arXiv:2304.04318v1 [cs.DC] 9 Apr 2023

关于仅扩展有向偏序集及派生的拜占庭容错复制数据类型 (扩展版)

FlorianJacob
florian.jacob@kit.edu 卡尔斯
鲁厄理工学院 卡尔斯鲁厄，德国

Hannes
H
artensteinhannes.hartenstein@kit.edu 卡
尔斯鲁厄理工学院 卡尔斯鲁厄，德国

摘要

我们揭示了仅扩展有向偏序集（EDP）结构，将近期讨论的基于有向无环图（DAG）的拜占庭容错无冲突复制数据类型（CRDT）统一起来。我们还展示了如何从 EDP 表述中推导出键值映射模型，并展望一种基于 EDP 的系统性访问控制无冲突复制数据类型，作为对 Matrix 消息系统中所使用无冲突复制数据类型的形式化表达。

CCS 概念：• **安全与隐私** → **分布式系统安全**；访问控制；• **软件及其工程** →**一致性**。

关键词：无冲突副本数据类型，强最终一致性，拜占庭故障模型，Matrix 事件图

1 引言

最近，我们注意到 [7] 一个用于去中心化消息传递的系统（Matrix，[15]）的无冲突复制数据类型在存在拜占庭节点的环境中也保留了其CRDT性质，即在节点任意偏离预期协议行为的环境中。Klepp- mann [8] 能够证明，任意容忍崩溃故障的无冲突复制数据类型都可以转换为能够容忍任意数量拜占庭节点的无冲突复制数据类型。这些针对拜占庭情形的方案有一个共同点，即事件由一个类图共享对象进行管理：事件在各个副本（网络分区期间可用）上独立生成，但会与该副本已知的先前事件相关联。

共享对象基本呈现如下动态行为。所有副本均以相同的创世事件作为初始状态。要追加新事件，副本会将该事件附加到其当前已知的所有“先于”该事件发生且没有“后代”的事件上，也就是“最新”的事件，而无需与其他副本协调。随后，副本进行同步，即交换更新，以再次达到一致状态。当副本追加并发发生的事件时，就会产生分支，从而形成树状结构。当副本获知存在分支后，会在下一次添加事件时将这些分支重新合并，从而消除分支。事件既包含有效载荷，也包含先前事件的哈希值，以确保完整性和强最终一致性。

面对拜占庭歧义行为 [8, 5]。此类对象随时间演变的一个示例见图1。

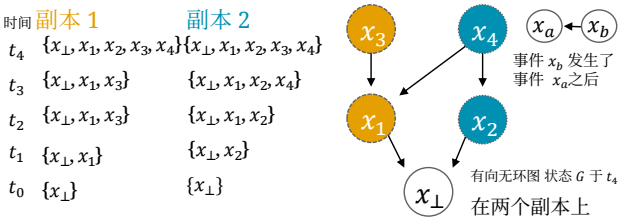

图 1. 某一时刻 G 两个副本所见的示例图状态 t_4，其中包含来自两个副本的事件以及预共享的创世事件 $x_\perp$。首先，两个副本并发发送事件 x_1 和 x_2。副本 2 发送事件x_4时，已从副本 1 获知 x_1，因此将 x_1 和 x_2 都作为发生后父节点。副本 1 尚未获知 x_2，因此仅将 x_3 追加到 x_1。

虽然在先前的工作中，相应对象被描述为有向无环图（DAG），但在这篇正在进行中的论文中，我们通过有向偏序集 （有向偏序集）提出了一种更为基础且统一的无冲突复制数据类型形式化方法，称为仅扩展有向偏序集（EDP），其展现出以下优势：

- 集合论形式化抓住了拜占庭容错有向无环图无冲突复制数据类型的本质，揭示了*a)* 一种不需要密码学元素的基于状态的 EDP，以及*b)* 一种作为效率优化的基于哈希的基于操作的构造。与基于有向无环图的变体相比，EDP 表述采用标准数学概念，并契合主要以集合论表述的 CRDT 理论 [12]。鉴于集合论，尤其是格论，与布尔代数 [11], 之间关系密切，我们认为 EDP 可能有助于未来的形式化验证。

- 基于通用的 EDP 形式化，还可以通过组合派生和／或构造其他类型，从而更容易证明它们在拜占庭环境中具有 CRDT性质。我们以派生映射类型为例，并展望一种用于系统性访问控制的组合式无冲突复制数据类型，它是对 Matrix 中所用 CRDT 的形式化。

相关工作在拜占庭环境中使用集合而非有向无环图，但并未将其视为 CRDT 并加以推广 [3]。在接下来的章节中，我们将把仅扩展有向偏序集定义为 CRDT，并说明为何它们构成了上述拜占庭容错 CRDT 类别的坚实基础。

2 仅扩展有向偏序集（EDPs）

2.1 基础知识与符号约定

EDP 复制数据类型的状态空间基于关系结构，接下来我们将介绍相关的数学基础和符号约定。

一个关系结构 $S = (X, R)$ 是一个由底集 X 和关系 $R \subseteq X \times X$ 组成的元组。为便于记号，我们有时写作 $S.X$ 和 $S.R$，分别表示 $S = (X, R)$ 的集合 X 和关系 R。偏序集，也称为偏序集，是一种关系结构，它具有自反性（$\forall a \in X\colon (a, a) \in R$）、传递性（$\forall a, b, c \in X\colon (c, b) \in R \wedge (b, a) \in R \Rightarrow (c, a) \in R$）以及反对称性（$\forall a, b \in X\colon (a, b) \in R \wedge (b, a) \in R \implies a = b$）。尽管在数学中通常使用 $\leq$-形式的 R，但为了与本文其余部分保持一致性，我们在这些定义中使用 $\geq$-形式的 R。如果 S 是偏序集，那么 R 称为偏序关系，或偏序。如果 S 还具有强连通性（$\forall a, b \in X\colon (a, b) \in R \vee (b, a) \in R$），则 S 称为全序集。

自反且传递的关系结构 S 称为向下有向集，如果任意两个元素都有一个下界，即 $\forall a, b \in X\colon \exists x_{lb} \in X \text{ s.t. } (a, x_{lb}) \in R \wedge (b, x_{lb}) \in R$。有限的向下有向偏序集朝着其唯一的最小元 $x_\perp$ 有向，该最小元也记作 $S^\perp$，即 $x_\perp \in X \text{ and } \forall x \in X\colon (x, x_\perp) \in R$。反之，有限的向上有向偏序集朝着其最大元 $S^\top = x_\top \in X$ 有向，即 $\forall x \in X\colon (x_\top, x) \in R$。同时向下有向和向上有向的有限偏序集称为由 $x_\perp$ 和 $x_\top$ 界定的偏序集。S 的极大元素集为 $\max(S) = \{m \in X | \forall x \in X\colon (x, m) \in R \Rightarrow (m, x) \in R\}$。反之，$S$ 的极小元素集为 $\min(S) = \{m \in X | \forall x \in X\colon (m, x) \in R \Rightarrow (x, m) \in R\}$。

如果关系结构 $S' = (X', R')$ 满足 $X \subseteq X'$ 和 $R = R'|_X$，其中限制 $R|_A$ 按通常方式定义为 $R|_A = R \cap (A \times A)$，则称其为另一个关系结构 S 的扩张。如果向下有向偏序集 S 还满足 $S'^\perp = S^\perp$，则称 S' 为其向上扩展。元素 $y \in S.X$ 的向下闭包 $y^{\downarrow S}$ 定义为 $y^{\downarrow S} = \{c \in X | (y, c) \in R\}$。子集 $Y \subseteq X$ 的向下闭包 $Y^{\downarrow S}$ 由单元素向下闭包推广定义为 $Y^{\downarrow S} = \bigcup_{y \in Y} y^{\downarrow S}$。$X$ 中元素和子集的向上闭包相应地定义为 $y^{\uparrow S} = \{c \in X | (c, y) \in R\}$。元素 $y \in X$ 的极大下界集合，是其向下闭包中除去 y 自身后的极大元素集合，即 $\mathrm{mlb}(y) = \max(y^{\downarrow S} \setminus \{y\})$。

2.2 规范

为了构建一种能够容忍任意数量拜占庭节点的仅追加 CRDT（如引言中所述），
关键思想是将一个事件 x 替换为其向下闭包，即添加与先前事件的所有关系。由此，拜占庭节点再也无法创建不一致的关系了——它只能创建垃圾添加。因此，EDP RDT 将被定义为“高阶”有向偏序集，

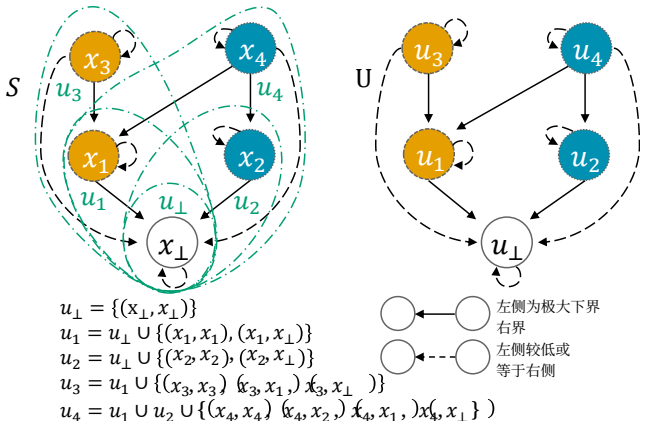

图2. 一个BDP状态 S 以及基于图1的等价EDP状态 U。向上扩展 $u_i \in U$ 是其在 S 中所包含关系的集合，用于表示 $x_i \in S$ 的形成历史。

即，将表示由有向偏序集构成的有向偏序集。由此得到的 EDP RDT 随后可以很容易地证明表示一种基于状态的 CRDT，并且可以通过使用密码哈希函数作为顺序保证机制来构造其基于操作的版本。为清晰起见，我们将用作构造EDPRDT的“基础层”的有向偏序集称为基本有向偏序集（拜占庭容错有向偏序集）。

要确定一个BDP，需选定一个有效元素的全集 $\mathbb{X}$ 和一个有效关系的全集 $\mathbb{R} \subseteq \mathrm{P}(\mathbb{X}^2)$。BDP状态是有限的、向下有向的偏序集 $S = (X, R)$、$X \subseteq \mathbb{X}$、$R \in \mathbb{R}$，它们具有共同的底元素 $S^\perp$（该元素将被解释为所有正确副本的初始状态）。例如，可以将全集 $\mathbb{X}$ 视为应用层消息，将全集 $\mathbb{R} \subseteq \mathrm{P}(\mathbb{X}^2)$ 视为因果序，并将某个特定的引导应用层消息视为底元素。

要定义仅扩展操作，我们只希望允许对BDP状态 S 进行单元素向上扩展。我们要求扩展满足最终性这一有效性标准：向BDP添加新元素以及添加该新元素的关系，必须是一次单一的原子向上扩展，即之后不能再更改。所要求的最终性属性可表述如下：当元素 x 是为获得状态 S 而添加的唯一元素时，对于 S 的所有向上扩展 S'，都必须满足[1]：$x^{\downarrow S'} = x^{\downarrow S}$ 和 $R'|_{x^{\downarrow S'}} = R|_{x^{\downarrow S}}$。

为了提供这种最终性概念，显而易见（但效率低下）的方法是，将一个针对元素 y with $x \geq y$ 的单元素向上扩展，与所有单元素向上扩展绑定起来；该扩展将 $S = (X, R)$ 扩展一个元素 $x \in \mathbb{X}$。这样一来，仅扩展操作的历史便得以固定，并作为这个新的单元素向上扩展的身份形成信息。要表示拜占庭容错有向偏序集元素的“形成历史”集合，就必须转向集合的集合。

[1]相比之下，一个元素的向上闭包及其关系可能永远不会最终确定：$x^{\uparrow S'} \supseteq x^{\uparrow S}$ 以及 $R'|_{x^{\uparrow S'}} \supseteq R|_{x^{\uparrow S}}$。一个新的上界元素始终可能正在传输中，或者据称曾处于传输中。

偏序集。因此，对于 EDP 的定义，我们“上移一级”，并按照以下两个步骤，将这些拜占庭容错有向偏序集映射为 EDP 的元素。图2展示了基于图1中的状态，将拜占庭容错有向偏序集映射为 EDP 状态的示意图。

步骤 1. 设有向偏序集 $S' = (X' R')$ 是 BDP 中有向偏序集$S = (X, R)$ 的单元素 $x \in \mathbb{X}$ 向上扩展。向下闭包 $x^{\downarrow S'}$（其 R'限制在 $x^{\downarrow S'}$ 上）是一个由 $x_\perp$ 和 x 界定的子偏序集 $(x^{\downarrow S'}, R'|_{x^{\downarrow S'}}) \subseteq S'$。令 $u_x := R'|_{x^{\downarrow S'}}$ 表示此类 BDP 向上扩展与 x 的关系，并令 $X(u_x) := \{y \in \mathbb{X} | (y, y) \in u_x\} = x^{\downarrow S'}$ 表示 u_x 中自反对的集合。我们可以从 u_x 得到 S 的向上扩展 S'，其中 x 为 $S' = (X \cup X(u_x)\ R \cup u_x)$。因此，作为简写，我们也将 u_x称为向上扩展。向上扩展 $u_x \in \mathbb{R}$是有效的，当且仅当 $(X(u_x)\ u_x)$ 构成由 $x_\perp$ 和 x界定的 S' 的 BDP 子偏序集。虽然我们关注其关系，但有界子偏序集是向上扩展的另一种简写形式。

步骤 2。现在我们“上移一个级别”，将向上扩展 u_x 作为元素，并通过子集关系构造这些向上扩展的偏序集，以此定义 EDP。EDP 状态$U \in \mathrm{P}(\mathbb{R})$ 是单元素向上扩展$U = \{u_\perp, u_1, \ldots\}$的集合，即 BDP 状态 S的形成历史。初始扩展有向偏序集状态为 $U = \{u_\perp = \{(x_\perp, x_\perp)\}\}$，也就是空集与底元素的向上扩展。EDP 状态 $U \in \mathrm{P}(\mathbb{R})$ 有效，当且仅当 $(U, \supseteq|_U)$ 是一个 $u_\perp$-有向偏序集，且$\forall u \in U: \mathrm{mlb}(u) \subseteq U \wedge |X(u)| = |\{X(\bigcup \mathrm{mlb}(u))| + 1$；也就是说， U 中的每个向上扩展，其极大下界也都在 U中，并且该扩展恰好为 BDP 状态增加一个元素，而该元素不存在于其任何一个极大下界中。图3展示了应用向上扩展的示例，该扩展源自图1中的图。EDP 状态U可以通过 $S(U) = (X(\bigcup U) \bigcup U)$ 转换回底层 BDP 上的有向偏序集 $S(U)$。上述 EDP 定义产生了一个基于状态的CRDT，其状态空间为 $\mathrm{P}(\mathbb{R})$，并以集合并 $U_1 \cup U_2$ 作为合并运算，如定理 1 所示。

定理 1. 假设所有正确副本构成一个连通分量，且该分量内最终通信，则即使存在任意数量的拜占庭副本，基于状态的 *EDP* 仍是一种无冲突复制数据类型。

证明概要。对于状态空间 $\mathrm{P}(\mathbb{R})$，集合并 $U_1 \cup U_2$ 是 U_1的最小上界，$U_2 \in \mathrm{P}(\mathbb{R})$，由此$\mathrm{P}(\mathbb{R})$ 与集合并构成一个并半格。通过定期的状态传播-
送达后，最终交付即可得到保证。结合纯数学状态合并所保证的终止性，基于状态的 EDP 是一种无冲突复制数据类型 [14]。由于更新仅由半格元素构成，因此没有可伪造的元数据，拜占庭副本受到限制：无效的拜占庭更新会因不属于半格而被拒绝；而且，由于完整的形成历史保护着有效更新的完整性，任何拜占庭方试图篡改历史、进行矛盾陈述或以其他方式损害一致性的行为，都等同于多个有效更新，只是这些更新未能成功发送给所有副本 [5]。□

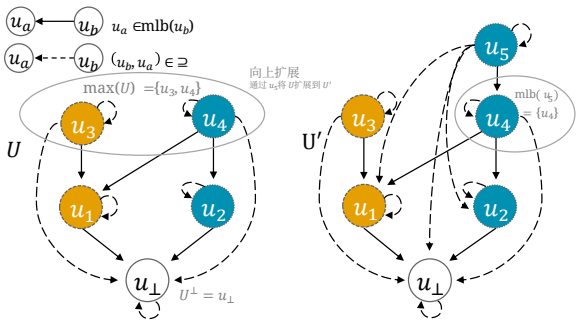

图 3. EDP 的副本状态 U，以及通过 u_5对 U进行向上扩展后得到的状态U'。

尽管将其建模为有向偏序集可能会引入一些形式化内容，但上述对EDP 无冲突复制数据类型的表述揭示了拜占庭容错有向无环图无冲突复制数据类型的简单集合论基础，即基于状态的结构。基于这一集合论视角，无需使用哈希值之类的密码学元素即可轻松证明定理 1，无需借助哈希值等密码学元素来实现拜占庭容错，因为集合元素之间的完整关系足以保证完整性。然而，从实际角度来看，基于状态的EDP效率极低：副本会持续广播其完整状态 U，而该状态只会不断增大。为了达到带有 EDP的有向无环图的效率，我们现在从基于状态的形式化逐步构建一种高效的基于操作的 EDP无冲突复制数据类型形式化，该形式化利用加密哈希值来保护完整性。该构造遵循 [7,8], 其目的在于阐明基于操作的形式化与基于状态的形式化之间的关系。我们需要进行两项优化，下面先给出直观解释，随后再进行形式化。具有韧性的操作广播将在第 2.3节中介绍，拜占庭强最终一致性的证明概要将在第2.4节中介绍。第一项优化是，向上扩展u_y表示由 $x_\perp$ 和 y界定的 S'的一个子偏序集，因此，仅发送该子偏序集的关系而不是状态 U' 就足够了。对于 基于操作的形式化，我们随后压缩该子集，将最坏情况下的更新大小从 U'的“深度”缩减为“宽度”：我们定义一个操作 o，该操作由 y 以及 u_y 在 U'中的经哈希的极大下界集合组成。当$\mathrm{mlb}(u_y) \subseteq U$时，可以基于操作 o 和现有状态 U重建 u_y。

为将 y 链接到其极大下界，使用了一个原像抗碰撞哈希函数 $h(a, b)$，该函数返回其链接参数的哈希值。我们递归地定义极大下界集合 U'_{mlb} 的哈希值集合为$H(U'_{\mathrm{mlb}}) := \{h(\max(X(u))\ H(\mathrm{mlb}(u)))\ | u \in U'_{\mathrm{mlb}}\}$。为了定义 h第二个参数中哈希值集合的哈希值，可以将其按字典序排序后得到的线性化结果连接起来。随后定义操作 $o := (y, H_{\mathrm{mlb}})$，将向上扩展 u 压缩为 $o(u) := (\max(X(u))\ H(\mathrm{mlb}(u)))$。经过哈希处理的极大下界

集合通过递归方式保护有向子偏序集直至底元素 $u_\perp$ 的完整性，使其免受拜占庭节点的攻击。要根据当前状态 U的知识，从操作 o重构向上扩展 u，我们需要 $o.y$所有前驱 $P := (\bigcup U' \text{ mlb } \subsetneq U | H(U' \text{ mlb}) = o.H_{mlb})$的关系集合。接着，向上扩展是 y的自反关系、 y 与其前驱元素 $X(P)$之间的关系，以及其传递关系（即其前驱关系 P）的并集。我们将此重构形式化为$u(y,P) := \{y\}^2 \cup \{y\} \times X(P) \cup P$。一旦所有极大下界都属于副本状态，操作 o便可在本地执行，即 $o.H_{\text{mlb}} \subseteq H(U)$。通过 $U' = U \cup u(o.y, P)$，将操作 o应用于状态 U；该操作继承了集合并的交换性，从而构成状态空间为 $\mathrm{P}(\mathbb{R})$ 的基于操作的无冲突复制数据类型。现在，更新大小受向下闭包中任一元素的极大下界集合的最大大小所限，而不再受整体状态大小限制。

基于操作的 EDP 如算法1所示。遇到未满足的断言时，算法会停止处理该函数并返回错误。遇到未满足的 await 时，算法会中断，以等待其未来可能得到满足，同时不会阻塞后续函数调用。扩张函数用于使用新的向上扩展 $u \in \mathbb{R}$来扩展当前状态U。无副作用的生成函数生成更新操作 o，并将其广播给所有副本，包括自身。接收到的广播由效果函数处理；在应用该操作之前，效果函数会等待所有极大下界都成为当前状态的一部分。第 2.4 节概述了拜占庭容错，其要求使用下一节所述的弱容错广播。

2.3 基于操作的更新的弹性广播

基于操作的 EDP 表述不需要崩溃容错／拜占庭容错可靠广播所提供的强保证。正如 [8], 中所解释的，对于此类拜占庭女巫攻击抗性 CRDT，只要正确副本构成一个连通分量，更新的最终交付就已足够。由于共享底元素且关系具有方向性，广播极大元素的集合就足够了，因为可以迭代地向其他副本请求缺失元素，并通过哈希链验证其完整性。 [8] 中介绍了一种遵循这一思路的优化广播方法。

定期传播副本当前状态的极大向上扩展集合 $\hat{}U = \max(U)$（或其节省空间的操作等价形式）时，该广播本身就是一种基于状态的CRDT：状态空间构成一个 join 半格，$\hat{}U_1$ 与$\hat{}U_2$ 的 join 为 $\max(\hat{}U_1 \cup \hat{}U_2)$。最坏情况下，更新大小接近 EDP 的大小；但如果所有副本都正确，更新大小会迅速收敛到约等于参与副本的数量 [7]。拜占庭副本可以发送大量伪造的扩展，但由于由此产生的基于状态的广播是一种具有拜占庭 Sybil 抵抗能力的CRDT [5], ，它们只能攻击性能，无法破坏正确性。

算法1基于操作的 EDP 无冲突复制数据类型，用于拜占庭容错有向偏序集 ($X \subseteq \mathbb{X},\ R \in \mathbb{R}$)

要求： 有效拜占庭容错有向偏序集元素的全集 $\mathbb{X}$
要求： 有效的拜占庭容错有向偏序集关系全集 $\mathbb{R} \subseteq \mathrm{P}(\mathbb{X}^2)$

状态 向上扩展集合 $U \in \mathrm{P}(\mathbb{R})$
初始 $U \leftarrow \{u_\perp = \{(x_\perp, x_\perp)\}\}$
查询 机器人 ()： $u_\perp = U^\perp \in U$
查询 最大值 ()： $U_{\max} =$ 最大值$(U) \subseteq U$
查询 $mlb(u \in U)$： $U_{\text{mlb}} = \text{mlb}(u) \subseteq U$
查询 $S()$：有向偏序集 $S = (X \subseteq \mathbb{X},\ R \in \mathbb{R})$
$S \leftarrow (X(\bigcup U) \bigcup U)$

更新扩展 $(u \in \mathbb{R})$
生成$(u \in \mathbb{R})$
断言 $\emptyset \neq u \notin U$
断言 $\emptyset \neq \text{mlb}(u) \subseteq U$
$o \leftarrow (\max(X(u))\ H(\text{mlb}(u)))$
效果 $(o = (y \in \mathbb{X}, H_{\text{mlb}}))$
等待 $H_{\text{mlb}} \subseteq H(U)$ ▷等待更新的效果 $\subsetneq u$
断言 $H_{\text{mlb}} \neq \emptyset$
$P \leftarrow \bigcup (U'_{\text{mlb}} \subseteq U | H(U' \text{ mlb}) = H_{mlb})$
$U \leftarrow U \cup (\{y\}^2 \cup \{y\} \times X(P) \cup P)$

2.4 基于操作的拜占庭强最终一致性

为证明基于操作的 EDP RDT 具备拜占庭强最终一致性，我们采用 [9] 中定义的强最终一致性（SEC）概念，该概念包含三个性质：a) **自更新**，即当且仅当正确副本生成一项更新时，它才会将该更新应用于自身状态；b) **最终更新**，即对于正确副本应用的任何更新，所有正确副本最终都会应用该更新；以及 c) **强收敛**，即任何两个已应用相同更新集合的正确副本都处于相同状态。

引理 1. 设 $U \in \mathcal{P}(\mathbb{R})$ 是对应于 $S = (X \subseteq \mathbb{X}, R \in \mathbb{R})$的有向偏序集，且 U'（分别）为应用向上扩展u的更新 $o = (y, H_{\text{mlb}})$ 后得到的状态。那么， U'（分别）是 U（分别）的偏序扩张，且朝向同一元素 $U^\perp$（分别）。

证明概要。 无效的向上扩展通过断言予以丢弃，然后得到 $U' = U$。有效的向上扩展只会向 S 添加单个元素 y。由于 P 是 $u_\perp$-有向子偏序集的并，且 $\{y\} \times X(P) \subseteq u$， $U'^\perp = U^\perp$，此外还有 $S'^\perp = S^\perp$，也就是说， $\perp$-有向性得以保持。基于同样的论证，偏序性质对于 U' 和 S' 也仍然成立。□

定理 2。 在所有正确副本构成一个连通分量且该分量内存在最终通信的假设下，即使面对任意数量的拜占庭副本，基于操作的 *EDP* 仍是一种无冲突复制数据类型。

证明概要。 **自更新：** 通过如第 2.3 节所述的广播，广播副本无需等待任何确认即可接收更新。生成函数创建一个更新，使得效果函数中的等待前置条-件立即得到满足，这意味着效果函数会直接更新副本自身的状态。拜占庭节点没有攻击途径干扰此过程。**最终更新：**通过如第2.3节所述的广播，只要一个正确副本接收到更新，最终所有正确副本都将接收到该更新。由于正确副本的连通分量之间存在最终通信，拜占庭节点没有攻击途径干扰正确更新在正确副本之间的最终交付。借助哈希链式极大下界，副本可以在每次新更新上验证向下闭包中一直到底元素的有向子偏序集的完整性和完备性，而不会受到拜占庭节点的干扰。只要一个正确副本能够满足应用更新的等待前置条件，即已接收到要添加的元- 素所需的向下闭包，它最终就会与所有正确副本共享该向下闭包；届时，这些副本的交付前置条件也最终会得到满足，因此它们同样可以继续执行效果函数。**强收敛：**由于引理1以及集合并的交换性，所有有效更新都是可交换的。由于效果函数的断言，只有有效更新会被应用。由于一个操作由其有效载荷以及其极大下界的哈希值共同组成，元素向下闭包的有向偏序集的完整性通过哈希链得到完整性保护。因此，拜占庭节点试图通过双重陈述，利用两个操作 $(y_a, H^a{}_{\text{mlb}})$ 和$(y_b, H^b{}_{\text{mlb}})$ 攻击一致性，其中 $y_a = y_b$或 $H^a{}_{\text{mlb}} = H^b{}_{\text{mlb}}$ 不成功，因为所有正确副本都不会将任一更新视为已接收而拒绝它们，而是将其视为独立的更新。由于有效性检查在所有副本上都是确定且相同的，拜占庭副本无法使某个更新仅在正确副本的真子集上被应用。□

在给定正确副本之间能够最终通信这一假设下，无论是基于操作的 EDP 还是基于状态的 EDP，就其本身而言，都不需要更新真实性即可提供强最终一致性（SEC）。由于它们的更新函数不依赖于创建更新的副本身份，因此伪造但有效的更新只不过是另一个有效更新。然而，一旦应用要求集合元素包含发送副本的标识符（例如用于访问控制），就需要使用数字签名；我们将在第 3.2 节讨论这一点。

2.5 因果仅扩展有向偏序集（CEDP）

我们将带有时间事件的仅扩展有向偏序集子类型表示为集合项及其因果关系，即其发生于……之后或同时的关系，记作 Causal 仅扩展有向偏序集（CEDP）。作为 EDP 的子类型，定理 2 同样适用。哈希链可以表示任意向上仅扩展偏序，但其固有特性是能够以拜占庭容错的方式证明，图像先于原像，并由此证明先于关系。

CEDP 是因果事件有向无环图方法（例如 Matrix 事件图（MEG） [15]）的基于集合的版本，也就是其自反传递闭包。 [7] 中给出了 MEG 属于拜占庭容错无冲突复制数据类型的具体证明。 [9], 中则采用事件有向无环图方法作为传输层，使任意无冲突复制数据类型具备拜占庭容错能力。关于拜占庭容错和广播要求的论证同样适用于 CEDP。

尽管离散逻辑时间的因果集方法在量子物理学中广为人知 [1],，但在分布式系统中，由 Lamport 定义的先发生关系 [10] 很常见。发生后相等关系是先发生关系的自反闭包的逆关系，因此这两个关系可以互换。不过，发生后相等关系表示哈希链的方向以及一种基于排序的相等概念。在物理学中，事件被定义为空间—时间中的一个点。操作$o = (x,\ H_{\text{mlb}})$可以理解为“事件 x 发生在离散逻辑时间坐标 H_{mlb} 以及（隐式）副本标识符对应的离散空间坐标处”。发生后序是因果序的超集，即事件彼此存在因果依赖时所遵循的顺序。应用程序要么提供新事件及其在因果序中的极大下界，要么通过 $\max(U)$ 使用发生后序作为极大下界；后者同样涵盖实际的因果序。

3 由仅扩展有向偏序集派生的复制数据类型

3.1 基于 EDP 的映射（EPM）

3.1.1 EPM 规范。基于 EDP 复制数据类型，我们推导出一种称为 EPM 的映射复制数据类型。基本上，全集 $\mathbb{X}$ 现在表示键值对，而与之前一样，关系 $\mathbb{R}$ 定义了更新操作的排序方式。对于基于 $\mathbb{R}$的更新操作排序，EPM 采用“最大元素优先”的语义。

形式化地，我们将 $\mathbb{M} \subseteq \mathbb{X}$ 定义为具有 $(k \mapsto v)$ 形式的有效键值对集合，并将映射 $M \subseteq \mathbb{M}$ 定义为所有有效键值对的“单射”子集，即每个给定的键只有一个唯一关联值。我们将方括号运算符定义为通过映射 M 查询键的运算符，形式为 $(k \mapsto v) \in M \Leftrightarrow M[k] = v$；并将映射更新运算符 $\uplus$ 定义为用于单元素向上扩展 u 的运算符，同时保持 M的单射性：

$$\forall u \in \mathbb{R}, x = \max(X(u)) :$$

$$M \uplus u := \begin{cases} M \setminus \{k \mapsto _\} \cup \{k \mapsto v\} & \text{if } x = (k \mapsto v) \in \mathbb{M} \\ M & \text{if } x \in \mathbb{X} \setminus \mathbb{M} \end{cases}$$

EPM 复制数据类型的函数见算法 2。该算法的核心是 线性化函数，它将一个由 $\subseteq$ 偏序的集合 $T \subseteq U$ 线性化为序列 T_n。我们定义关系 $R_\parallel$，它包含所有成对的向上扩展 $(u_1,\ u_2) \in T^2$，这些扩展彼此“并行”，即无法通过 $\subseteq$进行比较；并且它们在如下意义上是极小的：不存在比 u_1 更小且无法与 u_2比较的元素；反之，对于 u_2也同样如此。使用一个原像抗碰撞密码哈希函数 h，我们定义严格线性序关系 $R_h = \{(u_1,\ u_2) \in \mathbb{R}^2 | h(u_1) < h(u_2)\}$。关系 $R_\parallel \cap R_h$ 由此包含了使偏序 $\subseteq$ 获得线性序 $R_l = (\subseteq \cup (R_\parallel \cap R_h))^+$所需的破平局规则。该哈希函数能够消解平局，同时避免拜占庭节点操纵结果。

请注意，Matrix 还提供基于 CEDP 的映射（类似于第 3.1 节），用于为副本和复制对象分配其他属性，并利用这些映射实施访问控制；我们将在第 3.2 节讨论这一点。

3.1.2 拜占庭最终一致性验证。

定理 3。 假设所有正确副本构成一个连通分量，且该分量内存在最终通信，那么即使面对任意数量的拜占庭副本，*EPM* 也是一种基于操作的无冲突复制数据类型。

证明概要。 通过将其归约为底层基于操作的 EDP 的拜占庭强最终一致性。在正确副本中，键值对通过集合 $\max(U)$ 作为 $(k \mapsto v)$ 的极大下界被置于上下文中，这满足等待前置条件并保持了**自更新性质**。广播和更新应用均未发生变化，而线性化函数不与其他副本交互，而是将所有

算法 2 基于操作的 EPM 复制数据类型

状态 向上扩展集合 $U \in \mathrm{P}(\mathbb{R})$
初始 $U \leftarrow \{u_\perp = \{(x_\perp, x_\perp)\}\}$

更新 $put((k \mapsto v) \in \mathbb{M})$
  $y \leftarrow (k \mapsto v)$
  扩展$(u(y, \bigcup$最大值$(U)))$ ▷ 算法 1 的函数
函数 线性化 $(T \subseteq U)\colon T_n \in \mathbb{R}^n$
  $R_\parallel \leftarrow \{(u_1,\ u_2) \in T^2 | u_1 \not\subseteq u_2 \wedge u_2 \not\subseteq u_1$
    $\wedge \forall u \subsetneq u_1 \colon\ u \subseteq u_2 \wedge \forall u \subsetneq u_2 \colon\ u \subsetneq u_1\}$
  $R_l \leftarrow\ (\subseteq \cup (R_\parallel \cap R_h))^+$ ▷ 偏序 $R_\parallel$ 经由 R_h,
    ▷ $^+$ 表示自反传递闭包
  $T_n \leftarrow$ 枚举 $(T,\ R_l)$ ▷ 设置 $T \to$ 序列 T_n
查询 获取 $(T \subseteq U)\colon M \in \mathbb{M}$
  $T_n \leftarrow$ 线性化$(T^{\downarrow U})$
  $M \leftarrow T_0 \uplus T_1 \uplus \ldots \uplus T_n$

在 EDP 中应用的更新纳入考虑，从而保持**最终交付**。get 函数和线性化函数以及映射更新运算符 $\uplus$ 均具有确定性，并忽略无效更新，因此，给定相同的向下有向偏序集 $(U,\ \supseteq)$，即 EDP 的相同状态，它们会返回相同的映射 M，从而保持**强收敛**。□

3.2 系统性访问控制展望

访问控制通常由一个中心化实体以强一致的方式实施。在分布式、弱一致性的环境中，我们必须接受这样一个事实：时间只构成偏序，事件和管理变更会并发发生，并且不存在关于事件总顺序或哪些策略“现在”生效的共识 [16]。

此前关于 CRDT 访问控制的研究主要聚焦于封闭群组中的类似文件系统、可通过密码学强制执行的访问控制 [13, 18]，而我们的展望则受到 Matrix [15] 及类似系统 [2]中的细粒度授权和管理权限的启发。

其理念是提供系统性访问控制，即在类型中存储策略所需的属性以及策略本身，从而将访问决策集成到无冲突复制数据类型函数中，并实现去中心化强制执行。第 2.2 节中纯粹的 EDP 形式化并不要求更新真实性，但去中心化强制执行要求一个副本能够向另一个副本证明某次更新是由第三个副本负责的，因此需要通过数字签名实现可转移认证。这种系统性访问控制可以通过组合一个 CEDP（借助先于关系获得逻辑时间概念）和多个 EPM 来构建。

主要挑战在于安全地处理并发的管理变更。为了处理并发且相互冲突的变更，关键理念是：只要某个并发更新在其向下闭包范围内获得授权，就绝不会被拒绝；只不过在线性化时，如果另一项变更胜出，它可能会被忽略。附录A定义并讨论了这种组合数据类型的原型模型，并附有一份无冲突复制数据类型性质的证明概要。

4 Conclusion& FutureWork

基于围绕偏序集的数学形式化，我们将仅扩展有向偏序集（EDP）提出为拜占庭环境中围绕 DAG CRDT 的各种思想的统一推广。基于状态的形式化展示了这些 CRDT 的本质，而基于操作的形式化则明确体现了针对效率的优化。基于 EDP，我们可以推导出模块化的构建模块，并轻松组合出复杂的拜占庭容错CRDT，这一点已通过映射数据类型得到示例，并由系统性去中心化访问控制这一用例加以说明。在组合过程中，

这些构建模块或许是首批能够对面向公共部门的、基于 CRDT 的 Matrix 消息系统进行包含访问控制的形式化描述的构建模块 [4]。

这项进行中的工作为形式化验证拜占庭环境中基于 EDP 的系统设计及其相应实现（如 Matrix ）的强最终一致性迈出了一步，同时也涵盖去中心化访问控制的安全属性。

参考文献

[1] Luca Bombelli、Joohan Lee、David Meyer 和 Rafael D Sorkin。1987。时空作为因果集。《物理评论快报》，59，5，521。[2] Herb Caudill。2023。本地优先身份验证：使用安全的密码学签名链，为团队协作提供去中心化身份验证和授权。https://github.com/local-first-web/auth。[3] Vicent Cholvi、Antonio Fernández Anta、Chryssis Georgiou、Nicolas Nicolaou、Michel Raynal 和 Antonio Russo。2021。拜占庭容错分布式只增集合：规范与应用。（2021）。doi：10.48550/ARXIV.2103.08936。[4] Matthew Hodgson。2022。Matrix 2022 年节日更新。Matrix.org 基金会 C.I.C. https://matrix.org/blog/2022/12/25/the-matrix-holiday-update-2022。[5] Florian Jacob、Saskia Bayreuther 和 Hannes Hartenstein。2022。拜占庭环境中的 CRDT：冲突自由、容忍双重陈述，以及 Matrix 复制数据类型。载于《*Sicherheit 2022*》。德国信息学会。doi：10.18420/sicherheit2022 07。_[6] Florian Jacob、Luca Becker、Jan Grashöfer 和 Hannes Hartenstein。2020。Matrix 分解——对基于事务的有向无环图上无最终性的访问控制方法进行分析。载于第 25 届 ACM 访问控制模型与技术研讨会（SACMAT 2020）。doi：10.1145/3381991.3395399。[7] Florian Jacob、Carolin Beer、Norbert Henze 和 Hannes Hartenstein。2021。Matrix 事件图复制数据类型分析。《*IEEE Access*》，9，28317–28333。doi：10.1109/ACCESS.2021.3058576。[8] Martin Kleppmann。2022。使 CRDT 具备拜占庭容错能力。载于《第九届分布式数据一致性原理与实践研讨会论文集》，8–15。[9] Martin Kleppmann 和 Heidi Howard。2020。拜占庭最终一致性与点对点数据库的根本局限。*arXiv* 预印本 *arXiv:2012.00472*。[10] Leslie Lamport。2019。分布式系统中的时间、时钟与事件排序。载于《并发：*Leslie Lamport* 的著作》。[11] Ranganathan Padmanabhan 和 Sergiu Rudeanu。2008。《格与布尔代数公理》。World Scientific。[12] Nuno Preguiça、Carlos Baquero 和 Marc Shapiro。2018。无冲突复制数据类型（CRDT）。*arXiv* 预印本 *arXiv:1805.06358*。[13] Pierre-Antoine Rault、Claudia-Lavinia Ignat 和 Olivier Perrin。2022。使用 CRDT 为协作应用程序实现分布式访问控制。载于《第九届分布式数据一致性原理与实践研讨会论文集》，33–38。[14] Marc Shapiro、Nuno Preguiça、Carlos Baquero 和 Marek Zawirski。2011。无冲突复制数据类型。载于《自稳定系统研讨会》。Springer，386–400。[15] Matrix.org 基金会 C.I.C. 2022。Matrix 规范 v1.5。技术报告。https://spec.matrix.org/v1.5/。[16] Mathias Weber、Annette Bieniusa 和 Arnd Poetzsch-Heffter。2016。弱一致性复制信息系统的访问控制。载于《国际安全与信任管理研讨会》。Springer，82–97。[17] Matthew Weidner、Heather Miller 和 Christopher Meiklejohn。2020。使用半直积组合与分解基于操作的 CRDT。《*ACM* 程序语言会议录》，4，ICFP。doi：10.1145/3408976。[18] Elena Yanakieva、Michael Youssef、Ahmad Hussein Rezae 和 Annette Bieniusa。2021。使用 CRDT 解决分布式文件系统中的访问控制冲突。载于《第八届分布式数据一致性原理与实践研讨会论文集》，1–3。

A 在 CEDP/EPM 组合中集成系统性访问控制

在本附录中，我们组合一个 CEDP和一个或多个 EPM，为相应的复制数据类型提供访问控制。策略评估所需的属性以及策略本身都存储在复制数据类型中，并作为访问决策和执行的基础。此外，诸如更改权限和策略等管理性访问控制任务，也受复制数据类型中的属性和策略约束。因此，无需借助外部系统来管理访问控制。我们将这种方法称为 系统性 访问控制，并将组合后的数据类型称为 仅扩展有向偏序集及映射的访问控制 （ACEDPM）。该方法的灵感来自 Matrix 消息平台处理访问控制的方式。

访问控制是根据对对象执行不同动作的主体来定义的。在我们的消息传递示例中，用户主体会对其他用户或聊天群组等对象执行各种动作，例如读取和撰写聊天消息，或查询和更改组成员身份。事件是主体对对象执行动作这一行为。我们关注一致性；由于读取动作不会损害一致性，因此在下文中我们仅考虑写入动作。我们将事件作为集合元素存储在数据类型中。为了引入逻辑时间的概念，我们存储已存储事件之间的发生后相等关系，即 CEDP。聊天消息等一次性事件直接存储在 CEDP 中，而组成员身份等属性的变更事件则通过 CEDP 之上的一个或多个 EPM 来存储。

CEDP 和 EPM 复制对象的发起者也定义了组合 ACEDPM 对象的“规则”，即作为基础的初始属性和策略，所有后续常规事件和管理事件都必须根据这些规则获得授权。传统的访问控制理念假定采用集中式、强一致的访问控制执行机制，并依据数据和策略变更按真实世界时间的线性序排列来执行访问控制。这里的挑战在于，CEDP 和 EPM 为了在网络分区下保持可用性，只能提供强最终一致性（SEC）；而我们希望在保持这一权衡的同时整合访问控制。在去中心化、弱一致性的访问控制中，由于数据不一定一致，且变更按部分逻辑时间排序，不同副本在某一真实世界时间点不一定会做出相同的决策。然而，如果组合后的 ACEDPM 仍然是拜占庭容错的无冲突复制数据类型（CRDT），也就是说仍能提供 SEC，那么其状态和访问控制仍是安全的：拜占庭副本无法迫使正确副本在其状态和访问决策上永久分歧。挑战在于如何安全地处理并发管理事件。ACEDPM 会保留所有并发事件并将其线性化。我们在附录A.1中解释这一方法，并在附录A.2中给出 SEC 的证明概要。

A.1 ACEDPM 规范

ACEDPM 以 CRDT 的常见方式处理并发事件：接受所有有效事件，然后对并发事件打破平局；在此处，采用参与者之间基于访问控制的线性排序。我们假设采用点对点系统架构，其中每个副本对应一个拥有完全控制权的用户。ACEDPM 对象存储的事件由主体sbj（发送副本）的动作act及该动作的内容cnt组成。所有事件都隐含地以 ACEDPM 复制对象为对象，也可以有一个显式对象obj，例如另一位用户。我们将事件表示为元组$x = (act,\ sbj,\ cnt)$；如果存在显式对象，则表示为 $x = (act,\ sbj,\ obj \mapsto cnt)$。例如，Alice 撰写一条文本消息时，事件为 (chat，Alice，”Hi!”)。如果 Alice 更改群组成员资格，将 Bob 纳入其中，则得到 (membership，Alice，Bob $\mapsto IN$)。如果 Alice 将自己的访问级别设为 2342，则得到 (level，Alice，Alice $\mapsto 2342$)。由于副本需要验证主体标识符sbj 的正确性，并能够向他人证明其正确性，因此事件需要数字签名。

下文中，我们在 CEDP 因果事件集之上使用两个 EPM 来集成访问控制：一个映射 M 用于组成员身份属性，另一个映射 L 用于访问级别属性。第一个映射 M 通过算法 2 中查询函数 $getM$ 变体的 get 进行访问。可能的成员身份值例如有 $\{OUT,\ IN,\ INVITE,\ BAN\}$。在授权方面，我们要求主体为 IN；不过，例如 Matrix 也会限制成员身份状态转换。第二个映射 L 用于主体和动作的访问级别属性，可通过 $getL$ 访问。级别是与主体和动作相关联的整数，使主体能够存储所有级别小于或等于自身级别的动作所对应的事件。此外，级别还会对主体和对象，以及它们各自发送的事件施加基于访问控制的线性序。关于可能的访问控制系统的更多详情，请参见 [6]中定义的基于级别和属性的访问控制。

对于系统性访问控制，我们修改 EDP 和 EPM 模型，在应用更新之前，使用授权函数 $authorized(u, T)$ 来判断更新 $u \in \mathbb{R}$ 在逻辑时间点 $T \in \mathrm{P}(\mathbb{R})$ 是否获准，并使用基于访问控制的优先级关系R_a 来打破平局。我们首先解释模型中需要进行哪些修改以及原因，然后展示 $authorized(u, T)$ 和 R_a 的算法。在 EDP 的 $effect$函数中（参见算法 1），我们必须断言 $authorized(u, \mathrm{mlb}(u))$，以忽略其极大下界未授权的已接收更新。在 EPM 的 get 函数中（参见算法 2），只有当 $authorized(T_i, \bigcup_{j=0}^{j<i} T_j)$ 为真时，才能应用线性化后的更新 T_i，以忽略未获线性化之前更新授权的已存储更新。最后，在 EPM 的 $linearize$ 函数中，必须使用优先关系 R_a 替代基于哈希的排序 R_h，因为 R_h会以确定但任意的方式对更新排序。严格线性序 R_a 则会优先处理撤销操作，并按照主体的级别排列所有动作，之后再回退到 R_h。

算法 3 基于级别和属性的授权

函数 *authorized* $(u \in \mathbb{R}, T \in \mathcal{P}(\mathbb{R})):\ a \in 0, 1$
▷ 判断 u 在逻辑时间点 T是否获得授权
$x \leftarrow \max(X(u))\ \ M \leftarrow getM(T)\ \ L \leftarrow getL(T)$
组$_a \leftarrow M[x.sbj] = IN$
动作$_a \leftarrow L[x.$动作$] \leq L[x.$主语$]$
$object_a \leftarrow x.sbj\ = x.obj\ \vee L[x.obj] < L[x.sbj]$
级别_上限 $\leftarrow x.$操作 $=$ 级别 $\Rightarrow$ $x.$计数 $\leq L[x.$主题$]$
▷ 对于“设置级别”动作，将新级别限制为主体级别
$a \leftarrow$ 组$_a \wedge$ 动作$_a \wedge$ 对象$_a \wedge$级别_容量

Algorithm 3 中所示的授权 函数用于判断向上扩展 u 是否在逻辑时间点 T 根据属性获得授权。若主体满足以下条件，则事件 x获得授权：a) 是组成员；b) 获准执行该动作；c) 获准对（可选）对象执行操作；d) 未超出其访问级别。主体的级别必须严格高于对象，以防止相互冲突的权限撤销操作造成不一致：同一级别的主体无法撤销彼此的权限。

算法4 中所示的基于访问控制的优先级关系 R_a 是所有向上扩展 $\in \mathbb{R}$之间的严格线性序。在第一顺序中， R_a 优先处理撤销操作，而不是其他无法减少权限的动作，以确保撤销操作在其他并发向上扩展之前应用。在第二顺序中，对于要么都是撤销操作、要么都是其他动作的动作，它根据其主体的级别来确定动作的优先级，以确保级别较高的主体的动作先被应用。在第三顺序中，对于来自级别相同的主体的动作， R_a回退到确定性但任意的基于哈希的排序。

通过在线性化映射更新时优先处理撤销操作，撤销操作获得了“并发+因果逐一”语义，这是 Weidner 基于 [17] 所指出的概念[2]：撤销操作会对抗因果上后续的以及并发的映射更新。当将所有元素的线性化作为查询函数时，撤销操作对所有 ACEDPM 更新都具备这些所需语义，即对 CEDP 和 EPM 都适用。

A.2 ACEDPM 的拜占庭强最终一致性

人们可能会想，彼此之间的并发撤销操作是否会破坏最终交付或强收敛。然而，强最终一致性得到保证，因为并发更新如果得到其向下闭包的授权，就绝不会被 拒绝 ，在另一撤销操作获胜的情况下，它只可能在线性化阶段被忽略。因此，无论接收顺序和应用顺序如何，所有正确副本状态最终都会包含两个撤销操作及其向下闭包。线性化仅基于撤销操作与

算法 4 基于访问控制的优先级关系 R_a

$R_a \leftarrow \{(u_1,\ u_2)\ \in \mathbb{R}^2 |$ 先前$_a(u_1,\ u_2)\}$
函数 先验$_a(u_1, u_2 \in \mathbb{R}):\ b \in \{0, 1\}$
▷ 是否 u_1 在访问控制方面优先于 u_2
$x_1 =$最大值$(X(u_1))\ \ x_2 =$最大值$(X(u_2))$
$L_1 \leftarrow getL(\text{mlb}(u_1))\ \ L_2 \leftarrow getL(\text{mlb}(u_2))$
$b \leftarrow rvc(u_1)\ \wedge \neg rvc(u_2)$ ▷ 优先处理撤销操作
若 $rvc(u_1)\ = rvc(u_2)$ **则**
$b \leftarrow L_1[x_1.sbj] < L_2[x_2.sbj])$
▷ 如果两者都是撤销，或两者都不是撤销，则按级别排序
如果 $L_1[x_1.sbj] = L_2[x_2.sbj]$ **那么**
$b \leftarrow h(u_1)\ < h(u_2)$
▷ 如果主体级别相同，则按哈希值排序

函数 $rvc(u \in \mathbb{R}):\ b \in 0, 1$ ▷ 是否 u是撤销
$x \leftarrow$ 最大值$(X(u))$
$M_{pre} \leftarrow getM(\text{mlb}(u))\ \ M_{post} \leftarrow M_{pre} \uplus u$
$L_{pre} \leftarrow$获取$L(\text{mlb}(u)),\ \ L_{post} \leftarrow M_{pre} \uplus u$
$b \leftarrow (M_{pre}[x.obj] = IN \wedge M_{post}[x.obj] \neq IN)$
$\vee L_{pre}[x.$对象$] > L_{post}[x.$对象$]$

彼此及其各自的向下闭包，从而所有正确副本最终将达成相同的决定。

定理 4。 在所有正确副本构成一个连通分量且该分量内存在最终通信的假设下，即使面对任意数量的拜占庭副本，*ACEDPM* 仍是一种基于操作的无冲突复制数据类型。

证明概要。 基于底层 CEDP/EPM 的强最终一致性。**自更新：** 正确副本只会生成通过其本地当前授权检查的更新。因此，授权不会妨碍副本立即进行自更新。**最终更新：** 正确副本生成的更新会根据其极大下界进行授权。接收更新的正确副本仅根据该更新的极大下界检查其是否获得授权。因此，它们只会过滤掉无法由正确副本发送的无效更新。但它们会应用所有根据其极大下界获得授权的更新，即使并发更新已基于接收副本状态的极大元素，撤销了 EPM 状态中的权限。冲突的解决被推迟到线性化阶段，而不是通过最终不应用更新来解决。**强收敛：** 对于 CEDP，授权检查在应用更新之前进行，因此，授权不会导致已应用相同更新的两个副本产生差异。对于 EPM，线性化过程中会进行授权检查，若更新在线性化顺序中未获授权，则会忽略该更新。不过，线性化是确定性的，且不受应用更新顺序的影响。因此，处于相同 CEDP 状态的副本会得到相同的线性化结果，从而也会得到相同的 EPM 状态。□

[2]https://mattweidner.com/2022/02/10/collaborative-data-design.html