希尔大学计算机工程系。 faruk.alpay@bahcesehir.edu.tr
†巴赫切希尔大学计算机工程系。 levent.sarioglu@bahcesehir.edu.t r

可能会偏好能够实现完整隐含特征的实现，即使测量证据仅足以支持其中一个较小的子集。例如，在没有对抗性更新证据的情况下，稀疏图工作负载可能会被映射到高度动态的图机制；字符串处理任务可能会因微弱的局部性线索而触发繁重的基于后缀的索引；字典工作负载则可能因偶然出现的有序提示而被分配有序搜索结构。在标准的成对聚合模型下，这种偏好会以可证明的方式传递到学习得到的评分函数中（命题 3.2 和 3.4）。于是，算法层面的问题是：能否检测这种过度指定？能否对其进行统一修复？

我们的贡献。 我们确立了两个主要结果，并对各自的适用范围进行了明确审查。

(i) **可判定性边界（第 5节）。** 在无界输入域上，判定表示选择流水线是否存在超出测量证据支持度的结构承诺这一问题不可判定（定理 5.6），这是通过从停机问题归约得出的。在有限域上，同一谓词可通过穷举判定，但代价为指数级（命题 5.9）。这构成了一个明确的可计算性边界。

(ii) **不动点障碍（第 6节）。** 在对程序变换器施加保守约束的情况下——修复算子必须保持与证据相符的流水线不变——任何全可计算修复算子都会存在一个过度指定的不动点（定理 6.6），其构造基于克莱尼递归定理。没有任何保守修复算子能够在所有流水线中统一消除过度指定。

这些障碍在性质上不同于经典数据结构下界：单元探测和动态图库下界约束有限工作负载下的时间与空间权衡；我们的结果则是可计算性障碍，在无界流水线族上，无论投入多少计算资源都无法克服。

论文结构。 第 2 节介绍形式化框架。第 3 节和 4 节明确了过度指定模型：它们证明了不当的结构性偏好会通过基准测试聚合传播，且这种传播具有不对称性。这些是前提条件，而非贡献——它们形式化了本文主要结果所要解决的问题。第 5 节和 6节包含主要的算法贡献。

1.1 相关工作

自适应数据结构与算法选择。 Rice [2]提出了算法选择问题，强调算法的选择取决于实例特征和性能模型。对工作负载敏感的自调整结构（如动态树 [3]和伸展树 [4], ）表明，表示在结构层面至关重要。基于学习和基准测试的数据系统使这种依赖关系更加明确 [10, 11]。本文指出了另一种障碍：问题不在于如何高效适应，而在于检测不当结构承诺的不可判定性 以及保守修复的不完备性 。

图算法与动态数据结构。 动态连通性、最短路径和动态树表示展现出丰富的更新/查询权衡 [3, 5, 6]。相关研究探讨了如何高效支持不断变化的工作负载，而我们的关注点是元算法层面的：能否对选择流水线进行统一审查，以检查其中是否存在结构性过度规定。

字符串算法与索引结构。 Knuth–Morris–Pratt [7], 、后缀数组 [8], 和后缀树 [9]等经典字符串处理成果，展示了针对同一底层任务的多种相互竞争的表示范式。我们的框架将这些替代方案视为

表示选择流水线中的候选方案，并探究选择器何时会相对于证据承诺过多的结构。

程序修复与可计算性。 一致查询回答 [15, 16]研究如何在最小修复下查询不一致数据。Rice 定理 [17] 以及通过递归定理进行的不动点构造，是本文所用可计算性障碍的基础。我们的保守修复算子（定义 6.2）是最小修复的程序变换类比。

2 形式框架

2.1 实例、实现与可计算选择流水线

设有限字母表为 Σ。令 $\Sigma^* = \Sigma^*$ 表示由 Σ构成的所有有限字符串的集合。我们将工作负载实例建模为字符串 $x \in \Sigma^*$，将候选实现或表示建模为字符串$y \in \Sigma^*$。

定义 2.1（表示选择流水线类）。 一个表示选择流水线类 $\mathcal{F}$ 是全可计算函数 $f\colon \Sigma^* \to \Sigma^*$ 构成的递归可枚举集。等价地，存在一个有效枚举 $\{f_e\}_{e\in\mathbb{N}}$，使得 $\mathcal{F} = \{f_e : e \in \mathbb{N}\}$，且每个 f_e 都在所有输入上停机。

备注 2.2。 将定义域限制为 $\Sigma^{\le n}$ 会使其成为有限集，并使许多存在性谓词变得可判定。这对于第 5 节中的可判定性边界至关重要。

2.2 工作负载签名、测量证据支持度与结构兼容性 y

定义 2.3（工作负载签名提取器）。 工作负载签名提取器是一个全可计算函数 *:*，它将每个实例 映射到一个有限的结构特征字符串集合 *()*。这些特征可以编码有序访问、稀疏性、在线更新、局部性或子串复用等属性。

定义 2.4（测量证据支持度提取器）。 测量证据支持度提取器是一个全可计算函数 *:*，满足对于所有，*() ()*，其中 *()* 表示观测轨迹、基准测试或规范实际支持的结构特征。

定义 2.5（结构兼容关系与得分）。 结构兼容关系 是一个全可计算函数 $V\colon \Sigma^* \times \Sigma^* \to \{-1, 0, +1\}$。结构兼容性得分 是$v(x, y) = \sum_{s\in S(x)} V(y, s)$。超出证据支持度的过度指定得分 是

$$v_{\mathrm{bw}}(x, y) = \sum_{s\in S(x)\setminus W(x)} V(y, s). \tag{1}$$

正值表示实现 y 实例化了 x 的完整签名所建议、但测量证据并不支持的结构。

说明 2.6。 任何形式为“全可计算函数的输入/输出行为的性质”的性质，都受无界定义域上的 Rice 式限制（第5节）。

2.3 基准测试聚合与成对分数拟合

定义 2.7（基准测试档案与实现聚合）。固定 k 评估者、实例 x和候选集 Y_x。基准测试档案是 $\pi = (\succ_1, \ldots, \succ_k)$，其中每个 $\succ_i$对候选实现进行排序。实现聚合规则 A 将此类档案映射为聚合排序。

定义 2.8（决定性基准族）。 $C \subseteq [k]$ 对（y、 y'）是决定性的，当且仅当（$\forall i \in C$： $y \succ_i y'$）$\Rightarrow y \succ_A y'$，无论其余评估者的偏好如何。

定义 2.9 (逻辑斯蒂成对实现似然**)**。 设 $\sigma(t) = \frac{1}{1+e^{-t}}$。 评分函数 $R\colon \Sigma^* \times \Sigma^* \to \mathbb{R}$ 诱导出 $\Pr[y \succ y' \mid x] = \sigma(R(x,y) - R(x,y'))$。Bradley–Terry–Luce 拟合为这类结果提供了一种标准的成对评分模型 [13, 14]。

3 过度规定模型：通过基准测试聚合实现继承

本节将说明算法结果所依据的结构性过度规定模型。这里的命题并非我们的主要贡献；它们形式化地确立了这样一个前提：不合理的结构性偏好确实 会通过标准基准测试聚合传播，从而表明检测和修复是值得研究的问题。

3.1 决定性下的确定性继承

定义 3.1（签名单调评估者）。 评估者排序 $\succ_i$ 在（y, y'）上是签名单调的，如果 $v(x,y) > v(x,y')$ $\Rightarrow y \succ_i y'$。

命题 3.2（决定性基准族下的继承）。 固定 x、 Y_x，其中 $|Y_x| \geq 3$，以及一个实现聚合规则 A。如果存在一个对于 （y、 y'） 具有决定性的族 C，且其中每个$i \in C$都在 （y、 y'） 上签名单调，那么对于每个基准测试档案 π： v（x、 y） $> v$（x、 y'） $\Rightarrow y \succ_A y'$。

证明。 签名单调性迫使每个 $i \in C$ 对 $y \succ_i y'$进行排序；决定性则迫使 $y \succ_A y'$。 □

3.2 成对得分拟合下的统计继承

定义 3.3（随机效用评估者）。 对于评估者 i，潜在效用为 $u_i(x, y) = q_i(x, y) + \alpha_i v(x, y)$，其中 q_i 独立于结构性偏好， $\alpha_i \in \mathbb{R}$ 为签名敏感度。成对结果遵循 $\Pr_i[y \succ y' \mid x] = \sigma(u_i(x, y) - u_i(x, y'))$。

命题 *3.4*（质量控制下的成对拟合继承）。假设对所有 i、 $\alpha_i \geq 0$ 和 $\bar{\alpha} > 0$，定义 *3.3* 满足$q_i(x,y) = q_i(x,y')$。令 $\Delta v = v(x,y) - v(x,y')$。若 $\Delta v > 0$，则 $\Pr[y \succ y' \mid x] > \frac{1}{2}$，且该概率随 Δv 严格递增。

证明。 在质量控制下，$\Pr[y \succ y' \mid x] = \frac{1}{k}\sum_i \sigma(\alpha_i \Delta v)$。每一项 $\geq \frac{1}{2}$；至少有一项为严格不等式。取平均可得 $> \frac{1}{2}$。严格单调性： $\frac{d}{d\Delta v}\sigma(\alpha_i \Delta v) = \alpha_i \sigma(\alpha_i \Delta v)(1 - \sigma(\alpha_i \Delta v)) \geq 0$，当 $\alpha_{i_0} > 0$ 时为严格不等式。 □

推论 3.5 (学习得分继承结构性过度规定**)**。 在命题 *3.4*的 $\widehat{\quad}$ $\widehat{\quad}$条件下，任何一致的成对得分估计量都满足 $\Delta v > 0 \Rightarrow R(x, y) > R(x, y')$ 于大样本极限下成立。

证明。证 一致性给出 $\widehat{p} \to p^\star > \frac{1}{2}$；逻辑链接函数将其转换为正的得分差 e. □

注释 3.6。 命题 3.4 在混合符号条件下可能失效 α_i，也可能因工作负载轨迹测量不当、配对覆盖不足，或明确惩罚结构复杂度的正则化项而失效。

4 过度规定模型：不对称供给不足惩罚

模型还需指出，缺失结构与添加额外结构未必受到对称的惩罚。在算法工程中，配置不足通常被视为比配置过度代价更高：未能提供动态更新或局部性支持，所受惩罚可能重于携带冗余机制。这种不对称性有助于解释为何修复比检测更困难：修复算子必须消除方向相关的失真，而非仅仅减去一个常数。

定义 4.1（分段线性后悔加权）。评估者 i 对 $\delta \geq 0$ 使用 $w_i(\delta) = \alpha_i \delta$，对 $\delta < 0$ 使用 $w_i(\delta) = -\lambda_i \alpha_i |\delta|$，其中 $\alpha_i > 0$ 且 $\lambda_i \geq 1$ [19, 18]。给定 $u_i(x, y) = q_i(x, y) + w_i(v(x, y))$，定义总体得分 $R(x, y) = \frac{1}{k} \sum_i u_i(x, y)$。

命题 4.2（敏感度加权供给不足比率）。设 y^+、y^0、y^- 的结构兼容性得分分别为 $+\delta$、0、$-\delta$，且基准质量相同。定义 $\lambda_{\text{eff}} := \sum_i \lambda_i \alpha_i / \sum_i \alpha_i$。那么$R(x, y^0) - R(x, y^-) = \lambda_{\text{eff}} \cdot (R(x, y^+) - R(x, y^0))$，其中 最小值$_i \lambda_i \leq \lambda_{\text{eff}} \leq$ 最大值$_i \lambda_i$。

证明。在基线质量相同的情况下：

$$R(x, y^+) - R(x, y^0) = \frac{1}{k} \sum_i \alpha_i \delta \qquad R(x, y^0) - R(x, y^-) = \frac{1}{k} \sum_i \lambda_i \alpha_i \delta$$

λ_{eff} 是 λ_i 的凸组合。相除得到，这是 (x, y^+) 和 (x, y^-) 的凸组合。 □

5 主要结果 I：过度指定检测的可判定性边界

现在我们给出第一个主要结果：表示选择流水线中结构性过度规定检测的精确可判定性边界。

5.1 过度指定谓词

定义 5.1（超出依据的过度规定谓词）。对于一个全可计算流水线$f : \Sigma^* \to \Sigma^*$,

$$B_{\text{bw}}(f) = \begin{cases} 1, & \exists x \in \Sigma^* : \ v_{\text{bw}}(x, f(x)) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

定义 5.2（指标集形式）。固定一个全可计算流水线的有效枚举$\{f_e\}_{e \in \mathbb{N}}$。相应的指标集为

$$\mathcal{B}_{\text{bw}} := \{e \in \mathbb{N} : B_{\text{bw}}(f_e) = 1\}.$$

引理 5.3 (外延性**)**。若 $f, g : \Sigma^* \to \Sigma^*$ 是全可计算的，且对所有$x \in \Sigma^*$都有 $f(x) = g(x)$，那么 $B_{\text{bw}}(f) = B_{\text{bw}}(g)$。等价地，若 $f_e = f_{e'}$ 在外延意义上相等，则 $e \in B_{\text{bw}} \Leftrightarrow e' \in B_{\text{bw}}$。

证明。根据逐点相等，对每个 x，都有 $v_{\text{bw}}(x, f(x)) = v_{\text{bw}}(x, g(x))$。因此，式（2）中的存在性条件对 f 和 g而言相同。 □

引理 5.4 (非平凡性**)**。假设存在 $x_0 \in \Sigma^*$、$s_0 \in \Sigma^*$、$y^+ \in \Sigma^*$，以及一个填充符号 $\# \in \Sigma$，使得对每个 $n \in \mathbb{N}$,

$$s_0 \in S(x_0 \#^n) \setminus W(x_0 \#^n) \quad \textit{and} \quad V(y^+, s_0) = +1.$$

那么 B_{bw} 是非平凡的：$f_1 \equiv y^+$ 给出 $B_{\text{bw}}(f_1) = 1$；$f_2 \equiv \epsilon$（对所有 s，$V(\epsilon, s) = 0$）给出$B_{\text{bw}}(f_2) = 0$。

证明。 对于 $f_1 \equiv y^+$，在 x_0 处求值：

$$v_{\mathrm{bw}}(x_0, f_1(x_0)) = \sum_{s \in S(x_0) \setminus W(x_0)} V(y^+, s) \geq V(y^+, s_0) = 1 > 0,$$

因此 $B_{\mathrm{bw}}(f_1) = 1$。对于 $f_2 \equiv \epsilon$，根据假设，对所有 s， $V(\epsilon, s) = 0$，因此对于每个 x， $v_{\mathrm{bw}}(x, f_2(x)) = 0$，所以 $B_{\mathrm{bw}}(f_2) = 0$。 □

注释 5.5。 引理 5.4 中的填充条件是停机归约的一种技术手段：它提供了一个无界的“时钟”族 $x_0\#^n$，同时保持一个固定的超出保证范围的见证s_0。只要 S 和 W 在这种填充下保持不变，该条件便成立。

5.2 无界定义域上的不可判定性

定理 5.6 （过度指定检测的不可判定性）。 在引理 5.4的条件下，不存在图灵机能够对所有全可计算的 B_{bw}（f）作出判定 f。

证明。 令 $H = \{\langle M, w\rangle : M$在输入 $w\}$时停机。固定引理 5.4中的见证 x_0， s_0， y^+， ϵ， #。对于每一对 $\langle M, w\rangle$，定义一个全可计算函数 $f_{M, w}$： $\Sigma^* \to \Sigma^*$，其定义如下：

$$f_{M,w}(x) = \begin{cases} y^+, & x = x_0\#^n \text{ for some } n \in \mathbb{N} \text{ and } M(w) \text{ halts within } n^2 \text{ steps}, \\ \epsilon, & \text{otherwise.} \end{cases}$$

全性显然成立，因为有界模拟至 n^2 步总会停机。根据 s-m-n 定理，存在一个全可计算映射 r：$\langle M, w\rangle \mapsto e_{M, w}$，使得 $f_{e_{M, w}} = f_{M, w}$。

现在我们验证该归约的正确性。

若 $\langle M, w\rangle \in H$： 假设 $M(w)$ 恰好在 t 步后停机。取 $n \geq \lceil\sqrt{t}\rceil$；那么 $n^2 \geq t$，因此$f_{M, w}(x_0\#^n) = y^+$。所以

$$v_{\mathrm{bw}}(x_0\#^n, f_{M,w}(x_0\#^n)) = \sum_{s \in S(x_0\#^n) \setminus W(x_0\#^n)} V(y^+, s) \geq V(y^+, s_0) = 1 > 0,$$

因此 $B_{\mathrm{bw}}(f_{M, w}) = 1$。

若 $\langle M, w\rangle \notin H$： 没有任何有界模拟会观察到停机，因此 $f_{M, w}(x) = \epsilon$ 对所有 x 都成立。因此$v_{\mathrm{bw}}(x, f_{M, w}(x)) = 0$ 对所有 x 都成立，并且 $B_{\mathrm{bw}}(f_{M, w}) = 0$。

因此，

$$\langle M, w\rangle \in H \iff B_{\mathrm{bw}}(f_{M,w}) = 1 \iff e_{M,w} \in \mathcal{B}_{\mathrm{bw}}.$$

如果 B_{bw} 在所有全可计算流水线上都是可判定的，那么 H 也将是可判定的，这与假设矛盾。 □

备注 5.7。 或者， B_{bw} 是全可计算函数的一个非平凡语义性质，因此根据 Rice 定理 [17]可知其不可判定。显式归约还表明B_{bw} 是 Σ_1^0-困难的。

命题 5.8 (正实例的半可判定性)。 在输入索引表示全可计算流水线这一前提下，语言 B_{bw} 是递归可枚举的。

证明。 固定一个 Σ^*的有效枚举 $(x_j)_{j \in \mathbb{N}}$。给定一个索引 e，按阶段 $t = 0$，1，2，.....交错执行：对每个 $j \leq t$，模拟 $f_e(x_j)$，运行 t 步。每当模拟以输出 y停机时，计算 $v_{\mathrm{bw}}(x_j, y)$。如果该值为正，则接受。

若 $e \in \mathcal{B}_{\mathrm{bw}}$，则存在某个见证 x_j，使得 $v_{\mathrm{bw}}(x_j, f_e(x_j)) > 0$；由于 f_e 是全函数，该计算会在有限时间内停止，因此交错模拟器最终会接受。若 $e \notin \mathcal{B}_{\mathrm{bw}}$，则不存在接受见证，因此图灵机将永远运行下去。因此$\mathcal{B}_{\mathrm{bw}}$是递归可枚举的。 □

5.3 有限域上的可判定性

命题 5.9 (有限域可判定性及指数级代价)。 固定最大输入长度n 并定义 $D_n := \Sigma^{\le n}$, $N_n := |D_n| = \sum_{j=0}^{n} |\Sigma|^j$。令 $T_f(n)$ 表示对 $|x| \le n$计算$f(x)$ 的最坏情况时间，而 $T_{\mathrm{bw}}(n)$ 表示对 $|x| \le n$ 和 $y = f(x)$计算 $v_{\mathrm{bw}}(x, y)$ 的最坏情况时间。则在 D_n 上， $B_{\mathrm{bw}}(f)$ 是可判定的，且其时间为

$$O(N_n \cdot (T_f(n) + T_{\mathrm{bw}}(n))) = O(|\Sigma|^n \cdot (T_f(n) + T_{\mathrm{bw}}(n))).$$

证明。 由于$\mathcal{D}_n$ 是有限的，穷举搜索必定终止。算法：枚举所有 $x \in \mathcal{D}_n$；对每个 x，计算 $y = f(x)$，然后计算 $\delta_x := v_{\mathrm{bw}}(x, y)$。当且仅当存在某个 $\delta_x > 0$ 时，返回 1。

正确性可直接由定义 5.1 在将定义域限制为$\mathcal{D}_n$时得出。运行时间至多为 N_n 次迭代，每次耗时 $O(T_f(n) + T_{\mathrm{bw}}(n))$。 □

注 5.10. 定理 5.6和命题 5.9确立了一条明确的边界：当且仅当输入域有限时，过度指定检测才是可判定的。这在结构上不同于针对动态数据结构的单元探测下界或摊销下界 [12, 4]：那些是有限工作负载上的复杂度障碍，而我们的结果则是选择器族上的可计算性障碍。

6 主要结果 II：保守修复的不动点障碍 ir

一个平凡的修复算子可以通过输出通用基线实现，迫使各处都采用 $v = 0$ ，但这会使选择流水线失去意义。我们将这一约束形式化为保守 约束，并证明过度指定不动点必然存在。

6.1 保守程序变换器

定义 6.1 （可计算修复算子）。 可计算修复算子 是一个全可计算函数Φ： $\mathbb{N} \to \mathbb{N}$ 将流程索引映射到流程索引。

定义 6.2 （保守性）。 Φ 是保守的，当且仅当 $B_{\mathrm{bw}}(f_e) = 0 \Rightarrow \Phi(e) = e$。

注 6.3。 保守性体现了“不要修改已经与证据相符的程序”——这是程序变换类比中的最小修复，与一致查询回答中的最小修复相对应 [15, 16]。

定义 6.4 （一致消除保证）。 修复算子 Φ 具有一致消除，如果

$$\forall e \in \mathbb{N}, \quad B_{\mathrm{bw}}(f_{\Phi(e)}) = 0.$$

6.2 通过克莱尼递归定理得到的过度指定不动点

引理 6.5 (自指构件族)。 固定一个全可计算修复算子 Φ 以及见证 x_0， s_0， y^+， ϵ， $\#$ 来自引理 5.4。那么存在一个全可计算映射 $G\colon \mathbb{N} \to \mathbb{N}$，使得对每个 $e \in \mathbb{N}$，

$$f_{G(e)}(x) = \begin{cases} y^+, & \Phi(e) = e \text{ and } x = x_0\#^n \text{ for some } n \in \mathbb{N}, \\ \epsilon, & \text{otherwise.} \end{cases}$$

证明。 定义可计算谓词

$$P(e, x) := [\Phi(e) = e] \wedge [\exists n \in \mathbb{N} : x = x_0\#^n].$$

由于 Φ 是全可计算的，对于每个 e， $\Phi(e)$ 都是可计算的；通过线性时间扫描，可以判定是否属于 $\{x_0\#^n : n \in \mathbb{N}\}$ 。因此，该程序

$$Q(e, x) := \begin{cases} y^+, & P(e, x) = 1, \\ \epsilon, & P(e, x) = 0, \end{cases}$$

对两个参数都是全可计算的。根据 s-m-n 定理，存在一个全可计算的索引映射 G，使得对于所有 e, x，都有 $f_{G(e)}(x) = Q(e, x)$。 □

定理 6.6 (保守修复下的过度指定不动点)。 假设 B_{bw} 非平凡（引理 5.4）。对于任意全可计算的保守修复算子 Φ，存在 $e^* \in \mathbb{N}$ 使得

$$\Phi(e^*) = e^* \quad and \quad B_{\mathrm{bw}}(f_{e^*}) = 1. \tag{3}$$

证明。 固定 Φ 并根据引理 G 得到 6.5。将克莱尼递归定理（附录 C）应用于 G：存在 e^* 使得

$$f_{e^*} = f_{G(e^*)}.$$

根据引理 6.5，对于每个 x，

$$f_{e^*}(x) = \begin{cases} y^+, & \Phi(e^*) = e^* \text{ and } x = x_0 \# ^n \text{ for some } n, \\ \epsilon, & \text{otherwise.} \end{cases}$$

断言 1： $\Phi(e^*) = e^*$。反设 $\Phi(e^*) \neq e^*$。那么上面的公式给出 $f_{e^*}(x) = \epsilon$ 对所有 x都成立，因此 $B_{\mathrm{bw}}(f_{e^*}) = 0$。根据保守性（定义 6.2），$\Phi(e^*) = e^*$，矛盾。因此断言 1 成立。

断言 2： $B_{\mathrm{bw}}(f_{e^*}) = 1$。由断言 1 和构造件定义，$f_{e^*}(x_0 \# ^n) = y^+$ 对所有$n \in \mathbb{N}$都成立。利用引理 5.4，

$$v_{\mathrm{bw}}(x_0 \# ^n, f_{e^*}(x_0 \# ^n)) = \sum_{s \in S(x_0 \# ^n) \setminus W(x_0 \# ^n)} V(y^+, s) \geq V(y^+, s_0) = 1 > 0.$$

因此 $B_{\mathrm{bw}}(f_{e^*}) = 1$。

结合断言 1 和 2 得出（3）。 □

推论 6.7 (不存在能够完备的保守全修复算子)。 没有全可计算修复算子能够同时满足保守性（定义*6.2*）和一致消除（定义*6.4*）。

证明。 假设 Φ 是全可计算且保守的。根据定理 6.6，存在 e^* 满足$\Phi(e^*) = e^*$ 且 $B_{\mathrm{bw}}(f_{e^*}) = 1$。因此

$$B_{\mathrm{bw}}(f_{\Phi(e^*)}) = B_{\mathrm{bw}}(f_{e^*}) = 1,$$

这违反了一致消除。 □

备注 6.8。 定理 6.6 是最坏情况结果：它构造了一个特定的对抗性选择流程，使其能够规避 任何 保守修复算子。实际的修复启发式方法仍能减少典型工作负载中的过度指定。

备注 6.9。 放弃保守性会使定理不再成立：$\Phi(e) = e_0$（常量基线选择器）只有一个未过度指定的不动点，却会破坏所有表示敏感性。

6.3 算法层面的三方权衡

结合定理 5.6、6.6，以及推论 6.7，任何用于自适应表示选择的修复算法都面临以下选择：

(a) **放弃保守性**。 修改所有流程管线，可能导致已与证据相符的选择器性能下降。

(b) A **放弃完备性**。 接受某些过度指定的流程管线未经修正便通过 .

(c) **限制定义域**。 针对有限实例族进行操作（命题 5.9），并接受指数级代价。

算法选择和基准驱动调优的实用方法能够有效地选择 (b) [2, 10, 11]。我们的结果解释了为何在不放弃保守性或领域通用性的前提下，这是最强的统一策略。

这种权衡在结构上不同于数据结构下界 [12, 5]：这些下界限制了可判定情形中的效率；而我们的障碍则限制了无界选择器族中的可能性。

6.4 与最小修复的联系

保守性（定义 6.2）是基于修复的推理中最小性的选择器类比 [15, 16]。定理6.6指出：在这种准则下，存在这样的选择器：它们自身就是“修复”，却仍然在结构上过度指定。

7 结论

在一种模型下进行研究，在该模型中，结构性偏好会通过基准测试聚合和成对分数拟合传播（命题 3.2–3.4），我们确定了自适应数据结构选择中修复所面临的两种算法障碍：

(i) **可判定性边界**。 在无界定义域上，过度指定检测不可判定；在有限域上，则可判定，但代价为指数级。(ii) **不动点障碍**。 根据克莱尼递归定理，保守修复算子存在过度指定的不动点。

这些障碍在性质上不同于数据结构和算法分析中的经典下界 [12, 5]。那些下界限制的是支持操作的效率。我们的结果限制的是能否在选择器族中统一检测并修复不合理的结构承诺。由此产生的保守性、完备性与域限制三者之间的权衡，为评估图算法、字符串算法和动态数据结构中的任何自适应表示选择流程提供了一个有原则的框架。

参考文献

[1] Thomas H. Cormen、Charles E. Leiserson、Ronald L. Rivest 和 Clifford Stein。《算法导论》。麻省理工学院出版社，第三版，2009年。[2] John R. Rice。算法选择问题。《计算机进展》，15：65–118，1976年。[3] Daniel D. Sleator 和 Robert E. Tarjan。一种用于动态树的数据结构。《计算机与系统科学杂志》，26(3)：362–391，1983年。[4] Daniel D. Sleator 和 Robert E. Tarjan。自调整二叉搜索树。《ACM 汇刊》，32(3)：652–686，1985年。[5] Jacob Holm、Kristian de Lichtenberg 和 Mikkel Thorup。用于连通性、最小生成树、2-边连通性和双连通性的多对数时间确定性全动态算法。《ACM 汇刊》，48(4)：723–760，2001年。[6] Camil Demetrescu 和 Giuseppe F. Italiano。一种动态全点对最短路径的新方法。《ACM 汇刊》，51(6)：968–992，2004年。

[7] Donald E. Knuth、James H. Morris 和 Vaughan R. Pratt。字符串中的快速模式匹配。《SIAM计算杂志》，6(2)：323–350，1977年。[8] Udi Manber 和 Gene Myers。后缀数组：一种在线字符串搜索的新方法。《SIAM计算杂志》，22(5)：935–948，1993年。[9] Esko Ukkonen。后缀树的在线构建。《算法学》，14(3)：249–260，1995年。[10] Catherine C. McGeoch。《实验算法学指南》。剑桥大学出版社，2012年。[11] Tim Kraska、Alex Beutel、Ed H. Chi、Jeffrey Dean 和 Neoklis Polyzotis。学习型索引结构的必要性。载于 *SIGMOD*会议论文集，第489–504页，2018年。[12] Michael L. Fredman 和 Michael E. Saks。动态数据结构的单元探测复杂度。载于 *STOC*会议论文集，第345–354页，1989年。[13] Ralph A. Bradley 和 Milton E. Terry。不完全区组设计的秩分析：I. 成对比较法。《生物计量学》，39(3–4)：324–345，1952年。[14] David R. Hunter。广义Bradley–Terry模型的MM算法。《统计年鉴》，32(1)：384–406，2004年。[15] Marcelo Arenas、Leopoldo E. Bertossi 和 Jan Chomicki。不一致数据库中的一致查询答案。载于*PODS*会议论文集，第68–79页，1999年。[16] Leopoldo E. Bertossi。《数据库修复与一致查询回答》。Morgan & Claypool，2011年。[17] Henry G. Rice。递归可枚举集的类及其判定问题。《美国数学会汇刊》，74(2)：358–366，1953年。[18] Daniel Kahneman 和 Amos Tversky。前景理论：风险下决策的分析。《计量经济学》，47(2)：263–291，1979年。[19] Amos Tversky 和 Daniel Kahneman。前景理论的进展：不确定性的累积表征。《风险与不确定性杂志》，5(4)：297–323，1992年。

多数关联

命题 A.1（严格多数意味着决定性权力）。 在奇数 k下进行成对多数聚合时，任何满足 $C \subseteq [k]$ 且 $|C| > \frac{k}{2}$ 的联盟，对每个有序对 $(y,\ y')$ 都是决定性的。如果 C 中的每位评估者在 $(y,\ y')$ 上都是签名单调的，则命题 *3.2* 可推出继承性。

证明。 固定一个实例 x 和满足 $|Y_x| \geq 3$的候选集 Y_x。令 A 为成对多数规则。取任意满足 $|C| > \frac{k}{2}$ 的联盟 $C \subseteq [k]$。假设 C 中的每位评估者都将 $y \succ_i y'$排在更高位置。那么，严格超过半数的评估者将 y 排在 y'之上。根据多数聚合，无论 C之外评估者的排序如何，聚合结果都满足 $y \succ_A y'$。因此， C对 $(y,\ y')$ 具有决定性权力。

此外，如果 C 中的每个评估者在 (y, y')上都是签名单调的，那么 $v(x, y) > v(x, y')$意味着 $y \succ_i y'$ 对所有 $i \in C$都成立，而决定性给出 $y \succ_A y'$。这正是命题 3.2。 □

B 成对拟合一致性

命题 B.1 (设定正确时 Bradley–Terry 估计量的一致性)。 在规范正确且配对覆盖为正的条件下，对于所有足够大的样本，*Bradley–Terry* 估计量都能恢复真实的分数差 *[13, 14]*。

命题 B.2 （错误设定下的成对拟合）。 在错误设定下，一致的 M-估计量收敛于真实成对概率律在拟合成对比较族上的KL 投影。

C Kleene 不动点定理

引理 C.1 （从 *s-m-n*得到的自应用算子）。 存在一个全可计算函数q： $\mathbb{N} \to \mathbb{N}$，使得对于所有 n、$x \in \mathbb{N}$,

$$\varphi_{q(n)}(x) \simeq \varphi_{\varphi_n(n)}(x),$$

其中 $\simeq$ 表示部分可计算函数的相等性（两边均未定义，或两边均有定义且取值相等）。

证明。 定义一个部分可计算二元函数 H： $\mathbb{N} \times \mathbb{N} \rightharpoonup \mathbb{N}$，使得

$$H(n, x) := \varphi_{\varphi_n(n)}(x).$$

操作上：模拟 $\varphi_n(n)$；如果它停机并输出 m，则模拟 $\varphi_m(x)$ 并返回其输出。如果任一模拟发散，则 $H(n, x)$ 未定义。因此 H 是部分可计算的。

根据 *s-m-n* 定理，存在一个全可计算函数 q，使得

$$\forall n, x, \quad \varphi_{q(n)}(x) \simeq H(n, x) = \varphi_{\varphi_n(n)}(x).$$

这正是所述性质。 □

定理 C.2 (Kleene 递归定理)。 对于每个全可计算的 G： $\mathbb{N} \to \mathbb{N}$，都存在$e^* \in \mathbb{N}$ 满足 $\varphi_{e^*} = \varphi_{G(e^*)}$。

证明。 固定全可计算的 G： $\mathbb{N} \to \mathbb{N}$，并令 q 为引理 C.1中的全可计算映射。定义

$$h(n) := G(q(n)).$$

由于 G 和 q 都是全可计算的，因此 h 是全可计算的。根据部分可计算函数的有效枚举，存在一个索引 $p \in \mathbb{N}$ 使得

$$\forall n \in \mathbb{N}, \quad \varphi_p(n) = h(n) = G(q(n)).$$

现在令

$$e^* := q(p).$$

我们验证 $\varphi_{e^*} = \varphi_{G(e^*)}$。对于任意的 $x \in \mathbb{N}$,

$$\begin{aligned}
\varphi_{e^*}(x) &= \varphi_{q(p)}(x) \\
&\simeq \varphi_{\varphi_p(p)}(x) && \text{(引理 C.1)} \\
&= \varphi_{h(p)}(x) && \text{(} p \text{ 的定义)} \\
&= \varphi_{G(q(p))}(x) && \text{(} h \text{ 的定义)} \\
&= \varphi_{G(e^*)}(x).
\end{aligned}$$

由于 x 是任意的，$\varphi_{e^*} = \varphi_{G(e^*)}$。 □

D 有限域检测算法

算法 1 有限域过度指定检测

过程 判定过度指定(f, n)
    对所有 $x \in \Sigma^{\leq n}$ **执行**
        如果 $v_{\mathrm{bw}}(x, f(x)) > 0$ **则返回** 1
        结束如果
    结束循环返回 0
结束过程