无序数据群体协议

Michael Blondin 计算机科学系，舍布鲁克大学，加拿大
FrançoisLadouceur 计算机科学系，舍布鲁克大学，加拿大

摘要

群体协议是一种成熟的计算模型，由具有固定大小内存的被动移动匿名代理组成。众所周知，群体协议可以计算 Presburger 可定义的谓词，例如绝对多数和计数谓词。在这项工作中，我们开始研究在任意大的数据域上运行的群体协议。更确切地说，我们引入无序数据群体协议这一形式，用于对基于无序数据序列的匿名群体计算进行推理。我们首先证明，可以判定来自无限数据域的无序序列中是否存在一个占绝对多数的数据项。随后，我们确立了模型在“即时观察”限制下的表达能力；具体而言，在每次交互中，一个代理观察另一个并且对该交互毫不知情的代理。

2012 年 ACM 学科分类 计算理论 → 分布式计算模型；计算理论 → 无限对象上的自动机

关键词与短语 群体协议、无序数据、彩色 Petri 网

经费资助 *Michael Blondin*：获加拿大自然科学与工程研究委员会（NSERC）的 Discovery Grant 资助，并获魁北克自然与技术研究基金（FRQNT）资助。

François Ladouceur：获加拿大自然科学与工程研究委员会（NSERC）奖学金资助，并获 J.A. De Sève 基金会资助。

致谢 我们感谢 Manuel Lafond 在研究初期提出的想法并给予反馈。我们还感谢匿名审稿人提出的意见和富有洞见的建议。

arXiv:2305.00872v2 [cs.DC] 4 Jul 2023

1 引言

背景。 群体协议是一个成熟的计算模型，用于描述具有固定大小内存的被动移动匿名代理 [1]。群体协议可用于，例如，对化学反应网络和移动传感器网络进行形式化分析（有关群体协议以及更广义的动态网络综述，见 [23]）。

在群体协议中，匿名代理持有来自某个有限集合的可变状态。它们共同尝试计算群体初始全局状态上的一个谓词。在每个离散时刻，调度器选取两个代理，它们根据各自的当前状态共同更新状态。假设这样的调度器是“公平的”（或者等价地，以均匀随机的方式选取代理配对）。我们用一个用于绝对多数谓词的经典协议来说明这一模型。考虑一个由 ℓ 个（匿名）代理组成的群体，每个代理的初始状态为 Y 或 N，它们要计算 Y 的数量是否超过 N 的数量，即共同计算谓词 $\varphi(\#Y, \#N) :\equiv (\#Y > \#N)$。例如，一个由 $\ell = 5$ 个代理组成的群体可以初始化为 $\{\{Y, N, Y, Y, N\}\}$。两个代理按以下四条规则进行状态更新：

强到弱	胜方延续	决胜局
$\{\{Y, N\}\} \to \{\{n, n\}\}$	$\{\{Y, n\}\} \to \{\{Y, y\}\}$ $\{\{N, y\}\} \to \{\{N, n\}\}$	$\{\{y, n\}\} \to \{\{n, n\}\}$

上述群体的一种可能执行是 $\{\{Y, N, Y, Y, N\}\} \to \{\{Y, N, Y, n, n\}\} \to \{\{Y, n, n, n, n\}\} \to \{\{Y, y, n, n, n\}\} \to \cdots \to \{\{Y, y, y, y, y\}\}$。

状态为 $\{Y, y\}$ 的代理认为 φ 的输出应为 真，而状态为 $\{N, n\}$的代理则认为它应为假。因此，在上述执行过程中，群体达成了持久的真共识（尽管没有代理在局部能够确定这一点）。

众所周知，群体协议恰好能够计算可用 Presburger 算术定义的谓词，也就是带加法和序关系的自然数上的一阶逻辑。这一结论最初通过凸几何 [2], 得到证明，后来又利用向量加法系统理论 [15]再次得到证明。例如，这意味着，给定投票选项 $\{1, \ldots, k\}$，存在一种群体协议，可以判断某个选项 i 是否获得绝对多数，即是否有超过 $\ell/2$ 的 ℓ 代理在初始时持有相同的 $i \in \{1, \ldots, k\}$。

由于 k 必须存储在状态空间中，这种多数协议只能处理固定数量的投票选项。由于规则也取决于 k，这意味着若要处理更大的 k，就需要重新配置整个群体，例如在提供新的投票选项时便是如此。从概念上说，这在由匿名移动智能体组成的群体中并不实际。作为替代方案，我们提出，可以将代理的输入优雅地建模为取自一个无限集合 $\mathbb{D}$，且规则不依赖于 $\mathbb{D}$。

贡献。在这项工作中，我们开启了对运行于任意大域上的群体协议的研究。更准确地说，我们提出了一种更一般的模型，其中每个代理都携带一个来自无限域的只读数据项 $\mathbb{D}$ 以及一个来自有限集合的可变状态。在这种设置下，群体可以尝试确定是否存在绝对多数数据项。例如，若 $\mathbb{D} := \{1, 2, 3, \ldots\}$，那么以$\{\{(1, x), (1, x), (2, x), (3, x), (1, x)\}\}$初始化的群体应当达到持久的真共识，而以$\{\{(1, x), (1, x), (2, x), (3, x), (2, x)\}\}$初始化时则应当达到持久的假共识。

与标准模型一样，公平调度器会选取一对代理。交互依据如下形式的规则进行：$\{\{p, q\}\} \xrightarrow{d \sim e} \{\{p', q'\}\}$，其中 $d, e \in \mathbb{D}$ 是两个代理的数据值，而 $\sim \in \{=, \neq\}$ 用于比较它们。对于状态，我们假设可以有任意多个代理以同一数据项初始化；并且代理只能通过相等或不等关系比较数据。因此，$\mathbb{D}$ 并非一组（唯一的）标识符，代理因而仍像标准模型中那样保持匿名。

为说明我们提出的计算模型，我们首先展示它能够计算绝对多数谓词。这意味着，一个单一 协议可以处理绝对多数投票中的任意 数量的选项。从分布式计算的角度来看，这为研究无序数据序列上的匿名群体计算提供了一个框架。从计算机辅助验证的角度来看，这使得形式化分析单个协议成为可能（例如 将其建模为彩色 Petri 网），而不必诉诸参数化验证；在计数器系统的背景下，参数化验证尤其困难。

作为确定无序数据群体协议表达能力的一块垫脚石，我们接下来刻画即时观察协议。在这一经过充分研究的限制类别中，规则的形式为$\{\{p, q\}\} \xrightarrow{d \sim e} \{\{p, q'\}\}$，即 一个代理通过观察另一个代理（而后者并未察觉它）来更新自身状态。在标准群体协议中，已知这一类协议恰好能够计算来自$\mathbf{COUNT}_*$ [2]的谓词。后者是如下形式谓词的布尔闭包：$\#q \geq c$，其中 $\#q$ 表示状态为 q 的代理数量，而 $c \in \mathbb{N}$ 是一个常数。在我们的情形中，我们证明即时观察协议恰好能够计算

区间谓词，它们是此类简单区间谓词的布尔组合：

$$\exists \text{ pairwise distinct } d_1, d_2, \ldots, d_n \in \mathbb{D} : \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{m} \#(d_i, q_j) \in T(i,j), \tag{1}$$

此处 $\#(d_i, q_j)$ 统计状态为 q_j 且数据项为 d_i 的代理，而每个 $T(i, j) \subseteq \mathbb{N}$ 是一个区间。

为了证明即时观察协议计算的内容不超出区间谓词，我们利用（有限支撑的）数据向量构成良拟序这一事实。尽管我们的方法受到 [2], 的启发，但要同时处理多个无界来源更为棘手：数据值的数量、具有给定数据项的代理数量，以及具有给定状态的代理数量。作为一个附带结果，我们证明绝对多数谓词无法由即时观察协议计算。

为了证明反方向，即区间谓词可由即时观察协议计算，我们描述了一个用于简单区间谓词的协议。与标准设定不同，我们需要实现存在量化。这通过数据领导者选举和全局领导者选举来实现。我们将后者选出的代理称为“控制器”。它的作用是处理数据领导者选择其在 (1) 中角色的记账工作。我们精心实现了一种纠正机制，确保只有在为存在量化锁定正确赋值后，群体才会达成一个真-共识。

相关工作。 验证与并发领域的研究者已经发现，群体协议可以改写为佩特里网。特别是，这使得对群体协议进行自动形式化分析 [15, 7]并发现其状态复杂度界限 [12, 6]成为可能。我们的灵感来自相反的方向：受近期对彩色 Petri 网 [17, 19, 18]的关注启发，我们引入带数据的协议。我们的模型对应于无序数据佩特里网，其中颜色和令牌数量是不变量。

用于计算多数和相对多数的群体协议已得到广泛研究（例如，近期结果见 [14, 4, 5, 3] ）。据我们所知，最接近的工作是 [16]，其中作者提出了用于多数问题变体（包括多方共识）的节省空间的确定性协议族。他们将 k 投票选项视为“颜色”，并用存储在代理中的 $\lceil \log k \rceil$ 位来指定。

其他一些包含某种数据的分布式系统模型也彼此不可比较，例如寄存器自动机的广播网络 [13], 分布式寄存器自动机 [9], 和分布式存储器自动机 [10]。这些形式化模型受到寄存器自动机 [20], 的启发，允许使用身份、控制结构和替代通信机制；而这些在群体协议中均不被允许。

论文结构。 第2节给出了基本定义，并介绍了我们的模型。在第3节中，我们提出了一种用于计算绝对多数谓词的协议。第4节确立了即时观察协议的表达能力。最后，我们在第5节中总结全文。

2 预备知识

我们分别用 $\mathbb{N}$ 和 $[a..b]$ 表示集合 $\{0, 1, 2, \ldots\}$ 和 $\{a, a+1, \ldots, b\}$。多重集 $\boldsymbol{m}$ 在 E 上的支撑集定义为 $\mathsf{act}(\boldsymbol{m}) \coloneqq \{e \in E : \boldsymbol{m}(e) > 0\}$（我们使用记号 $\mathsf{act}(\boldsymbol{m})$ 而非 $\mathsf{supp}(\boldsymbol{m})$，因为后文将提到“活跃状态”。）我们用 $\mathbb{N}^E$ 表示 E 上支撑集有限的多重集集合。空多重集记作 $\mathbf{0}$，满足对于所有 $e \in E$，$\mathbf{0}(e) = 0$。设 $\boldsymbol{m}, \boldsymbol{m}' \in \mathbb{N}^E$。当且仅当对于所有 $e \in E$，$\boldsymbol{m} \leq \boldsymbol{m}'$ 满足 $\boldsymbol{m}(e) \leq \boldsymbol{m}'(e)$ 时，我们记作 $\boldsymbol{m} \leq \boldsymbol{m}'$。我们将 $\boldsymbol{m} + \boldsymbol{m}'$ 定义为满足对于所有 $e \in E$，$(\boldsymbol{m} + \boldsymbol{m}')(e) := \boldsymbol{m}(e) + \boldsymbol{m}'(e)$ 的多重集。差（记作 $\boldsymbol{m} - \boldsymbol{m}'$）的定义类似，但要求 $\boldsymbol{m} \geq \boldsymbol{m}'$。

2.1 无序数据群体协议

定义在配备相等关系的无限域 $\mathbb{D}$ 上的无序数据群体协议 ，
是一个元组（Q，δ，I，O），其中

- Q 是一个有限的元素集合，称为状态，$\delta \subseteq Q^2 \times$
- $\{=, \neq\} \times Q^2$ 是转移集合，$I \subseteq Q$ 是初始状态
- 集合，并且O: $Q \to \{$假，真$\}$ 是输出函数。
-

我们将 $\mathbb{D}$ 中的一个元素称为数据项或颜色。在本文中，我们始终隐含地假设，对于所有 p，$q \in Q$和 $\sim \in \{=, \neq\}$， δ都包含 $((p, q) \sim, (p, q)$。

形式 $\boldsymbol{f}$ 是 $\mathbb{N}^Q$中的一个元素。我们用 $\mathbb{F}$表示所有形式的集合。给定 $Q' \subseteq Q$，令$\boldsymbol{f}(Q')$:$= \sum_{q \in Q'} \boldsymbol{f}(q)$。配置是一个从 $\mathbb{D}$ 到 $\mathbb{F}$ 的映射 C，满足 $\mathsf{supp}(\mathsf{C}) := \{d \in \mathbb{D} : \mathsf{C}(d) \neq \mathbf{0}\}$为有限集，并且$\sum_{d \in \mathbb{D}, q \in Q} \mathsf{C}(d)(q) \geq 2$。我们常将 $\mathsf{C}(d)(q)$写作 $\mathsf{C}(d, q)$。直观地说，后者表示数据项 d 处于状态 q的代理数量。我们将此记法扩展到状态子集：$\mathsf{C}(d, Q') := \mathsf{C}(d)(Q')$。我们自然地将 $+$、 $-$ 和 $\leq$ 扩展到$\mathbb{D} \to \mathbb{F}$，例如，$\mathsf{C} + \mathsf{C}'$ 满足：对于所有 $d \in \mathbb{D}$，$(\mathsf{C} + \mathsf{C}')(d) := \mathsf{C}(d) + \mathsf{C}'(d)$。

设 C 为一个配置。我们将活跃状态定义为集合 $\mathsf{act}(\mathsf{C}) := \{q \in Q : \mathsf{C}(d, q) > 0$ 对于某个 $d \in \mathbb{D}\}$。如果 $\mathsf{act}(\mathsf{C}) \subseteq I$，则称 C 为初始配置。给定 $Q' \subseteq Q$，令 $|\mathsf{C}|_{Q'} := \sum_{d \in \mathbb{D}} \mathsf{C}(d, Q')$，且 $|\mathsf{C}| := |\mathsf{C}|_Q$。C 的输出定义为 $O(\mathsf{C}) := b$ 当且仅当 $O(q) = b$对每个 $q \in \mathsf{act}(\mathsf{C})$ 都成立；否则，定义为 $O(\mathsf{C}) = \perp$。直观而言， $O(\mathsf{C})$ 表示所有代理是否就某个输出 b达成一致。

▶ 例1。设 $\mathbb{D} := \{, , \ldots\}$，且 $Q := \{p, q\}$。设 $\boldsymbol{f} := \{\{p, p, q\}\}$，且 $\boldsymbol{f}' := \{\{q\}\}$。设 $\mathsf{C} := \{ \mapsto \boldsymbol{f}, \mapsto \boldsymbol{f}', \mapsto \mathbf{0}, \ldots\}$。我们有 $\mathsf{C}(\ p) = 2$，$\mathsf{C}(\ q) = \mathsf{C}(\ q) = 1$，$\mathsf{C}(\ p) = \mathsf{C}(\ p) = \mathsf{C}(\ q) = \mathbf{0}$，且 $|\mathsf{C}|_{\{ , \}} = 2$。配置 C 表示一个由四个代理组成的群体，每个代理携带一个不可变数据项和一个可变状态：$\{\{(\ p), (\ p), (\ q), (\ q)\}\}$。◀

为简洁起见，给定一个形式 $\boldsymbol{f}$，令 $\boldsymbol{f}_d$: $\mathbb{D} \to \mathbb{F}$定义为 $\boldsymbol{f}_d(d) := \boldsymbol{f}$和$\boldsymbol{f}_d(d') := \mathbf{0}$，对每个 $d' \neq d$均成立。此外，给定一个状态 q，令 $\boldsymbol{q}_d$: $\mathbb{D} \to \mathbb{F}$ 定义为$\boldsymbol{q}_d(d)(q) := 1$ 和 $\boldsymbol{q}_d(d')(q') := 0$，对每个 $(d', q') \neq (d, q)$ 均成立。

设 C 为一个配置，并令 $t = ((p, q) \sim, (p', q')) \in \delta$。如果存在 $d, e \in \mathbb{D}$ 使得 $d \sim e, \mathsf{C} \geq \boldsymbol{p}_d + \boldsymbol{q}_e$，我们就称转移 t 在 C 中可启用。若后者成立，则可使用 t 得到配置 $\mathsf{C}' := \mathsf{C} - (\boldsymbol{p}_d + \boldsymbol{q}_e) + (\boldsymbol{p}'_d + \boldsymbol{q}'_e)$，记作 $\mathsf{C} \xrightarrow{t} \mathsf{C}'$。我们用 $\mathsf{C} \to \mathsf{C}'$ 表示对于某个 $t \in \delta$，$\mathsf{C} \xrightarrow{t} \mathsf{C}'$ 成立。我们进一步定义

$\xrightarrow{*}$ 作为 $\to$的自反传递闭包。

▶ **示例 2.** 令 $O(p) :=$ 假， $O(q) :=$ 真 且 $t := ((p, q) =, (q, q))$。使用示例 1 中表示配置的记法，我们有：

$$\{\!\{(\bullet, p), (\bullet, p), (\bullet, q), (\blacksquare, q)\}\!\} \xrightarrow{t} \{\!\{(\bullet, p), (\bullet, q), (\bullet, q), (\blacksquare, q)\}\!\} \xrightarrow{t} \{\!\{(\bullet, q), (\bullet, q), (\bullet, q), (\blacksquare, q)\}\!\}.$$

令 $\mathbf{C}$、$\mathbf{C}'$ 和 $\mathbf{C}''$ 表示上述三个配置。我们有 $O(\mathbf{C}) = O(\mathbf{C}') = \perp$且 $O(\mathbf{C}'') =$真。此外，转移 t 在 $\mathbf{C}''$ 中未启用，因为不存在满足$\mathbf{C}''(d, p) \geq 1$ 且 $\mathbf{C}''(d, q) \geq 1$的数据项。因此，代理“已收敛到真共识”。◀

执行是一个无限的配置序列 $\mathsf{C}_0\mathsf{C}_1 \cdots$，满足 $\mathsf{C}_0 \to \mathsf{C}_1 \to \cdots$。如果存在 $\tau \in \mathbb{N}$ 使得 $O(\mathsf{C}_\tau) = O(\mathsf{C}_{\tau+1}) = \cdots = b$，我们就称这样的执行收敛到输出 $b \in \{\mathsf{false}, \mathsf{true}\}$。执行 $\mathsf{C}_0\mathsf{C}_1 \cdots$是公平的，如果对于每个配置 C'，只要 $|\{i \in \mathbb{N} : \mathsf{C}_i \xrightarrow{*} \mathsf{C}'\}| = \infty$，就有 $|\{i \in \mathbb{N} : \mathsf{C}_i = \mathsf{C}'\}| = \infty$。换言之，公平性表示，如果配置 C' 无限次可达，那么它就会在执行过程中无限次出现。非正式地说，这意味着某些“进展”无法永远被阻止。

设 Σ 是一个非空有限集合。输入是某个$\mathbf{M} \in \mathbb{N}^{\mathbb{D}\times\Sigma}$，其中$\sum_{d\in\mathbb{D},\ \sigma\in\Sigma} \mathbf{M}(d,\ \sigma) \geq 2$。输入$\mathbf{M}$通过双射输入映射 $\iota\colon \Sigma \to I$，转换为初始配置$\iota(\mathbf{M}) := \sum_{d\in\mathbb{D},\ \sigma\in\Sigma} \sum_{j=1}^{\mathbf{M}(d,\sigma)} \boldsymbol{\iota}(\boldsymbol{\sigma})_d$。如果对于每个输入$\mathbf{M}$，从 $\iota(\mathbf{M})$ 开始的每个公平执行都收敛到输出 $\varphi(\mathbf{M})$，我们就称协议计算谓词 φ。有时我们会滥用记号，用 $\varphi(\mathbf{C}_0)$ 表示 $\varphi(\iota^{-1}(\mathbf{C}_0))$。

▶ 例 3。令 $\mathbb{D} := \{ , , \ldots\}$，$\Sigma := \{x_1, \ldots, x_4\}$，$I := \{q_1, \ldots, q_4\}$ 和 $\iota(x_i) := q_i$。输入 $\mathbf{M} := \{\{(\ x_1), (\ x_1), (\ x_2), (\ x_2), (\ x_2), (\ x_3), (\ x_3)\}\}$ 产生初始配置 $\iota(\mathbf{M}) = \{ \mapsto \{\{q_1, q_1, q_2, q_2, q_2, q_4\}\} \mapsto \{\{q_1, q_3\}\} \mapsto 0, \ldots\}$。

注意，与标准协议一样，无序数据群体协议所计算的谓词集合对布尔运算封闭。给定一个计算 φ 的协议，只需将所有 $q \in Q$ 对应的 $O(q)$ 的值改为 $\neg O(q)$，即可得到一个计算 $\neg\varphi$ 的协议。给定分别由协议 $(Q_1, \delta_1, I_1, O_1)$ 和 $(Q_2, \delta_2, I_2, O_2)$ 计算的谓词 ψ_1 和 ψ_2，让两个协议并行运行，便可轻松得到一个计算 $\psi = \psi_1 \wedge \psi_2$ 的协议。具体做法是定义 $(Q := Q_1 \times Q_2, \delta, I := I_1 \times I_2, O)$，其中

- δ对每个 $((p_1, q_1) \ \sim, (p_1', q_1')) \in \delta_1$，包含 $(((p_1, p_2), (q_1, q_2)) \ \sim, ((p_1', p_2), (q_1', q_2)))$；$\delta$对每个 $((p_2, q_2) \ \sim, (p_2', q_2')) \in \delta_2$，包含 $(((p_1, p_2), (q_1, q_2)) \ \sim, ((p_1, p_2'), (q_1, q_2')))$；$O(q_1, q_2) = O_1(q_1) \wedge O_2(q_2)$。

3 多数谓词协议

令 $\Sigma := \{x\}$。本节介绍一个用于绝对多数谓词的协议，该谓词定义为 $\varphi_{\mathrm{maj}}(\mathbf{M}) := \exists d \in \mathbb{D} : \mathbf{M}(d, x) > \sum_{d'\neq d} \mathbf{M}(d', x)$。由于每个输入对的形式为 (d, x)，且 $d \in \mathbb{D}$，我们在对协议的非正式介绍中省略“虚拟元素” x。请注意，为简洁起见，本文余下部分将使用“多数”一词代替“绝对多数”。

我们的协议与经典（顺序）Boyer–Moore 算法 [11] 并无二致：我们试图选出一种颜色作为多数候选者，然后检查它是否确实占多数。该协议旨在分阶段运行。在配对阶段，每个未配对智能体都会尝试与一个颜色不同的未配对智能体配成一对。例如，如果初始群体为{{ , , , , , , }}，那么我们（以非确定性的方式）最终会得到以下两种配对之一：

已配对的代理	未配对的代理
<escape>{{ , , }} </escape>	{{ }}
<escape>{{ , </escape> }}	{{ , , }}

未配对的代理必须全部具有相同的颜色 d，例如上例中的“ ”。此外，如果群体存在多数颜色，那么该颜色必须是 d。

由于代理是匿名的，且具有有限内存，因此它们实际上无法记住自己曾与谁配对。因此，一旦选出候选颜色（例如上述示例中的“ ”），便会进入分组阶段。在这一阶段，未配对代理会向与自己颜色相同的代理表明，它们属于候选多数群体。具体做法是将值“Y”存储在内部，以表示“是”。类似地，未配对代理通过使用“N”向颜色不同的代理表明，它们属于候选少数群体。完成这一过程后，便会按照引言中介绍的经典协议进入多数阶段。

这种理想化描述会引出两个问题。首先，协议旨在分阶段运行，但由于其分布式特性，各阶段可能会并发发生。因此，我们加入一种纠正机制：

- 如果候选多数颜色为 d 的未配对智能体找到一个颜色为 d 的已配对智能体（分别为 $d' \neq d$）且其带有“N”（分别为“Y”），则将其翻转为“Y”（分别为“N”）；
- 如果候选多数颜色为 d 的未配对智能体找到一个颜色为 d 的已配对智能体（分别为 $d' \neq d$）且其带有“n”或“y”中的任意一个，则将其翻转为“Y”（分别为“N”）。

中间值 Y（分别为 N）必须恢复为 Y（分别为 N），方法是找到一个最初扮演角色 Y（分别为 N）的代理，并将其重置为原始值。

第二个问题与以下事实有关：在代理数量为偶数的群体中，所有代理都可能完成配对。在这种情况下，不会留下未配对智能体来对代理进行分组。为解决这一问题，每个代理都携带一个“偶数位”，用于表示其对于是否仍有某个未配对智能体的判断。

3.1 状态

状态集合定义为 $Q \coloneqq \{\text{假}, \text{真}\}^3 \times \{Y, N, Y, N, y, n\}$。为了便于读者理解，我们使用四个“宏”来操作状态。每个宏都有一组可能的值；每个状态都是不同宏对应值的组合。

$name$	$q \in Q$ 的值	$q \in I$ 的值
配对(q)	{假, 真}	*false*
组(q)	{假, 真}	$true$

$name$	$q \in Q$ 的值	$q \in I$ 的值
偶数(q)	{假, 真}	假
多数(q)	$\{Y, N, \overline{Y}, \overline{N}, y, n\}$	Y

输入映射由 $\iota(x) \coloneqq q_I$ 定义，其中 q_I 是 I 的唯一状态。非正式地说，pair(q) 表示代理是否已配对；grp(q) 表示代理是否属于候选多数群体；maj(q) 是多数计算的当前值；而 even(q) 是偶数位。

3.2 转换与阶段

我们通过为每个阶段引入相应的规则来描述该协议。注意，规则是一种可作为转移依据的结构；因此，单条规则可以产生多个性质相同的转移。为方便起见，有些引理会在实际证明之前陈述，因为必须先定义完整的转移集合才能证明它们。附录中的证明会考虑完整的转移列表。

由于该协议的转移集合较长，我们采用“前置条件-更新”记法来呈现：对于分别处于状态 p、$q \in Q$、且颜色分别为 d_1、$d_2 \in \mathbb{D}$的任意两个代理，只要满足对 p、q和 d_1、d_2 的前置条件，就使用一条转移。此类交互的结果是，初始处于状态 p 的代理将其状态更新为 p'，其中 p'与 p 相同，只是指定的宏有所不同； q也同理。为提高可读性，后续表格将状态 p 和 q 的前置条件与更新分别列在不同的行中。

3.2.1 配对阶段

第一条规则用于配对阶段，其主要目标是尽可能多地将代理与颜色不同的代理配对：

$rule$	状态前置条件	颜色前置条件	状态更新
(1)	¬配对(p) ¬配对(q)	$d_1 \neq d_2$	配对(p') ∧偶数(p') 配对(q') ∧偶数(q')

这条规则引出了以下引理，讨论配对阶段的结束以及未配对代理（如果存在）的性质。在本节余下部分，我们固定一个公平执行 $\mathbf{C}_0\mathbf{C}_1\cdots$，其中 $\mathbf{C}_0$ 是初始的。此外，令 $P := \{q \in Q : \text{配对}(q)\}$ 且 $U := Q \setminus P$。

▶ **引理 4.** 存在 $\tau \in \mathbb{N}$ 使得 $|\mathbf{C}_\tau|_U = |\mathbf{C}_{\tau+1}|_U = \cdots$。此外，对于每个 $i \geq \tau$，$\mathbf{C}_i$ 中的所有未配对代理都具有相同的颜色，即集合 $\{d \in \mathbb{D} : \mathbf{C}_i(d, U) > 0\}$ 要么为空，要么是单元素集合。

令 α 表示由引理 4 给出的最小阈值 τ，非正式地说，它就是“配对阶段的终点”。

▶ **引理 5.** 设 $i \in \mathbb{N}$。若 $\varphi_{maj}(\mathbf{C}_0)$ 且 d 是多数颜色，则 $\mathbf{C}_i(d, U) > 0$。

3.2.2 分组阶段

接下来这组转移旨在正确设置每个代理的群体，以表示其在多数计算中的状态。一个代理要么属于候选多数群体（真），要么属于候选少数群体（$false$）。请注意，如果 $\mathbf{C}_\alpha$ 中不存在未配对代理，则该群体（及其相关的多数计算值）无关紧要；这一特殊情况通过偶数位处理，目前暂不考虑。

$rule$	状态前置条件	颜色前置条件	状态更新
(2)	¬配对(p) 配对(q) ∧ 组(q) ∧多数(q) $= Y$	$d_1 \neq d_2$	$none$ ¬组(q') ∧ 多数(q') $= N$
(3)	¬配对(p) 配对(q) ∧ ¬组(q) ∧ 多数(q) $= N$	$d_1 = d_2$	$none$ 组(q') ∧ 多数(q') $= Y$
(4)	¬配对(p) 配对(q) ∧ 组(q) ∧ 多数(q) $\in \{y, n\}$	$d_1 \neq d_2$	$none$ 多数(q') $= \overline{N}$
(5)	¬配对(p) 配对(q) ∧ ¬组(q) ∧ 多数(q) $\in \{y, n\}$	$d_1 = d_2$	$none$ 多数(q') $= \overline{Y}$

以下规则属于一个由两条规则组成的组合，其目的是纠正分组分配中的错误。对于在候选少数群体（分别为多数群体）中参与计算、且遇到当前有效的同色多数候选者（分别为不同颜色的多数候选者）的代理，该组合允许他们通过找到另一名同样参与计算的代理并让其执行相同操作，将自己的值重置为 Y（分别为 N），并将自己的组重置为 真（分别为 假）。这一组合与下一小节中描述的规则一起，确保以下不变量成立。设 $Q_a := \{q \in Q : \text{多数}(q) = a\}$，$Q_M := \{q \in Q : \text{组}(q)\}$ 且 $Q_m := Q \setminus Q_M$。

▶ **引理 6.** 对于每个 $i \in \mathbb{N}$，都有 $|\mathbf{C}_i|_{Q_Y} - |\mathbf{C}_i|_{Q_N} = |\mathbf{C}_i|_{Q_M} - |\mathbf{C}_i|_{Q_m}$。

$rule$	状态前置条件	颜色前置条件	状态更新
(6)	组(p) ∧多数(p) $= \overline{N}$ ¬组(q) ∧ 多数(q) $\in \{y, n\}$	$none$	¬组(p') ∧ 多数(p') $= N$ 多数(q') $= N$
(7)	¬组(p) ∧多数(p) $= \overline{Y}$ 组(q) ∧ 多数(q) $\in \{y, n\}$	$none$	组(p') ∧ 多数(p') $= Y$ 多数(q') $= Y$
(8)	组(p) ∧多数(p) $= \overline{N}$ ¬组(q) ∧多数(q) $= \overline{Y}$	$none$	¬组(p') ∧多数(p') $= n$ 组(q') ∧多数(q') $= n$

我们给出以下示例，以帮助说明在所提出的协议中使用中间状态的必要性Y， N。

▶ **例7**。 我们首先考虑从初始群体 {{ , , , }}开始的一次可能执行，其中不存在多数数据。请注意，由于代理数量为奇数，在任何执行中，配对阶段结束后都会有一个数据项对应一个未配对代理。为便于演示，假设该数据项为蓝色（ ）。进入分组阶段后，该蓝色代理最终会让其他代理知晓它们不属于多数候选组，并且在某个时刻会发生规则（11），从而使所有代理永久处于 maj(q) $\in \{N, n\}$。这在下面三张快照中进行了总结（为清晰起见，省略了偶数位）：

输入	pair	grp	maj
●	×	✓	Y
●	×	✓	Y
■	×	✓	Y
■	×	✓	Y
◆	×	✓	Y

$\xrightarrow{*}$

输入	pair	grp	maj
●	✓	×	N
●	✓	×	N
■	✓	×	N
■	✓	×	N
◆	×	✓	Y

$\rightarrow$

输入	pair	grp	maj
●	✓	×	N
●	✓	×	n
■	✓	×	N
■	✓	×	N
◆	×	✓	n

现在，考虑一个确实存在多数数据的群体：{{ , , , , , , }}。请注意，该群体严格大于前一个群体。因此，我们可以立即获得一个与上述描述相似的配置，其中还有两个数据项为红色（ ）的代理尚未参与计算。由于计算必须输出 真，由蓝色代理发起的关于 $\{N, n\}$ 的共识必须被撤销。

在这种情况下，最后一次配对通过蓝色代理（ ）与一个新引入的红色代理（ ）之间的交互完成后，错误处理首先依照规则（4）或（5）进行：未配对的红色代理（ ）将其计算值设为 N，以此通知蓝色代理（ ）其分组不正确；同样，它还会通知所有此前参与了（现已不正确的）多数阶段的红色代理（ ），让它们将计算值切换为 Y。这总结在以下三个快照中：

…

输入	pair	grp	maj
●	✓	×	N
●	✓	×	n
■	✓	×	N
■	✓	×	N
◆	✓	✓	n
●	✓	✓	Y
●	×	✓	Y

$\xrightarrow{*}$

输入	pair	grp	maj
●	✓	✓	Y
●	✓	×	$\overline{Y}$
■	✓	×	N
■	✓	×	N
◆	✓	✓	$\overline{N}$
●	✓	✓	Y
●	×	✓	Y

$\rightarrow$

输入	pair	grp	maj
●	✓	✓	Y
●	✓	✓	n
■	✓	×	N
■	✓	×	N
◆	✓	×	n
●	✓	✓	Y
●	×	✓	Y

根据规则（6）、（7）或（8），这必然会促使每个分组错误的代理纠正其分组位以及计算值。这样，我们就得到一个配置，其中分组阶段已经结束，并且多数阶段要么尚未启动，要么已使用正确的多数候选者正确启动。◀

以下引理表明，如果 $\mathbf{C}_\alpha$ 中存在未配对代理，分组阶段最终会结束。此外，这些引理还表明，多数候选颜色 d 最终会将多数群体扩散至颜色为 d 的代理，并将少数群体扩散至颜色为 $d' \neq d$ 的代理。

▶ **引理 8.** 令 E 为参与多数计算的状态集合，即 $E := \{q \in Q : \mathrm{maj}(q) \in \{y, n, Y, N\}\}$。令 $E_M := E \cap Q_M$ 且 $E_m := E \cap Q_m$。对于每个 $i \in \mathbb{N}$，下列结论成立：$|\mathbf{C}_i|_{E_M} = |\mathbf{C}_i|_{E_m}$。

▶ **引理 9。** 设 $d \in \mathbb{D}$。若 $\mathbf{C}_\alpha(d, U) > 0$，则存在某个 $\tau \geq \alpha$，使得对于所有 $i \geq \tau$、$d' \in \mathbb{D}$ 和 $q \in$ 活动状态集合$(\mathbf{C}_i(d'))$，以下结论成立：组$(q) = (d' = d)$。

3.2.3 多数阶段

最后一组转移模拟了一个用于多数谓词的标准群体协议。对于没有多数的偶数规模群体，会出现需要谨慎处理的情况。事实上，对于这样的群体，配对阶段可能不会留下任何未配对的代理。因此，我们给出以下规则来解决这个特定问题。

rule	状态前置条件	颜色前置条件	状态更新
(9)	¬配对(p) 偶数(q)	*none*	*none* ¬偶数(q')
(10)	配对(p) ∧ 偶数(p) 配对(q) ∧ ¬偶数(q)	*none*	*none* 偶数(q')

▶ **引理 10.** 存在 $\tau \geq \alpha$，使得对于每个 $i \geq \tau$ 和 $q \in$活动状态集合$(\mathbf{C}_i)$，当且仅当 $|\mathbf{C}_i|_U = 0$ 时，偶数(q) 成立。

对于其他群体，在配对阶段之后会存在一个唯一的多数候选颜色。要使谓词为真，该候选颜色的代理数量必须多于所有其他颜色的代理数量之和。以下规则可用于验证（或否定）这一点。

rule	状态前置条件	列前置条件	状态更新
(11)	多数$(p) = Y$ 多数$(q) = N$	*none*	多数$(p') = n$ 多数$(q') = n$
(12)	多数$(p) = Y$ 多数$(q) = n$	*none*	*none* 多数$(q') = y$

rule	状态前置条件	列前置条件	状态更新
(13)	多数$(p) = N$ 多数$(q) = y$	*none*	*none* 多数$(q') = n$
(14)	多数$(p) = n$ 多数$(q) = y$	*none*	*none* 多数$(q') = n$

▶ **引理 11。** 如果 $\varphi_{maj}(\mathbf{C}_0)$ 成立，那么存在 $\tau \geq \alpha$，使得对于每个 $i \geq \tau$ 和$q \in$ 活动状态集合$(\mathbf{C}_i)$，都有多数$(q) \in \{Y, y\}$ 和 ¬偶数(q) 成立。

▶ 引理 *12.* 如果 $\neg\varphi_{maj}(\mathbf{C}_0)$ 成立，那么存在 $\tau \geq \alpha$，使得以下条件之一成立：对于每个 $i \geq \tau$ 和 $q \in$ 活动状态集合$(\mathbf{C}_i)$，$even(q)$ 都成立；或者对于每个 $i \geq \tau$ 和 $q \in$ 活动状态集合$(\mathbf{C}_i)$，$maj(q) \in \{N, n\}$ 都成立。

我们将给定状态 $q \in Q$ 的输出定义为 $O(q) := ($多数$(q) \in \{Y, y\} \wedge$ ¬偶数$(q))$。根据引理 11 和 12，该协议的正确性可立即得出：

▶ **推论 13。** 存在 $\tau \in \mathbb{N}$ 使得 $O(\mathbf{C}_\tau) = O(\mathbf{C}_{\tau+1}) = \cdots = \varphi_{maj}(\mathbf{C}_0)$。

4 即时观察协议

如果群体协议的每个转移都具有以下形式，则称其为即时观察（*IO*）：$((p,q) \sim (p,q'))$，也就是说，只有一个代理可以通过“观察”另一个代理来更新自己的状态。对 $\sim$没有限制，但也可以设想被观察的是数据项。

本节将刻画即时观察协议的表达能力。首先，我们将建立 IO协议关于截断的性质，从而证明多数谓词不可计算。接着，我们将证明 IO协议计算的谓词不超出区间谓词。最后，我们将证明每个区间谓词都可由 IO协议计算。在继续之前，我们先定义区间谓词。

Let $\dot{\exists} d_1, d_2, \ldots\ldots, d_n$ 表示互不相交的存在量化，即表示对于所有 $d_i \neq d_j\ i, \ j \in [1..n]$ 满足 $i \neq j$。一个简单区间谓词，定义在来自$\mathbb{N}^{\mathbb{D}\times\Sigma}$的输入上，其中 $\Sigma = \{x_1, \ldots\ldots, x_m\}$，具有如下形式：

$$\psi(\mathbf{M}) = \dot{\exists} d_1, d_2, \ldots, d_n \in \mathbb{D} : \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{m} \mathbf{M}(d_i, x_j) \in T(i,j), \tag{2}$$

其中 $m, \ n \in \mathbb{N}_{>0}$，每个 $T(i, \ j) \subseteq \mathbb{N}$ 都是非空区间，并且对于每个 $i \in [1..n]$，都存在 $j \in [1..m]$ 使得 $0 \notin T(i, \ j)$。一个区间谓词是简单区间谓词的布尔组合。

4.1 状态与形式截断

给定配置 C、C′，若存在单射 ρ: $\mathbb{D} \to \mathbb{D}$，使得对于每个 $d \in \mathbb{D}$，$\mathsf{C}(d) \leq \mathsf{C}'(\rho(d))$，则记作 $\mathsf{C} \sqsubseteq \mathsf{C}'$。若 $\mathsf{C} \sqsubseteq \mathsf{C}'$ 且 $\mathsf{C}' \sqsubseteq \mathsf{C}$，则记作 $\mathsf{C} \equiv \mathsf{C}'$。若配置子集 X 满足：对于 $\mathsf{C} \in X$，只要 $\mathsf{C} \sqsubseteq \mathsf{C}'$，就有 $\mathsf{C}' \in X$，则称其向上封闭。若集合 B 满足 $X = \{\mathsf{C}' : \mathsf{C} \sqsubseteq \mathsf{C}'$ 对于某个 $\mathsf{C} \in B\}$，则称其为向上封闭集合 X 的基。

若配置$\mathbf{C}$ 满足以下任一条件，则称其为不稳定：$O(\mathbf{C}) = \perp$，或者存在$\mathbf{C}'$，使得 $\mathbf{C} \xrightarrow{*} \mathbf{C}'$ 且 $O(\mathbf{C}) \neq O(\mathbf{C}')$。令$\mathcal{U}$ 表示不稳定配置的集合，并令$S_b := \{\mathbf{C} : \mathbf{C} \notin \mathcal{U}\ O(\mathbf{C}) = b\}$ 表示输出为 b 的稳定配置集合。与标准协议（不含数据）的情形一样，[1], 很容易看出$\mathcal{U}$ 是向上封闭的。此外，由于 $\sqsubseteq$ 是良拟序，因此$\mathcal{U}$ 有有限基。

这使我们能够将截断的概念从 [1]推广开来。某种形式 $\boldsymbol{f}$的状态截断 到 $k \geq 1$，记为 $\tau_k(\boldsymbol{f})$，是满足以下条件的形式：对于所有 $q \in Q$，$\tau_k(\boldsymbol{f})(q) := \min(\boldsymbol{f}(q)\ k)$。状态截断的概念也推广到了配置：$\tau_k(\mathbf{C})$ 是满足以下条件的配置：对于所有 $d \in \mathbb{D}$，$\tau_k(\mathbf{C})(d) := \tau_k(\mathbf{C}(d))$。当阈值足够大时，配置的稳定性和输出在状态截断后保持不变：

▶**引理 14.** 设 ψ 是由一个具有无序数据的群体协议计算的谓词。设 S_b 为该协议输出为 b 的稳定配置集合。存在 $k \geq 1$，使得对于所有 $b \in \{0,1\}$，有 $\mathbf{C} \in S_b$ 当且仅当 $\tau_k(\mathbf{C}) \in S_b$。

▶**引理 15.** 设 ψ 是由一个具有无序数据的即时观察协议计算的谓词。设 k 为引理 *14* 给出的阈值，且$\mathbf{C}_0$ 是一个初始配置，使得对于某个 $d \in \mathbb{D}$ 和某个 $q \in I$，有 $\mathbf{C}_0(d, q) \geq \ell = |Q| \cdot (k-1) + 1$。那么，$\psi(\mathbf{C}_0 + \boldsymbol{q}_d) = \psi(\mathbf{C}_0)$。

给定一个配置 $\mathbf{C}$和一种形式 $\boldsymbol{f}$，定义 $\#\boldsymbol{f}(\mathbf{C}) := |\{d \in \mathbb{D} : \mathbf{C}(d) = \boldsymbol{f}\}|$。由于即时观察协议的性质，对于配置 $\mathbf{C}$中存在的一种颜色为 d的形式 $\boldsymbol{f}$，总是可以将 $\boldsymbol{f}$ 复制一份并赋予新颜色 d'，使后者模仿前者的行为。

▶ **引理 16.** 设 $\mathbf{C}$ 和 $\mathbf{C}'$ 为配置，满足 $\mathbf{C} \xrightarrow{*} \mathbf{C}'$。对于每个 $d \in \mathrm{supp}(\mathbf{C})$和 $d' \in \mathbb{D} \setminus \mathrm{supp}(\mathbf{C})$，都有 $\mathbf{C} + (\mathbf{C}(d))_{d'} \xrightarrow{*} \mathbf{C}' + (\mathbf{C}'(d))_{d'}$。

结合 $\mathcal{U}$ 具有有限基这一事实，我们可以证明：当达到某个阈值后，用新颜色复制形式不会改变群体的输出。

▶**引理 17。** 设 ψ 是由具有无序数据的群体协议计算的谓词。设 $\boldsymbol{f}$是一个满足 $\mathrm{act}(\boldsymbol{f}) \subseteq I$ 的形式。存在 $h(\boldsymbol{f}) \in \mathbb{N}$，使得对于所有初始配置$\mathbf{C}_0$以及满足 $d \in \mathbb{D} \setminus \mathrm{supp}(\mathbf{C}_0)$且 $\#\boldsymbol{f}(\mathbf{C}_0) \geq h(\boldsymbol{f})$ 的情况，均有 $\psi(\mathbf{C}_0 + \boldsymbol{f}_d) = \psi(\mathbf{C}_0)$。

配置 $\mathbf{C}$ 的形式截断 记作 $\sigma(\mathbf{C})$，它是一个（任意的）配置，满足 $\sigma(\mathbf{C}) \sqsubseteq \mathbf{C}$，并且对于每个形式 $\boldsymbol{f}$，都有 $\#\boldsymbol{f}(\sigma(\mathbf{C})) = \min(\#\boldsymbol{f}(\mathbf{C}), h(\boldsymbol{f}))$，其中 $h(\boldsymbol{f})$ 由引理 17 给出。根据引理 17，$\psi(\mathbf{C}_0)$ 成立当且仅当 $\psi(\sigma(\mathbf{C}_0))$ 成立。此外，引理 17 使我们能够证明，即时观察协议的表达能力弱于一般模型。

▶**命题 18。** 没有任何即时观察人口协议能够计算多数谓词 φ_{maj}。

证明。 反设，存在某个 IO协议计算 φ_{maj}。令 q_I为唯一的初始状态，并令 $\boldsymbol{f} := \{\{q_I\}\}$。令 $h(\boldsymbol{f})$ 由引理 17 给出。令 $\mathbf{C}_0$ 为一个初始配置，使得

- $\mathbf{C}_0(d) = \sum_{i=1}^{h(\boldsymbol{f})+1} \boldsymbol{f}$ 对唯一的数据项 $d \in \mathbb{D}$成立，并且$\mathbf{C}_0(d') = \boldsymbol{f}$恰
- 好对其他 $h(\boldsymbol{f})$ 个数据 $d' \in \{d_1, d_2, \ldots, d_{h(\boldsymbol{f})}\}$成立。

由于群体中有 $h(\boldsymbol{f})+1$ 个代理，而群体总数为 $2 \cdot h(\boldsymbol{f})+1$ 个代理，因此 $\varphi_{\mathrm{maj}}(\mathbf{C}_0) = $ 真。令 $\mathbf{C}_0'$ 为由 $\mathbf{C}_0$ 添加一个数据项 $d^* \notin \mathrm{supp}(\mathbf{C}_0)$ 得到的初始配置，使得 $\mathbf{C}_0'(d^*) = \boldsymbol{f}$。根据引理 17，$\varphi_{\mathrm{maj}}(\mathbf{C}_0') = \varphi_{\mathrm{maj}}(\mathbf{C}_0) = $ 真。然而，数据项 d 在 $\mathbf{C}_0'$ 中不再占多数，这就产生了矛盾。◀

4.2 由即时观察协议计算的谓词是区间谓词

▶ **定理 19。** 设 (Q, δ, I, O) 是一个带有无序数据、计算谓词 ψ 的即时观察协议。谓词 ψ 可以表示为区间谓词。

证明。 我们将把 ψ 表示为简单区间谓词的有限布尔组合。

令 h 为引理 17 给出的映射。令 $\mathbb{T} := \{\mathsf{C} : \psi(\mathsf{C}) = \text{真}\}$，以及 $\mathbb{T}_1 := \{\mathsf{C} \in \mathbb{T} : \mathsf{C}(d, q) \leq \ell$ 对所有 $d \in \mathbb{D}, q \in Q\}$。由引理 15 可知，在初始配置上进行状态截断不会改变协议所达成的共识。因此，$\psi(\mathsf{C})$ 成立，当且仅当 $\bigvee_{\mathsf{C}' \in \mathbb{T}_1} \tau_\ell(\mathsf{C}) = \mathsf{C}'$成立。令 $\mathbb{T}_2 := \{\mathsf{C} \in \mathbb{T}_1 : \#\boldsymbol{f}(\mathsf{C}) \leq h(\boldsymbol{f})$ 对所有 $\boldsymbol{f} \in \mathbb{F}\}$。由引理 17 可得，$\psi(\mathsf{C})$ 成立，当且仅当 $\bigvee_{\mathsf{C}' \in \mathbb{T}_2} \sigma(\tau_\ell(\mathsf{C})) = \mathsf{C}'$ 成立。

后者是一个无限析取。让我们将其化为有限形式。注意，若 $\mathsf{C} \xrightarrow{*} \mathsf{C}'$ 且 $\mathsf{C} \equiv \mathsf{C}$ 成立，则存在 $\mathsf{C}' \equiv \mathsf{C}'$ 使得 $\mathsf{C} \xrightarrow{*} \mathsf{C}'$。此外，等价配置具有相同的输出，因为它们共享相同的活跃状态。事实上，$\mathsf{C} \equiv \mathsf{C}$当且仅当$\bigwedge_{\boldsymbol{f} \in \mathbb{F}} \#\boldsymbol{f}(\mathsf{C}) = \#\boldsymbol{f}(\mathsf{C})$。因此，对于每个初始配置 $\mathsf{C} \equiv \mathsf{C}$，都有 $\psi(\mathsf{C}) = \psi(\mathsf{C})$。令 $\mathbb{T}_2/\equiv$ 为 $\equiv$ 在 $\mathbb{T}_2$上的所有等价类所构成的集合，并令 $\mathbb{T}_3$ 为一个集合，其中每个 $\mathbb{T}_2/\equiv$的等价类都包含一个代表配置。显然，$\psi(\mathsf{C})$成立，当且仅当$\bigvee_{\mathsf{C}' \in \mathbb{T}_3} \sigma(\tau_\ell(\mathsf{C})) \equiv \mathsf{C}'$ 成立。

下面论证 $\mathbb{T}_3$ 是有限的。令 $\mathbb{F}_\ell := \{\boldsymbol{f} \neq \mathbf{0} : \boldsymbol{f}(q) \leq \ell$ 对所有 $q \in Q\}$。对于每个配置$\mathbf{C} \in \mathbb{T}_1$，每个满足$\#\boldsymbol{f}(\mathbf{C}) > 0$ 的形式 $\boldsymbol{f}$ 都属于 $\mathbb{F}_\ell$。由于 $\mathbb{T}_2 \subseteq \mathbb{T}_1$，这对 $\mathbb{T}_2$中的配置也成立。给定$\mathbf{C} \in \mathbb{T}_2$，对于所有 $\boldsymbol{f} \in \mathbb{F}_\ell$，都有$\#\boldsymbol{f}(\mathbf{C}) \leq h(\boldsymbol{f})$；对于所有 $\boldsymbol{f} \notin \mathbb{F}_\ell$，都有$\#\boldsymbol{f}(\mathbf{C}) = 0$。因此，由于 $\mathbb{F}_\ell$ 是有限的，我们得出 $\mathbb{T}_3$ 也是有限的。

现在让我们利用这些观察结果，将 ψ 表示为一个区间谓词。固定某个$\mathbf{C}' \in \mathbb{T}_3$。只需说明如何表示“$\sigma(\tau_\ell(\mathbf{C})) \equiv \mathbf{C}'$”。事实上，由于 $\mathbb{T}_3$ 是有限的，我们可以取有限析取式 $\bigvee_{\mathbf{C}' \in \mathbb{T}_3} \sigma(\tau_\ell(\mathbf{C})) \equiv \mathbf{C}'$，从而得出结论。

对于每个形式 $\boldsymbol{f} \in \mathbb{F}_\ell$，令 $\mathsf{lt}(\boldsymbol{f}) \coloneqq \{q \in Q\colon \boldsymbol{f}(q) < \ell\}$，$\mathsf{eq}(\boldsymbol{f}) \coloneqq \{q \in Q\colon \boldsymbol{f}(q) = \ell\}$ 以及

$$\varphi_{\boldsymbol{f},d}(\mathbf{C}) \coloneqq \bigwedge_{q \in \mathsf{lt}(\boldsymbol{f})} (\mathbf{C}(d,q) = \boldsymbol{f}(q)) \wedge \bigwedge_{q \in \mathsf{eq}(\boldsymbol{f})} (\mathbf{C}(d,q) \geq \boldsymbol{f}(q)).$$

注意，$\varphi_{\boldsymbol{f},\,d}(\mathbf{C})$ 成立，当且仅当 $\tau_\ell(\mathbf{C})(d) = \boldsymbol{f}$。

对于每个满足 $\#\boldsymbol{f}(\mathbf{C}') < h(\boldsymbol{f})$ 的 $\boldsymbol{f} \in \mathbb{F}_\ell$，我们定义这个公式，其中 $n \coloneqq \#\boldsymbol{f}(\mathbf{C}')$：

$$\psi_{\boldsymbol{f}}(\mathbf{C}) \coloneqq \dot{\exists} d_1, d_2, \ldots, d_n \in \mathbb{D} : \bigwedge_{i=1}^{n} \varphi_{\boldsymbol{f},d_i}(\mathbf{C}) \wedge \neg \dot{\exists} d_1, d_2, \ldots, d_{n+1} \in \mathbb{D} : \bigwedge_{i=1}^{n+1} \varphi_{\boldsymbol{f},d_i}(\mathbf{C}).$$

对于每个满足 $\#\boldsymbol{f}(\mathbf{C}') = h(\boldsymbol{f})$ 的 $\boldsymbol{f} \in \mathbb{F}_\ell$，我们定义这个公式，其中 $n \coloneqq \#\boldsymbol{f}(\mathbf{C}')$：

$$\psi_{\boldsymbol{f}}(\mathbf{C}) \coloneqq \dot{\exists} d_1, d_2, \ldots, d_n \in \mathbb{D} : \bigwedge_{i=1}^{n} \varphi_{\boldsymbol{f},d_i}(\mathbf{C}).$$

注意，$\psi_{\boldsymbol{f}}$ 要么是简单区间谓词，要么是两个简单区间谓词的布尔组合。注意，$\psi_{\boldsymbol{f}}(\mathbf{C})$ 成立，当且仅当 $\#\boldsymbol{f}(\sigma(\tau_\ell(\mathbf{C}))) = \#\boldsymbol{f}(\mathbf{C}')$ 成立。这意味着，$\bigwedge_{\boldsymbol{f} \in \mathbb{F}_\ell} \psi_{\boldsymbol{f}}(\mathbf{C})$ 成立，当且仅当 $\sigma(\tau_\ell(\mathbf{C})) \equiv \mathbf{C}'$ 成立，因此证毕。◀

4.3 简单区间谓词的 IO协议

由于布尔组合可以实现（见第 2.1 节末尾），因此只需描述一个用于形式为 (2) 的简单区间谓词的协议。我们将每个 $i \in [1..n]$ 称为一个角色。在接下来的状态集合 Q中，我们为每个 $q \in Q$ 关联一个元素 $\mathsf{elem}(q) \in [1..m]$。每个代理的元素通过输入设置；例如，从符号 $(\quad x_1)$ 映射而来的代理最初处于状态 q，使得 $\mathsf{elem}(q) = 1$。令 $Q_j \coloneqq \{q \in Q : \mathsf{elem}(q) = j\}$。对于一次执行中的任意两个配置 C_a 和 C_b、任意数据项 $d \in \mathbb{D}$ 以及任意元素 $j \in [1..m]$，不变量 $\mathsf{C}_a(d, Q_j) = \mathsf{C}_b(d, Q_j)$ 成立。若对于所有 $j \in [1..m]$，$\mathsf{C}(d, Q_j) \in T(i, j)$ 都成立，则称 $d \in \mathbb{D}$ 在配置 C 中匹配角色 i。令 $r \coloneqq \max(r_1, \ldots, r_n) + 1$，其中

$$r_i \coloneqq \max(\{\min T(i,j) : j \in [1..m]\} \cup \{\max T(i,j) : j \in [1..m], \sup T(i,j) < \infty\}).$$

代理无需计数超过值 r，即可判断某个角色是否匹配。

与多数协议类似，我们的简单区间协议分阶段运行，每个阶段都是确保后续阶段满足各项性质和不变量所必需的。在选举阶段，为群体选出一名唯一的控制器，并为支撑集中的每个数据项选出一名领导者；前者负责向后者分配一组角色。

所有代理都会在计数阶段为其不可变元素 j 的计数作出贡献。此过程采用 [2]，中描述的“塔方法”：同一数据项、元素和值相同的两名代理相遇后，其中一名代理的值便会递增。按此方式计算出的最大值随后会传达给（唯一的）数据领导者。

各领导者为其对应数据项中的每个元素持有正确计数后，便会在分配阶段承担与之匹配的角色。只要满足要求，这些角色就可以通过向控制器查询而与其他角色互换。控制器会持续收到已选角色的通知，并据此更新其任务列表。

如果控制器获得了一个已完全分配的任务列表，它就会在整个群体中传播真输出，这一过程我们称为输出传播阶段。如果无法做到这一点，领导者就会针对角色分配持续进行一致的试错，最终未能完全填满任务列表，从而使控制器可以自由传播*false*输出。

▶ **例20.** 考虑 $n = m = 2$，其中 $T(1,1) := [2..\infty)$，$T(1,2) := [0..4]$，$T(2,1) := \mathbb{N}$，$T(2,2) := [1..\infty)$。令 $\mathbf{M} := \{\{(\quad x_1), (\quad x_1), (\quad x_1), (\quad x_2), (\quad x_1), (\quad x_2)\}\}$。注意，$= 5$。数据项“ ”可以匹配角色 1 和 2，“ ”无法匹配任何角色，而“ ”可以匹配角色 2。

执行该协议一段时间后，我们最终可能会得到下表所示的配置。第三、第四、第五和第六个代理所含的数据项和元素的值均正确：$\mathbf{M}(\quad x_1) = 3$ 和 $\mathbf{M}(\quad x_2) = \mathbf{M}(\quad x_1) = \mathbf{M}(\quad x_2) = 1$。第二个代理已被选为控制器。最后三个代理已分别被选为各自数据项的领导者，并已收集到每个元素的正确计数。-领导者或-领导者（也可能两者都是）已通知控制器，它们扮演角色 2。

输入	val	lead	ctrl	role	[#x_1，#x_2]的计数	角色 1、角色 2] 的任务列表 [
(x_1)	1					
(x_1)	2		✔			[✘, √]
(x_1)	**3**					
(x_2)	**1**	✔		2	[3, 1]	
(x_1)	**1**	✔			[1, 0]	
(x_2)	**1**	✔		2	[0, 1]	

-领导者可能改变主意，在注意到控制器没有分配其任务 1 后，决定担任角色 1。这会将其角色切换为 -2。-领导者通知控制器后，其角色被设为 0，而控制器（对此存疑）认为角色 2 不再处于已分配状态。随后，-领导者将其角色改为 1。最终，-领导者和-领导者通知控制器，角色 1 和 2 已被占用。以下三个快照总结了这一过程：

输入 ⋯	role ⋯	task	⋯ role ⋯	task	⋯ role ⋯	task
(x_1)						
(x_1)		[✘, p]		[✘, p]		[✔, √]
(x_1)						
(x_2)	-2		0		1	
(x_1						
(x_2)	2		2		2	

◀

请注意，虽然我们依靠阶段来描述我们的协议，但该模型的分布式特性意味着某些阶段可能会相互干扰。因此，我们在此列出一些潜在问题，以及我们的协议解决这些问题的方式。

- 虽然领导者选举很简单，但领导者的角色分配可能在实际选出领导者之前的任何时候发生。这可能导致控制器收到一项角色分配通知，但当前并未分配领导者。因此，当代理失去领导状态时，它会将自己的角色恢复为负值，这意味着在恢复为被动值之前，它必须先将这一变化通知控制器。
- 领导者可能会在计数正确之前就承担某个角色。我们提供了一种重置机制，使领导者进入“负角色”状态。这样会促使它联系控制器并纠正这一情况。
- 领导者可能此前已经承担了某个角色，但后来才意识到该角色实际上并不符合要求。此时，领导者不得不向控制器传达自己的错误。然而，它所通知的控制器最终未必是群体的控制器。因此，在失去控制器状态后，代理必须进入负控制器状态，这意味着它必须先重置控制器的任务，然后才能恢复为被动值。

- 可能存在多个具有相同角色的领导者。为防止死锁，如果领导者发现控制器尚未填充该任务，则允许其自行重新分配到一个新角色。

4.3.1 状态

状态集合定义为 $Q := \{\text{假}, \text{真}\}^{n+2} \times [1..m] \times [1..r]^{m+1} \times [-n..n] \times \{-1, 0, 1\}$。为了提高可读性，我们使用以下宏来指定和操作状态：

$name$	$q \in Q$ 的值	$q \in I$ 的值
元素(q)	$[1..m]$	$j \in [1..m]$
值(q)	$[1..r]$	1
输出(q)	*{假, 真}*	假
领导者(q)	*{假, 真}*	$true$
角色(q)	$[-n..n]$	0
计数$_\ell$(q)	$[1..r]$	如果为 $\ell = j$，则为 1；否则为 0。
控制器(q)	*{−1, 0, 1}*	1
任务$_i$(q)	*{false, true}*	*false*

输入映射定义为 $\iota(x_j) := p_j$，其中 p_j 是 I 中唯一满足 elem(p_j) $= j$的状态。通俗地说，elem(q) $= j$ 表示代理持有第 j个元素；val(q) 是该代理所对应数据项中元素 elem(q) 当前的计数；lead(q) 和 ctrl(q) 分别表示代理是否为数据领导者或控制器；role(q) 表示领导者的角色；count$_j$(q) 使数据领导者能够维护目前观察到的元素 j的最高计数；task$_i$(q) 使控制器能够维护当前已匹配角色的列表；out(q) 则表示代理对协议输出的当前判断。

请注意，本节给出的规则用于简洁地描述转移。单条规则可能会导出多个转移。此外，为求简洁，我们在允许镜像转移 的规则编号旁标上星号（$*$）。镜像转移是指代理可以观察自身状态并据此作出反应的转移。因此，生成前置条件公式为 $A(p) \wedge B(q)$ 的 $*$-规则，也会生成前置条件为 $A(q) \wedge B(q)$ 的转移，从而使状态 p 成为任意“虚拟代理”的状态。请注意，仍然只有 q 这一状态会被更新为 q'。

4.3.2 领导者和控制器选举

前两条规则旨在为群体中存在的每个数据项选出唯一的领导者，并为整个群体选出唯一的全局控制器。在本节余下部分中，我们固定一个公平执行 $\mathbf{C}_0\mathbf{C}_1\cdots$，其中 $\mathbf{C}_0$ 为初始状态。

$rule$	状态前置条件	颜色前置条件	状态更新
(1)	领导者(p) 领导者(q)	$d_1 = d_2$	角色(q') $= -\lvert$角色(q)$\rvert$ ¬领导者(q')
(2)	控制器(p) $= 1$ 控制器(q) $= 1$	$none$	控制器(q') $= -1$

请注意，规则 (1) 保证失去领导权的代理将其角色设为非正值。同样，规则 (2) 将非控制器推入其控制器值的临时中间状态，即 -1。令 $Q_L := \{q \in Q : \text{领导者}(q)\}$ 和 $Q_C := \{q \in Q : \text{控制器}(q) = 1\}$。以下引理指出了两次选举的结束时刻。

▶ **引理 21.** 存在 $\tau \in \mathbb{N}$ 使得 $|\mathbf{C}_\tau|_{Q_C} = |\mathbf{C}_{\tau+1}|_{Q_C} = \cdots = 1$，并且 $|\mathbf{C}_\tau(d)|_{Q_L} = |\mathbf{C}_{\tau+1}(d)|_{Q_L} = \cdots = 1$ 对每个 $d \in \mathbb{D}$ 都成立。

令 α 表示引理 21 给出的最小值 τ，我们将其称为选举阶段的结束。

4.3.3 按数据项计数元素

接下来的规则允许统计持有相同元素且具有共同数据项的代理数量。最终会将此计数传达给数据领导者。给定 $d \in \mathbb{D}$ 和 $\tau \in \mathbb{N}$，若状态 $q \in Q$ 满足 $\text{elem}(q) = j$ 且 $\text{val}(q) = \min(\mathbf{C}_\tau(d, Q_j)\ r)$，则称其为 (d, j)-有效。

rule	状态前置条件	颜色前置条件	状态更新
(3)	元素(p) = 元素(q) 值(q) = 值(p) < r	$d_1 = d_2$	值(q') = 值$(q)+1$
(4)*	计数$_{\text{elem}(p)}(q)$ <值(p) 领导者(q)	$d_1 = d_2$	计数$_{\text{elem}(p)}(q')$ = 值(p) `if` (角色$(q) > 0 \wedge$ 值$(p) \notin T$(角色(q), 元素(p))): 角色(q') = −角色(q)

观察另一种纠正机制；规则（4）保证，具有分配的角色 $i > 0$ 的领导者在更新其计数后，会验证自己是否仍能担任角色 i。以下引理说明，最终会向每个数据项领导者提供正确的计数。

▶ **引理 22.** 存在 $\tau \in \mathbb{N}$，使得对于每个 $\tau' \geq \tau$，$d \in \text{supp}(\mathbf{C}_{\tau'})$ 和 $j \in [1..m]$，如果 $\mathbf{C}_0(d, Q_j) > 0$，那么对某个 (d, j)-有效状态 q，$\mathbf{C}_{\tau'}(d, q) > 0$ 成立。

▶ **引理 23.** 存在 $\tau \geq \alpha$，使得对于每个 $\tau' \geq \tau$、$d \in \text{supp}(\mathbf{C}_{\tau'})$、$j \in [1..m]$ 和 $q \in \text{act}(\mathbf{C}_{\tau'}(d)) \cap Q_L$，都有 计数$_j(q)$ = $\min(\mathbf{C}_{\tau'}(d, Q_j)\ r)$。

令 τ' 和 τ'' 表示引理 22 和 23 给出的最小值 τ。从现在起，令 $\beta := \max(\tau', \tau'')$。

4.3.4 角色分布与任务跟踪

以下规则为领导者分配角色，并允许领导者在可能时重置其角色，从而避免死锁。在规则（5）中，变量 i 可以取 $[1..n]$中的任意值。

rule	状态前置条件	颜色前置条件	状态更新
(5)	领导者(q) 角色$(q) = 0$ $\bigwedge_{j \in [1..m]}$ 计数$_j(q) \in T(i, j)$	*none*	角色$(q') = i$
(6)*	控制器(p) 领导者(q) 角色$(q) = i > 0$ $\bigvee_{i' \in [1..n] \setminus \{i\}}$ $\big(\neg$任务$_{i'}(p) \wedge$ $\bigwedge_{j \in [1..m]}$ 计数$_j(q) \in T(i', j)\big)$	*none*	角色$(q') = -i$

这导出了以下结果，非正式地说，意味着如果某个领导者已经担任了某个角色，那么它目前认为自己可以担任该角色。

16 带无序数据的群体协议

▶ **引理 24.** 对于每个 $\tau \in \mathbb{N}$、 $j \in [1..m]$ 和 $q \in \mathrm{act}(\mathbf{C}_\tau) \cap Q_L$ ， 只要 $\mathrm{role}(q) > 0$，就有 $\mathrm{count}_j(q) \in T(\mathrm{role}(q)\ j)$。

这些规则允许在需要时更新控制器的任务列表并重置角色：

rule	状态前置条件	颜色前置条件	状态更新
(7)*	角色$(p) \neq 0$ 控制器$(q) = 1$	*none*	任务$_{\lvert\mathrm{role}(p)\rvert}(q') =$ (角色$(p) > 0$)
(8)*	控制器$(p) = 1$ 角色$(q) < 0$ ¬任务$_{\lvert\mathrm{role}(q)\rvert}(p)$	*none*	角色$(q') = 0$

为了说明规则 (5) 至 (8) 的运行方式，我们给出以下示例。

▶ **示例 25.** 回顾前文介绍的例20。考虑其第一个快照的配置。虽然我们最初直观地说明了如何从这一特定配置中进行角色重新分配，但这里将对这一过程中涉及的重要配置进行更深入的分析。

输入	val	lead	ctrl	role	[#x_1 的计数,#x_2]	[角色1、角色 2]的任务列表
(x_1)	1					
(x_1)	2		✔			[✘, √]
(x_1)	**3**					
(x_2)	**1**	✔		2	[3, 1]	
(x_1)	**1**	✔			[1, 0]	
(x_2)	**1**	✔		2	[0, 1]	

如上所示，-领导者目前认为（而且确实如此）它可以担任角色 1 和 2。注意，控制器已分配任务 2，但任务 1 仍未分配。因此，规则（6）允许 -领导者通过与控制器交互，将其角色设为 -2，从而启动重新分配。由此得到以下配置：

输入	val	lead	ctrl	role	计数 [#x_1，#x_2]	[的任务列表：角色 1，角色 2]
(x_1)	1					
(x_1)	2		✔			[✘, √]
(x_1)	**3**					
(x_2)	**1**	✔		-2	[3, 1]	
(x_1)	**1**	✔			[1, 0]	
(x_2)	**1**	✔		2	[0, 1]	

由于其角色被设为 -2，-领导者现在试图告知控制器，应从其任务列表中取消分配角色 2。下一次会面时，他们通过规则（7）完成此操作。于是得到如下配置：

输入	val	lead	ctrl	role	计数 [#x_1，#x_2]	角色 1、角色 2]的任务列表
(x_1)	1					
(x_1)	2		✔			[✘, p]
(x_1)	**3**					
(x_2)	**1**	✔		-2	[3, 1]	
(x_1)	**1**	✔			[1, 0]	
(x_2)	**1**	✔		2	[0, 1]	

请注意，这并不意味着当前没有领导者的角色被设为 2；事实上，-领导者的角色仍被设为 2。现在假设，在达到此配置后，-领导者和控制器立即再次会面。由于 -领导者观察到控制器不再分配任务 2，因此可以推断，要么是它取消了该任务的分配，要么是其他领导者取消了分配，要么该任务从未被分配。无论如何，它都可以通过规则（8）安全地将自己的角色重置为 0，从而得到如下配置：

输入	val	lead	ctrl	role	[#x_1 的计数，#x_2]	[角色 1、角色 2] 的任务列表
(x_1)	1					
(x_1)	2		✔			[✖, p]
(x_1)	**3**					
(x_2)	**1**	✔		0	[3, 1]	
(x_1)	**1**	✔			[1, 0]	
(x_2)	**1**	✔		2	[0, 1]	

注意，-领导者可能在控制器之前遇到该领导者，从而重新分配控制器任务列表中的角色 2，并撤销该领导者所做的工作。这只会延迟该领导者重置角色；在公平性的作用下，不会无限期地阻止这一过程。

从这个最后的配置来看，由于-领导者的角色被设为 0，现在它可以通过规则（5）担任自己能够胜任的任何角色。为简洁起见，假设它担任角色 1：

输入	val	lead	ctrl	role	计数 [#x_1,#x_2]	[角色 1、角色 2] 的任务列表
(x_1)	1					
(x_1)	2		✔			[✖, p]
(x_1)	**3**					
(x_2)	**1**	✔		1	[3, 1]	
(x_1)	**1**	✔			[1, 0]	
(x_2)	**1**	✔		2	[0, 1]	

假设 -领导者 在 -领导者 之前遇到了控制器。那么，规则（7）会将任务 2 分配到控制器的任务列表中。由于控制器已经分配了任务 2，-领导者 无法再通过规则（6）重置其角色。因此，当 -领导者 最终再次遇到控制器时，它终于通过规则（7）将任务 1 分配到自己的任务列表中。

输入	val	lead	ctrl	role	[#x_1 的计数,#x_2]	[角色 1 的任务列表，角色 2]
(x_1	1					
(x_1)	2		✔			[✔, √]
(x_1)	**3**					
(x_2)	**1**	✔		1	[3, 1]	
(x_1)	**1**	✔			[1, 0]	
(x_2)	**1**	✔		2	[0, 1]	

以下引理表明，在执行过程中的某个时刻，会达到这样一种配置：既不是领导者也不是控制器的代理不再与其他代理交互。

▶ **引理 26.** 存在某个 $\tau \in \mathbb{N}$，使得对于每个 $\tau' \geq \tau$ 和 $q \in$ 活动状态集合$(\mathbf{C}_{\tau'}) \mid Q_L$，都有角色$(q) = 0$。

rule	状态前置条件	颜色前置条件	状态更新
(9)	控制器$(p) = -1$ 控制器$(q) = 1$	*none*	$\bigwedge_{i\in[1..m]} \neg$任务$_i(q')$
(10)	控制器$(p) = 1$ 控制器$(q) = -1$ $\bigwedge_{i\in[1..m]} \neg$任务$_i(p)$	*none*	控制器$(q') = 0$

▶ **引理 27.** 存在 $\tau \geq \alpha$，使得对于每个 $\tau' \geq \tau$ 和 $q \in$ 活动状态集合$(\mathbf{C}_{\tau'})$，都有 控制器$(q) \in \{0,1\}$。

令 τ' 和 τ'' 表示引理 26 和 27 给出的最小值 τ。从现在起，令$\gamma := \max(\beta, \tau, \tau', \tau'')$。直观地说，这限定了不存在负控制器的配置，从而通过规则 (9) 防止控制器的任务被反复重置。最后，下面的引理论证了，在 $\mathbf{C}_\gamma$ 之后，只有当某个领导者当前承担相应角色（无论是正角色还是负角色）时，控制器才能有任务被设置为真。

▶ **引理 28.** 对于每个 $\tau \geq \gamma$、$i \in [1..n]$ 和 $q \in$ 活动状态集合$(\mathbf{C}_\tau) \cap Q_C$，只要 任务$_i(q)$ 成立，就存在 $q' \in$ 活动状态集合$(\mathbf{C}_\tau)$，使得 |角色$(q')| = i$。

4.3.5 输出传播

最后一条规则允许控制器向其他代理传达其任务列表当前是否已全部分配。注意，状态 q 的输出恰好是 out(q) 的值，即 $O(q) := \text{out}(q)$。

rule	状态前置条件	颜色前置条件	状态更新
(11)*	控制器(p)	*none*	输出$(q') = \bigwedge_{i=1}^{n}$ 任务$_i(p)$

▶ **引理 29。** 当且仅当存在某个 $\tau \geq \gamma$，使得对于每个$\tau' \geq \tau$，存在 $q \in$活动状态集合$(\mathbf{C}_{\tau'}) \cap Q_C$，使得 $\bigwedge_{i\in[1..n]}$任务$_i(q)$ 成立时，$\psi(\mathbf{C}_0)$ 成立。

▶ **推论30。** 存在 $\tau \in \mathbb{N}$，使得 $O(\mathbf{C}_\tau) = O(\mathbf{C}_{\tau+1}) = \cdots = \psi(\mathbf{C}_0)$。

5 结论

本文介绍了无序数据群体协议；提出了一种可在无限数据域上计算多数的此类协议；并确定了即时观察协议的表达能力：它们能够计算区间谓词。

这项工作开启了对在任意大的域上运行的群体协议的研究。因此，它为许多令人振奋的问题打开了大门，例如 如何设计空间高效和时间高效的协议。特别是，我们模型的表达能力仍有待确定。

关于数据多重集上的逻辑，已有一些研究结果（例如，见 [22, 24]）。具体而言，[22] 的作者提出了一种可判定的逻辑，其形式与 Presburger 算术相似。无序数据群体协议似乎能够计算这种逻辑（或许能够精确刻画）。虽然我们相当确信，在我们的模型中可以计算与数据计数相关的余数谓词和阈值谓词，但存在量化似乎比简单区间谓词中的存在量化更难实现；前者出现在 [22]，的（无歧义）已解形式中。

我们的模型还与数据词上的逻辑和自动机有关：带数据的协议的输入可以看作数据词，其中 Q 是字母表， $\mathbb{D}$ 是数据域。重要的是，这些数据词具有交换性，也就是说，置换不会改变接受结果。例如， [8] 的逻辑 $\mathrm{FO}^2(+1, \sim, <)$ 可以刻画非交换性质，例如“存在一个由 a 组成的块，后面紧跟一个由 b 组成的块”。就这一点而言，这种逻辑过于“强”。它也过于“弱”，因为它无法表达“对于每个数据项， a 的数量都是偶数”。出于同样的原因，$\mathrm{EMSO}^2(+1, \sim)$ 及其等价形式弱数据自动机 [21]，也过于“弱”。逻辑 $\mathrm{EMSO}^2_{\#}(+1, \sim)$ 及其等价形式交换数据自动机 [25], 可以表达后一性质，但后继关系同样允许表达字母上的非交换性质。因此，尽管与数据词相关的模型已有研究，且可能为完整刻画我们模型的表达能力提供启发，我们尚未将它们与自己的模型直接联系起来。

参考文献

1 Dana Angluin、James Aspnes、Zoë Diamadi、Michael J. Fischer 和 René Peralta。被动移动有限状态传感器网络中的计算。《分布式计算》，18(4):235–253，2006。2 Dana Angluin、James Aspnes、David Eisenstat 和 Eric Ruppert。群体协议的计算能力。《分布式计算》，20(4):279–304，2007。3 Gregor Bankhamer、Petra Berenbrink、Felix Biermeier、Robert Elsässer、Hamed Hosseinpour、Dominik Kaaser 和 Peter Kling。用于精确相对多数共识的群体协议：失败概率虽小，却有助于消除无关紧要的意见。载于第 41st 届ACM分布式计算原理研讨会（PODC）论文集，第224–234页，2022年。4 Petra Berenbrink、Felix Biermeier、Christopher Hahn 和 Dominik Kaaser。松散稳定的阶段时钟与自适应多数问题。载于 1st 动态网络算法基础研讨会（SAND）论文集，第7:1–7:17页，2022年。5 Petra Berenbrink、Robert Elsässer、Tom Friedetzky、Dominik Kaaser、Peter Kling 和 Tomasz Radzik。一个具有 $O(\log^{5/3} n)$稳定时间和$\Theta(\log n)$个状态的精确多数群体协议。载于 32nd 国际分布式计算研讨会（DISC）论文集，第10:1–10:18页，2018年。6 Michael Blondin、Javier Esparza、Blaise Genest、Martin Helfrich 和 Stefan Jaax。用于 Presburger 算术的简洁群体协议。载于 37th 国际计算机科学理论方面研讨会（STACS）论文集，第40:1–40:15页，2020年。7 Michael Blondin、Javier Esparza、Stefan Jaax 和 Philipp J. Meyer。迈向群体协议的高效验证。《系统设计中的形式化方法》（FMSD），57(3):305–342，2021年。8 Mikołaj Bojańczyk、Claire David、Anca Muscholl、Thomas Schwentick 和 Luc Segoufin。数据词上的二变量逻辑。《ACM计算逻辑汇刊》（TOCL），12(4):27:1–27:26，2011年。9 Benedikt Bollig、Patricia Bouyer 和 Fabian Reiter。寄存器中的标识符——用逻辑描述网络算法。载于 22nd 国际软件科学与计算结构基础会议（FoSSaCS）论文集，第115–132页，2019年。10 Benedikt Bollig、Fedor Ryabinin 和 Arnaud Sangnier。分布式存储器自动机中的可达性。载于 29th EACSL计算机科学逻辑年会（CSL）论文集，第13:1–13:16页，2021年。11 Robert S. Boyer 和 J. Strother Moore。Mjrty：一种快速多数投票算法。载于《自动推理：纪念 Woody Bledsoe 文集》，1991年。12 Philipp Czerner、Roland Guttenberg、Martin Helfrich 和 Javier Esparza。用于 Presburger 算术的快速简洁群体协议。载于 1st 动态网络算法基础研讨会（SAND）论文集，第11:1–11:17页，2022年。

13 Giorgio Delzanno、Arnaud Sangnier 和 Riccardo Traverso。为带广播的参数化网络添加数据寄存器。Fundamenta Informaticae，143(3-4)：287–316，2016。14 David Doty、Mahsa Eftekhari、Leszek Gasieniec、Eric E. Severson、Przemyslaw Uznanski 和 Grzegorz Stachowiak。一种求解精确多数问题、在时间和空间上最优的稳定群体协议。载于： 62nd IEEE计算机科学基础年会（FOCS）论文集，第1044–1055页，2021。15 Javier Esparza、Pierre Ganty、Jérôme Leroux 和 Rupak Majumdar。群体协议验证。Acta Informatica，54(2)：191–215，2017。16 Leszek Gasieniec、David D. Hamilton、Russell Martin、Paul G. Spirakis 和 Grzegorz Stachowiak。用于精确多数和相对多数的确定性群体协议。载于： 20^{th}分布式系统原理国际会议（OPODIS）论文集，第14:1–14:14页，2016。17 Utkarsh Gupta、Preey Shah、S. Akshay 和 Piotr Hofman。无序数据佩特里网的连续可达性属于多项式时间。载于： 22nd 国际软件科学与计算结构基础会议（FoSSaCS）论文集，第260–276页，2019。18 Piotr Hofman、Slawomir Lasota、Ranko Lazic、Jérôme Leroux、Sylvain Schmitz 和 Patrick Totzke。无序数据佩特里网的覆盖树。载于： 19th 国际软件科学与计算结构基础会议（FoSSaCS）论文集，第445–461页，2016。19 Piotr Hofman、Jérôme Leroux 和 Patrick Totzke。无序数据向量的线性组合。载于： 32nd ACM/IEEE计算机科学逻辑年会（LICS）论文集，第1–11页，2017。20 Michael Kaminski 和 Nissim Francez。有限记忆自动机。Theoretical Computer Science，134(2)：329–363，1994。21 Ahmet Kara、Thomas Schwentick 和 Tony Tan。数据词上带后继关系的二变量逻辑的可行自动机。载于： 6th 语言与自动机理论及应用国际会议（LATA）论文集，第7183卷，第351–362页，2012。22 Denis Lugiez。可计数的多树自动机。Theoretical Computer Science，333(1-2)：225–263，2005。23 Othon Michail 和 Paul G. Spirakis。动态网络理论基础。Communications of the ACM，61(2)：72，2018。24 Ruzica Piskac 和 Viktor Kuncak。带基数约束的多重集判定过程。载于： 9th 验证、模型检测与抽象解释国际会议（VMCAI）论文集，第218–232页，2008。25 Zhilin Wu。交换数据自动机。载于： 26th 国际研讨会/21st EACSL计算机科学逻辑年会（CSL）论文集，第16卷，第528–542页，2012。

第3节缺失的证明

我们先在一张表中回顾所有转移:

配对阶段			
rule	状态前置条件	颜色前置条件	状态更新
(1)	¬配对(p) ¬配对(q)	$d_1 \neq d_2$	配对(p') ∧偶数(p') 配对(q') ∧偶数(q')
分组阶段			
rule	状态前置条件	颜色前置条件	状态更新
(2)	¬配对(p) 配对(q) ∧ 组(q) ∧多数(q) $= Y$	$d_1 \neq d_2$	*none* ¬组(q') ∧ 多数(q') $= N$
(3)	¬配对(p) 配对(q) ∧ ¬组(q) ∧ 多数(q) $= N$	$d_1 = d_2$	*none* 组(q') ∧ 多数(q') $= Y$
(4)	¬配对(p) 配对(q) ∧ 组(q) ∧ 多数(q) $\in \{y, n\}$	$d_1 \neq d_2$	*none* 多数(q') $= \overline{N}$
(5)	¬配对(p) 配对(q) ∧ ¬组(q) ∧ 多数(q) $\in \{y, n\}$	$d_1 = d_2$	*none* 多数(q') $= \overline{Y}$
(6)	组(p) ∧ 多数(p) $= \overline{N}$ ¬组(q) ∧ 多数(q) $\in \{y, n\}$	*none*	¬组(p') ∧ 多数(p') $= N$ 多数(q') $= N$
(7)	¬组(p) ∧ 多数(p) $= \overline{Y}$ 组(q) ∧ 多数(q) $\in \{y, n\}$	*none*	组(p') ∧ 多数(p') $= Y$ 多数(q') $= Y$
(8)	组(p) ∧ 多数(p) $= \overline{N}$ ¬组(q) ∧多数(q) $= \overline{Y}$	*none*	¬组(p') ∧多数(p') $= n$ 组(q') ∧多数(q') $= n$
多数阶段			
rule	状态前置条件	颜色前置条件	状态更新
(9)	¬配对(p) 偶数(q)	*none*	*none* ¬偶数(q')
(10)	配对(p) ∧ 偶数(p) 配对(q) ∧ ¬偶数(q)	*none*	*none* 偶数(q')
(11)	多数(p) $= Y$ 多数(q) $= N$	*none*	多数(p') $= n$ 多数(q') $= n$
(12)	多数(p) $= Y$ 多数(q) $= n$	*none*	*none* 多数(q') $= y$
(13)	多数(p) $= N$ 多数(q) $= y$	*none*	*none* 多数(q') $= n$
(14)	多数(p) $= n$ 多数(q) $= y$	*none*	*none* 多数(q') $= n$

请注意，为简洁起见，在本节余下部分中，我们使用记号$\mathbf{C} \xrightarrow{(n)} \mathbf{C}'$来表示$\mathbf{C} \xrightarrow{t} \mathbf{C}'$使用由规则（$n$）导出的转移 t 。

▶ **引理 4.** 存在 $\tau \in \mathbb{N}$，使得 $|\mathbf{C}_\tau|_U = |\mathbf{C}_{\tau+1}|_U = \cdots$。此外，对于每个$i \geq \tau$，$\mathbf{C}_i$中的所有未配对代理都具有相同的颜色，也就是说，集合 $\{d \in \mathbb{D} : \mathbf{C}_i(d, U) > 0\}$ 要么为空，要么是单元素集合。

证明。 首先请注意，没有规则会将状态从已配对变为未配对。这意味着对于任意两个 $\mathbf{C}$、$\mathbf{C}'$，满足 $\mathbf{C} \xrightarrow{*} \mathbf{C}'$，都有 $|\mathbf{C}|_U \geq |\mathbf{C}'|_U$。由此可得一个阈值$\tau \in \mathbb{N}$，满足 $|\mathbf{C}_\tau|_U = |\mathbf{C}_{\tau+1}|_U = \cdots$，因为该值不可能降至零以下。

假设存在无穷多个 $i \geq \tau$，使得 $\mathbf{C}_i(d, U) > 0$且 $\mathbf{C}_i(d', U) > 0$对两个不同的颜色 $d, d' \in \mathbb{D}$ 成立。规则 (1) 在每个这样的 $\mathbf{C}_i$中均处于启用状态。这意味着存在一个配置 $\mathbf{C}'_i$，使得 $\mathbf{C}_i \xrightarrow{*} \mathbf{C}'_i$ 且 $|\mathbf{C}_i|_U > |\mathbf{C}'_i|_U$。根据公平性，沿着执行会到达集合 $\{\mathbf{C}'_i : i \in \mathbb{N}\}$ 中的某个配置，这就产生了矛盾。◀

▶ **引理 5**。 设 $i \in \mathbb{N}$。若 $\varphi_{maj}(\mathbf{C}_0)$ 且 d 是多数颜色，则 $\mathbf{C}_i(d, U) > 0$。

证明。 由于 d 是多数颜色，且所有代理最初都未配对，因此有

$$\mathbf{C}_0(d, U) > \sum_{d' \neq d} \mathbf{C}_0(d', U). \tag{3}$$

此外，对于满足 $i > 0$ 的每个 $i > 0$，使得 $\mathbf{C}_{i-1} \xrightarrow{(1)} \mathbf{C}_i$，至多有一个状态 $q \in$ 活动状态集合$(\mathbf{C}_{i-1}(d))$ 已配对，而至少有一种颜色 $d' \neq d$ 的一个状态 $q' \in$ 活动状态集合$(\mathbf{C}_{i-1}(d'))$ 已配对。因此，我们要么有 $\mathbf{C}_i(d, U) = \mathbf{C}_{i-1}(d, U)$，要么有 $\mathbf{C}_i(d, U) = \mathbf{C}_{i-1}(d, U) - 1$，并且有

$$\sum_{d' \neq d} \mathbf{C}_i(d', U) < \sum_{d' \neq d} \mathbf{C}_{i-1}(d', U).$$

因此，根据 (3)，有 $\mathbf{C}_i(d, U) > \sum_{d' \neq d} \mathbf{C}_i(d', U) \geq 0$。

请注意，只有规则 (1) 会改变状态的配对。因此，$\mathbf{C}_i(d, U) > 0$ 对所有$i \in \mathbb{N}$均成立。◀

▶ **引理 6.** 对于每个 $i \in \mathbb{N}$，都有 $|\mathbf{C}_i|_{Q_Y} - |\mathbf{C}_i|_{Q_N} = |\mathbf{C}_i|_{Q_M} - |\mathbf{C}_i|_{Q_m}$。

证明。 我们对 $i \geq 0$ 进行归纳法来证明该结论。我们有 $|\mathbf{C}_0|_{Q_N} = |\mathbf{C}_0|_{Q_m} = 0$ 和$|\mathbf{C}_0|_{Q_Y} = |\mathbf{C}_0|_{Q_M} = |\mathbf{C}_0|$。因此，该结论在初始情况下成立。

给定一个转移 $t = ((p, q) \sim (p', q')) \in \delta$和 $x \in \{Y, N, M, m\}$，我们定义 $\Delta_x(t) := |\{\{p', q'\}\}(Q_x)| - |\{\{p, q\}\}(Q_x)|$。此表展示了每个 t 对应的 $\Delta_x(t)$ 的值：

$t \in \delta$ 来自规则 #	$\Delta_Y(t)$	$\Delta_N(t)$	$\Delta_M(t)$	$\Delta_m(t)$	$\Delta_Y(t) - \Delta_N(t)$	$\Delta_M(t) - \Delta_m(t)$
(2)	-1	1	-1	1	-2	-2
(3)	1	-1	1	-1	2	2
(6)	0	2	-1	1	-2	-2
(7)	2	0	1	-1	2	2
(11)	-1	-1	0	0	0	0
其他规则	0	0	0	0	0	0

令 $i > 0$。注意，根据定义，对于每个 $t \in \delta$，$|\mathbf{C}_i|_{Q_x} = |\mathbf{C}_{i-1}|_{Q_x} + \Delta_x(t)$ 都成立，并且

$x \in \{Y,\ N,\ M,\ n\}$。因此，

$$\begin{aligned}
|\mathbf{C}_i|_{Q_Y} - |\mathbf{C}_i|_{Q_N} &= (|\mathbf{C}_{i-1}|_{Q_Y} + \Delta_Y(t)) - (|\mathbf{C}_{i-1}|_{Q_N} + \Delta_N(t)) \\
&= \big(|\mathbf{C}_{i-1}|_{Q_Y} - |\mathbf{C}_{i-1}|_{Q_N}\big) + \big(\Delta_Y(t) - \Delta_N(t)\big) \\
&= \big(|\mathbf{C}_{i-1}|_{Q_M} - |\mathbf{C}_{i-1}|_{Q_m}\big) + \big(\Delta_Y(t) - \Delta_N(t)\big) \quad \text{(by induction hyp.)} \\
&= \big(|\mathbf{C}_{i-1}|_{Q_M} - |\mathbf{C}_{i-1}|_{Q_m}\big) + \big(\Delta_M(t) - \Delta_m(t)\big) \quad \text{(by above table)} \\
&= \big(|\mathbf{C}_{i-1}|_{Q_M} + \Delta_M(t)\big) - \big(|\mathbf{C}_{i-1}|_{Q_m} + \Delta_m(t)\big) \\
&= |\mathbf{C}_i|_{Q_M} - |\mathbf{C}_i|_{Q_m}.
\end{aligned}$$

◀

▶ **引理 8**。令 E 为参与多数计算的状态集合，即 $E := \{q \in Q :$ 多数$(q) \in \{y, n, Y, N\}\}$。令 $E_M := E \cap Q_M$ 且 $E_m := E \cap Q_m$。对于每个 $i \in \mathbb{N}$，以下结论成立：$|\mathbf{C}_i|_{E_M} = |\mathbf{C}_i|_{E_m}$。

证明。我们对 $i \in \mathbb{N}$ 进行归纳法。该断言对于 $\mathbf{C}_0$，作为唯一初始状态时显然成立，因为 q_I 满足多数$(q_I) \notin E$，这意味着 $|\mathbf{C}_0|_E = |\mathbf{C}_0|_{E_M} = |\mathbf{C}_0|_{E_m} = 0$。

令 $i > 0$，并令 t 使得 $\mathbf{C}_{i-1} \xrightarrow{t} \mathbf{C}_i$。对于 $|\mathbf{C}_i|_{E_M}$（分别地，$|\mathbf{C}_i|_{E_m}$）不同于 $|\mathbf{C}_{i-1}|_{E_M}$（分别地，$|\mathbf{C}_{i-1}|_{E_m}$），t 必须是 (6)、(7) 或 (11)。让我们看看这些转移各自的影响：

$t \in \delta$ 来自规则 #	$\lvert\mathbf{C}_i\rvert_{E_M} - \lvert\mathbf{C}_{i-1}\rvert_{E_M}$	$\lvert\mathbf{C}_i\rvert_{E_m} - \lvert\mathbf{C}_{i-1}\rvert_{E_m}$
(6)	-1	-1
(7)	-1	-1
(11)	1	1

请注意，规则（11）没有明确指定状态组 p 和 q。然而，没有任何规则会生成状态 q，使得 maj$(q) = Y$（分别是 maj$(q) = N$）且 $\neg$grp(q)（分别是 grp(q)），并且 q_I 满足 maj$(q_I) = Y \wedge$ grp(q_I)；因此，对于规则（11）的状态 p 和 q，grp(p) 和$\neg$grp(q) 隐含成立。由于对于所有转移，等式两边的效果相同，根据归纳假设，结论成立。◀

▶ **引理 9**。设 $d \in \mathbb{D}$。若 $\mathbf{C}_\alpha(d, U) > 0$，则存在某个 $\tau \geq \alpha$，使得对于所有$i \geq \tau$、$d' \in \mathbb{D}$ 和 $q \in \mathrm{act}(\mathbf{C}_i(d'))$，以下结论成立：$\mathrm{grp}(q) = (d' = d)$。

证明。我们声称存在 τ，τ'，τ''，$\tau''' \geq \alpha$ 使得

1. $\mathbf{C}_i(d, Q_{\overline{N}}) = 0$ for all $i \geq \tau$;
2. $\sum_{d' \neq d} \mathbf{C}_i(d', Q_{\overline{Y}}) = 0$ for all $i \geq \tau'$;
3. $\mathbf{C}_i(d, Q_m) = 0$ for all $i \geq \tau''$;
4. $\sum_{d' \neq d} \mathbf{C}_i(d', Q_M) = 0$ for all $i \geq \tau'''$.

在证明这些断言之前，注意到，只需取 $\max(\tau, \tau', \tau'', \tau''')$即可得到该引理。

断言 *1*。根据 α 的定义和引理 4，不存在颜色为 $d' \neq d$ 的未配对代理位于$\mathbf{C}_\alpha$ 中，以便在规则（4）中扮演相应的角色。此外，规则（4）是唯一能从 Q_N生成状态的规则。因此，对于任意 $i \geq \alpha$ 和 $\mathbf{C}_i \xrightarrow{*} \mathbf{C}$，都有 $\mathbf{C}_i(d, Q_N) \geq \mathbf{C}(d, Q_N)$。

剩下要证明值会降至零。如果只有有限多个指标 $i \geq \alpha$ 满足 $\mathbf{C}_i(d, Q_N) > 0$，那么结论成立。否则，必定有无限多个 $i \geq \alpha$ 满足 $\mathbf{C}_i(d, Q_N \cap Q_M) > 0$。事实上，规则（4）在前置条件 grp$(q)$ 下更新 maj$(q') = N$，且没有规则能从 $Q_N \cap Q_m$ 产生状态。根据引理 8，这意味着存在一个来自活动状态集合 $(\mathbf{C}_i) \cap E_m$ 的状态，使规则（6）或规则（8）能够被执行。因此，存在一个配置 $\mathbf{C}'_i$，使得 $\mathbf{C}_i \xrightarrow{*} \mathbf{C}'_i$ 且 $\mathbf{C}_i(d, Q_N) > \mathbf{C}'_i(d, Q_N)$。因此，根据公平性，对于某个 $\tau \geq \alpha$，我们得到 $\mathbf{C}_\tau(d, Q_N) = 0$。于是，$\mathbf{C}_\tau(d, Q_N) = \mathbf{C}_{\tau+1}(d, Q_N) = \cdots = 0$（因为值不会增加）。

论断 *2.* 与论断 1 类似，由于未配对代理共享颜色 d，规则（5）只对颜色为 d 的代理启用，条件为 $\mathbf{C}_\alpha$。因此，由于规则（5）是唯一能从 Q_Y 产生状态的规则，我们有

$$\sum_{d'\neq d}\mathbf{C}_i(d',Q_{\overline{Y}})\geq\sum_{d'\neq d}\mathbf{C}(d',Q_{\overline{Y}})\text{ for every }i\geq\alpha\text{ and }\mathbf{C}_i\xrightarrow{*}\mathbf{C}.$$

仍需证明值会降至零。如果仅对有限多个索引 $i\geq\alpha$ 有 $\sum_{d'\neq d}\mathbf{C}_i(d',Q_Y)>0$，那么我们就完成了。否则，必然对无限多个 $i\geq\alpha$ 有 $\mathbf{C}_i(d,Q_Y\cap Q_m)>0$。事实上，规则（5）在前置条件 $\neg\mathsf{grp}(q)$ 下更新多数$(q')=Y$，且没有任何规则会从 $Q_Y\cap Q_M$ 产生状态。根据引理 8，活动状态集合$(\mathbf{C}_i)$ 中的某个状态会启用规则（7）或规则（8）。因此，存在一个配置 $\mathbf{C}_i'$，使得 $\mathbf{C}_i\xrightarrow{*}\mathbf{C}_i'$，并且

$$\sum_{d'\neq d}\mathbf{C}_i(d',Q_{\overline{Y}})>\sum_{d'\neq d}\mathbf{C}_i'(d',Q_{\overline{Y}}).$$

因此，根据公平性，我们得到 $\sum_{d'\neq d}\mathbf{C}_{\tau'}(d',Q_Y)=0$ 对于某个 $\tau'\geq\alpha$。因此，$\sum_{d'\neq d}\mathbf{C}_{\tau'}(d',Q_Y)=\sum_{d'\neq d}\mathbf{C}_{\tau'+1}(d',Q_Y)=\cdots=0$（因为值不能增加）。

断言 *3.* 令 $\alpha':=\max(\tau,\tau')$。根据断言 1，从 $\mathbf{C}_{\alpha'}$ 开始，由于没有颜色为 d 的代理处于 Q_N 中的状态，规则 (6) 和 (8) 不再能将颜色为 d 的代理所在的组改为少数组，也就是说，将 $\mathsf{grp}(q)$ 设为 $false$。此外，由于根据引理 4，颜色为 $d'\neq d$ 的代理都已配对，规则 (2) 也不再能将颜色为 d 的代理所在的组改为少数组。因此，对于任意满足 $\mathbf{C}_{\alpha'}\xrightarrow{*}\mathbf{C}'$ 的 $\mathbf{C}'$，都有 $\mathbf{C}_{\alpha'}(d,Q_m)\geq\mathbf{C}'(d,Q_m)$。

如果 $\mathsf{C}_i(d,Q_m)>0$ 仅对有限多个指标 $i\geq\alpha'$ 成立，那么结论成立。因此，假设 $\mathsf{C}_i(d,Q_m)>0$ 对无限多个 $i\geq\alpha'$ 成立。若存在 $q\in\mathsf{act}(\mathsf{C}_i(d))\cap Q_m\cap P$，那么根据 $\mathsf{maj}(q)$ 的值，规则 (3)、规则（5）后接规则 (7)，或规则（5）后接规则 (8) 将处于可启用状态。每种情况下，都可得到 C_i'，使得 $\mathsf{C}_i\xrightarrow{*}\mathsf{C}_i'$ 且 $\mathsf{C}_i(d,Q_m\cap P)>\mathsf{C}_i'(d,Q_m\cap P)$。因此，根据公平性，我们得到 $\tau''\geq\alpha$，使得对所有 $i\geq\tau''$，$\mathsf{C}_i(d,Q_m\cap P)=0$。

还需证明$\mathbf{C}_{\tau''}(d,\ Q_m\cap U)=0$。规则 (2) 和 (3) 对唯一被修改的状态 q' 有前置条件 $\mathsf{pair}(q)$。因此，它们无法改变未配对代理所属的组。虽然规则 (6)、(7) 和 (8) 没有这样的前置条件，但它们要求状态 p 满足 $\mathsf{maj}(p)\in\{Y,\ N\}$。只有规则 (4) 和 (5) 能改变状态，使其满足该条件。由于这两条规则都只会在 $\mathsf{pair}(q)$ 成立时，才将状态 q 改为满足 $\mathsf{maj}(q')\in\{Y,\ N\}$ 的，未配对代理永远不会处于满足更改其所属组要求的状态。由此可见，状态 $act(C_{\tau''})$ 不可能同时满足 $\neg\mathsf{grp}(q)$ 和 $\neg\mathsf{pair}(q)$。因此，所需的$\mathbf{C}_{\tau''}(d,\ Q_m\cap U)=0$ 得证。

断言 *4*。令 $\alpha':=\max(\tau,\tau')$。根据断言 2，从 $\mathbf{C}_{\alpha'}$开始，没有颜色为 $d'\neq d$的代理处于 Q_Y中的状态。因此，规则 (7) 和 (8) 不再能将颜色为$d'\neq d$ 的代理所属群体改为多数群体。根据引理 4，由于不存在颜色为 $d'\neq d$ 的未配对智能体，规则 (3) 也不再能将他们所属的群体改为多数群体。因此，对于任何满足 $\mathbf{C}_{\alpha'}\xrightarrow{*}\mathbf{C}'$ 的 $\mathbf{C}'$，以及任意 $d'\neq d$，都有 $\mathbf{C}_{\alpha'}(d',Q_M)\geq\mathbf{C}'(d',Q_M)$。

如果 $\sum_{d'\neq d}\mathbf{C}_i(d',Q_M)>0$ 仅对有限多个索引 $i\geq\alpha'$成立，那么结论得证。因此，假设存在无限多个 $i\geq\alpha'$，使得对于 $d'\neq d$，存在 $q\in\mathsf{act}(\mathbf{C}_i(d'))\cap Q_M$。根据 $\mathsf{maj}(q)$ 的值，规则 (2)、先执行规则 (4) 再执行规则 (6)，或先执行规则 (4) 再执行规则 (8) 中的某一项可以启用。在每种情况下，都存在一个配置$\mathbf{C}_i'$，使得 $\mathbf{C}_i\xrightarrow{*}\mathbf{C}_i'$，且 $\mathbf{C}_i(d',Q_M)>\mathbf{C}_i'(d',Q_M)$。因此，根据公平性，结论得证。◀

▶ **引理 10.** 存在 $\tau\geq\alpha$，使得对于每个 $i\geq\tau$ 和 $q\in$ 活动状态集合$(\mathbf{C}_i)$，当且仅当 $|\mathbf{C}_i|_U=0$ 时，$\mathsf{偶数}(q)$ 成立。

证明。 注意，从 $\mathbf{C}_\alpha$ 开始，只有规则 (9) 和 (10) 可能将状态 q 变为状态q'，使得 偶数(q') $\neq$ 偶数(q)。令 $Q_e \coloneqq \{q \in Q\text{: 偶数}(q)\}$。我们进行分类讨论。

情况 $|\mathbf{C}_\alpha|_U > 0$。假设存在无限多个 $i \geq \alpha$，使得对于某个$q \in$活动状态集合($\mathbf{C}_i$)，even(q) 成立，否则我们已经完成证明。由于规则 (9) 已启用，因此存在 $\mathbf{C}'_i$，使得 $\mathbf{C}_i \xrightarrow{*} \mathbf{C}'_i$ 且 $|\mathbf{C}_i|_{Q_e} > |\mathbf{C}'_i|_{Q_e}$。通过反复应用公平性，我们得到 $\tau \geq \alpha$，使得 $\mathbf{C}_\alpha \xrightarrow{*} \mathbf{C}_\tau$ 且 $|\mathbf{C}_\tau|_{Q_e} = 0$。此外，对于任何满足 $\mathbf{C}_\tau \xrightarrow{*} \mathbf{C}'$ 的 $\mathbf{C}'$，规则 (9) 和 (10) 永久处于禁用状态，因为 $|\mathbf{C}'|_{Q_e} = 0$。因此，对于任何 $q \in$活动状态集合($\mathbf{C}'$)，都有 $\neg$even(q)。

情况 $|\mathbf{C}_\alpha|_U = 0$。由于每个代理最初都未配对，因此必有 $\alpha > 0$。对于每个$i \geq \alpha$，注意规则 (9) 在 $\mathbf{C}_i$ 中处于禁用状态。因此，只有规则 (10) 能将状态 q 改变为状态 q'，使得 even(q') $\neq$ even(q)。根据 α 的极小性，我们有

$$\mathbf{C}_{\alpha-1} \xrightarrow{(1)} \mathbf{C}_\alpha, \text{ and hence } |\mathbf{C}_\alpha|_{Q_e} \geq 2.$$

由于规则 (9) 永久禁用，因此对于每个 $i \geq \alpha$，都有 $|\mathbf{C}_i|_{Q_e} > 0$。

如果 act($\mathbf{C}_i$) $\not\subseteq Q_e$ 仅对有限多个指标 $i \geq \alpha$成立，那么结论成立。因此，假设存在无限多个 i，使得对于某个 $q \in$, act($\mathbf{C}_i$)，且 $\neg$even(q) 成立。规则 (10) 已启用，这意味着存在 $\mathbf{C}'_i$，使得$\mathbf{C}_i \xrightarrow{*} \mathbf{C}'_i$，并且 $|\mathbf{C}_i|_{Q_e} < |\mathbf{C}'_i|_{Q_e}$。因此，通过反复应用公平性，我们可以得到某个 $\tau \geq \alpha$，使得 $\mathbf{C}_\alpha \xrightarrow{*} \mathbf{C}_\tau$，并且$|\mathbf{C}_\tau|_{Q_e} = |\mathbf{C}_\tau|$。于是，对于所有 $i \geq \tau$，act($\mathbf{C}_i$) $\subseteq Q_e$ 成立。◀

▶ **引理 11.** 如果 $\varphi_{maj}(\mathbf{C}_0)$ 成立，那么存在 $\tau \geq \alpha$，使得对于每个 $i \geq \tau$ 和$q \in$ 活动状态集合($\mathbf{C}_i$)，都有 多数(q) $\in \{Y, y\}$，且 $\neg$偶数(q) 成立。

证明。 设 $d \in \mathbb{D}$为多数颜色。根据引理 5，我们有 $\mathbf{C}\alpha(d, U) > 0$。由于 d占多数，我们有 $\mathbf{C}_i(d, Q) > \sum_{d' \neq d} \mathbf{C}_i(d', Q)$ 对所有 $i \in \mathbb{N}$均成立。令 τ由引理 9 针对颜色 d给出。对所有 $i \geq \tau$，以下结论成立：

$$\text{act}(\mathbf{C}_i(d)) \subseteq Q_M \text{ and } \text{act}(\mathbf{C}_i(d')) \subseteq Q_m \text{ for all } d' \neq d.$$

我们可以从前述陈述推断，对于所有 $i \geq \tau$，$|\mathbf{C}_i|_{Q_M} > |\mathbf{C}_i|_{Q_m}$。结合引理 6，我们得出，对于所有 $i \geq \tau$，$|\mathbf{C}_i|_{Q_Y} > |\mathbf{C}_i|_{Q_N}$。

我们来证明，Q_N 中活跃状态的数量会永久减少至零。假设存在无穷多个索引 $i \geq \tau$，使得 $|\mathbf{C}_i|_{Q_N} > 0$，否则我们已经完成证明。由于至少存在一个状态 $q \in$ 活动状态集合($\mathbf{C}_i$) $\cap Q_N$，且 $|\mathbf{C}_i|_{Q_Y} > |\mathbf{C}_i|_{Q_N}$，因此规则（11）是可启用的。于是，存在 $\mathbf{C}'_i$，使得 $\mathbf{C}_i \xrightarrow{*} \mathbf{C}'_i$，且 $|\mathbf{C}_i|_{Q_N} > |\mathbf{C}'_i|_{Q_N}$。因此，通过反复应用公平性，我们可以得到某个 $\tau' \geq \tau$，使得 $|\mathbf{C}_{\tau'}|_{Q_N} = 0$。从 $\mathbf{C}_{\tau'}$ 开始，规则（11）和（13）均被禁用。

现在我们证明，来自 Q_n 的活跃状态数量会永久减少至零。假设存在无穷多个 $i \geq \tau'$，使得 $|\mathbf{C}_i|_{Q_n} > 0$；否则结论已成立。由于至少有一个状态 $q \in$ act($\mathbf{C}_i$) $\cap Q_Y$，规则（12）处于启用状态。因此，存在 $\mathbf{C}'_i$，使得 $\mathbf{C}_i \xrightarrow{*} \mathbf{C}'_i$ 且 $|\mathbf{C}_i|_{Q_n} > |\mathbf{C}'_i|_{Q_n}$。通过反复应用公平性，我们得到某个 τ''，使得 $|\mathbf{C}_{\tau''}|_{Q_n} = 0$。从那时起，规则（12）和（14）均处于禁用状态。

现在我们证明，在 q in $\mathbf{C}_\tau$（以及任何可从$\mathbf{C}_\tau$到达的 $\mathbf{C}'$）中，不存在满足 maj(q) $\in \{Y, N\}$ 的活跃状态。回顾引理 9 中的论断：d的代理不再处于 Q_N 中的状态，而 $d' \neq d$ 的代理不再处于 Q_Y中的状态。此外，所有代理都处于正确的组中，即 d 的代理属于多数群体，而 $d' \neq d$ 的代理属于少数群体。假设对于 $i \geq \tau$，$d' \neq d$（分别为 d）中的某个代理处于状态 $q \in$ act($\mathbf{C}_i$)，且满足 maj(q) $= N$（分别满足 maj(q) $= Y$）。这意味着存在一个可从 $\mathbf{C}_i$ 到达的配置 $\mathbf{C}'$，使得某个状态 $q' \in$act($\mathbf{C}'$(d'))

（分别为 $q' \in \mathsf{act}(\mathbf{C}'(d))$）满足 $\mathsf{grp}(q')$（分别满足 $\neg\mathsf{grp}(q')$）。根据公平性，最终会到达这样的配置，这与引理 9 中 τ 的定义矛盾。因此，对于任意状态$q \in \mathsf{act}(\mathbf{C}_{\tau''})$，都有 $\mathsf{maj}(q) \in \{Y, y\}$。

最后，根据引理 10，存在 $\tau''' \geq \tau''$，使得对于每个 $i \geq \tau'''$ 和 $q \in \in \mathsf{act}(\mathbf{C}_i)$，$\neg \mathsf{even}(q)$ 都成立。◀

▶ 引理 *12.* 若 $\neg\varphi_{maj}(C_0)$ 成立，则存在 $\tau \geq \alpha$，使得以下条件之一成立：对于每个 $i \geq \tau$ 和 $q \in \in act(C_i)$，$even(q)$ 成立；或者对于每个 $i \geq \tau$ 和 $q \in \in act(C_i)$，$maj(q) \in \{N, n\}$ 成立。

证明。若 $|\mathbf{C}_\alpha|_U = 0$ 成立，则根据引理 4 和引理 10，存在 $u \geq \alpha$，使得对于任意 $i \geq u$，每个状态 $q \in \mathrm{act}(\mathbf{C}_i)$ 都满足 $\mathrm{even}(q)$。

现在，假设情况并非如此。根据引理 4，存在唯一颜色 $d \in \mathbb{D}$，使得$|\mathbf{C}_\alpha(d)|_U > 0$。由于不存在多数颜色，对于每个$i \in \mathbb{N}$，都有 $\mathbf{C}_i(d, Q) \leq \sum_{d' \neq d} \mathbf{C}_i(d', Q)$。令 $\tau \geq \alpha$ 为引理 9 给出的最小值。根据引理 9，对于每个 $i \geq \tau$:

$$\mathrm{act}(\mathbf{C}_i(d)) \subseteq Q_M \text{ and } \mathrm{act}(\mathbf{C}_i(d')) \subseteq Q_m \text{ for all } d' \neq d.$$

因此，我们有 $|\mathbf{C}_i|_{Q_M} \leq |\mathbf{C}_i|_{Q_m}$ 对所有 $i \geq \tau$均成立。根据引理 6，这意味着 $|\mathbf{C}_i|_{Q_Y} \leq |\mathbf{C}_i|_{Q_N}$ 对所有 $i \geq \tau$均成立。

我们来证明，Q_Y 中的活跃状态数永久减少至零。假设存在无穷多个指标 $i \geq \tau$ 使得 $|\mathbf{C}_i|_{Q_Y} > 0$。由于至少存在一个状态 $q \in$ 活动状态集合$(\mathbf{C}_i) \cap Q_Y$且 $|\mathbf{C}_i|_{Q_Y} \leq |\mathbf{C}_i|_{Q_N}$，规则（11）处于启用状态。因此，存在$\mathbf{C}'_i$ 使得 $\mathbf{C}_i \xrightarrow{*} \mathbf{C}'_i$ 且 $|\mathbf{C}_i|_{Q_Y} > |\mathbf{C}'_i|_{Q_Y}$。通过反复应用公平性，我们得到某个 $\tau' \geq \tau$，满足 $|\mathbf{C}_{\tau'}|_{Q_Y} = 0$。令 $\tau\prime$为最小的。此时，规则（11）和规则（12）均已被禁用。

由于规则（12）已被禁用，我们有 $|\mathbf{C}_i|_{Q_y} \geq |\mathbf{C}_{i+1}|_{Q_y}$ 对所有 $i \geq \tau'$均成立。我们来证明，Q_y 中的活跃状态数永久减少至零。分为两种情况。

首先，考虑 $|\mathbf{C}_{\tau'}|_{Q_N} > 0$ 的情形。假设存在无穷多个$i \geq \tau'$，使得 $|\mathbf{C}_i|_{Q_y} > 0$；否则问题便已解决。由于至少存在一个状态$q \in \mathsf{act}(\mathbf{C}_i) \cap Q_N$，因此规则（13）已启用。因此，存在 $\mathbf{C}'_i$，使得 $\mathbf{C}_i \xrightarrow{*} \mathbf{C}'_i$，且 $|\mathbf{C}_i|_{Q_y} > |\mathbf{C}'_i|_{Q_y}$。反复应用公平性，可得到某个 $\tau'' \geq \tau'$，使得$|\mathbf{C}_{\tau''}|_{Q_y} = 0$。这会禁用规则（13）和（14）。

现在考虑剩余的情形，即 $|\mathbf{C}_{\tau'}|_{Q_N} = 0$。由引理 6 可知，对于任意 $v \geq \tau$，都有 $|\mathbf{C}_v|_{Q_M} = |\mathbf{C}_v|_{Q_m}$。由于每个代理最初都处于满足 $\mathsf{grp}(q)$ 的状态 q，又由于 $|\mathbf{C}_\alpha|_U > 0$，且 $\varphi_{\mathrm{maj}}(\mathbf{C}_0) =$ 为假，因此执行过程中至少有一个代理的组发生了变化。存在一个转移 $t = ((p, q) \rightsquigarrow (p', q'))$ 以及数据 $d_1, d_2 \in \mathbb{D}$，使得 $(\mathsf{grp}(p) \neq \mathsf{grp}(p')) \vee (\mathsf{grp}(q) \neq \mathsf{grp}(q'))$ 成立，$\mathbf{C}_{\tau-1} \xrightarrow{t} \mathbf{C}_\tau$ 且 $\mathbf{C}_\tau = \mathbf{C}_{\tau-1} - (\boldsymbol{p}_{d_1} + \boldsymbol{q}_{d_2}) + (\boldsymbol{p}'_{d_1} + \boldsymbol{q}'_{d_2})$。注意，$t$ 必定是由规则 (2)、(3)、(6)、(7) 或 (8) 导致的转移。我们区分两种子情况。

子情况：转移 t 源于规则 *(2)*、*(3)*、*(6)* 或 *(7)*。回顾一下，对于任意 $v \geq \tau$，群在 $\mathbf{C}_v$ 中均保持不变，因此 $|\mathbf{C}_v|_{Q_Y} = |\mathbf{C}_v|_{Q_N}$，因为 $|\mathbf{C}_v|_{Q_M} = |\mathbf{C}_v|_{Q_m}$。根据规则的定义，我们知道，要么 $|\mathbf{C}_\tau|_{Q_N} > 0$（这意味着 $|\mathbf{C}_\tau|_{Q_Y} > 0$），要么 $|\mathbf{C}_\tau|_{Q_Y} > 0$（这意味着 $|\mathbf{C}_\tau|_{Q_N} > 0$）。因此，

$$\mathbf{C}_{\tau'-1} \xrightarrow{(11)} \mathbf{C}_{\tau'} \text{ and } |\mathbf{C}_{\tau'-1}|_{Q_Y} = |\mathbf{C}_{\tau'-1}|_{Q_N} = 1.$$

根据规则（12）的定义，可得 $|\mathbf{C}_{\tau'}|_{Q_n} \geq 2$。

子情形：转移 t 源于规则（*8*）。根据（8）的定义，我们有 $|\mathbf{C}_\tau|_{Q_n} \geq 2$。如果$|\mathbf{C}_\tau|_{Q_Y} = |\mathbf{C}_\tau|_{Q_N} = 0$，那么 $\tau = \tau'$。否则，我们有

$$|\mathbf{C}_\tau|_{Q_Y} = |\mathbf{C}_\tau|_{Q_N} > 0 \text{ and } \mathbf{C}_{\tau'-1} \xrightarrow{(11)} \mathbf{C}_{\tau'}.$$

同样可得，$|\mathbf{C}_{\tau'}|_{Q_n} \geq 2$。

在这两种子情况下，我们都有 $|\mathbf{C}_{\tau'}|_{Q_n} > 0$。注意，规则（13）已被禁用。因此，现在只有规则（14）能够将一个状态 q 改变为满足 $\mathsf{maj}(q') \neq \mathsf{maj}(q)$ 的状态 q'。由于 $|\mathbf{C}_{\tau'}|_{Q_n} > 0$，对于任何满足 $|\mathbf{C}_j|_{Q_y} > 0$ 的 $j \geq \tau'$，规则（14）均已启用。因此，通过反复应用公平性，我们可以得到某个 $\tau'' \geq \tau'$，使得 $|\mathbf{C}_{\tau''}|_{Q_y} = 0$。

利用引理 11 证明中给出的结果，我们知道在 $\mathbf{C}_\tau$ 中（以及在任何可从 $\mathbf{C}_\tau$ 到达的 $\mathbf{C}'$ 中），不存在满足 $\mathsf{maj}(q) \in \{Y, N\}$ 的活跃状态q。因此，证明完毕，因为对于每个 $i \geq \tau''$ 和 $q \in \mathsf{act}(\mathbf{C}_i)$，$\mathsf{maj}(q) \in \{N, n\}$ 都成立。◀

B 第 4.1 节缺失的证明

▶ **引理 A1** (单调性)。设 $\mathbf{C}$、$\mathbf{C}$ 和$\mathbf{C}'$ 是配置，且满足$\mathbf{C} \xrightarrow{*} \mathbf{C}$ 和$\mathbf{C} \sqsubseteq \mathbf{C}'$。存在$\mathbf{C}'$，使得$\mathbf{C}' \xrightarrow{*} \mathbf{C}'$ 且$\mathbf{C} \sqsubseteq \mathbf{C}'$。

证明。考虑 $\mathbf{C} \xrightarrow{t} \mathbf{C}$ 的情形。一般情形“$\xrightarrow{*}$”可由归纳法得出。令ρ: $\mathbb{D} \to \mathbb{D}$ 为一个单射，使得对所有 $x \in \mathbb{D}$，$\mathbf{C}(x) \leq \mathbf{C}'(\rho(x))$。令 $t = ((p, q) \rightsquigarrow (p', q'))$ 为 $\mathbf{C}$ 中使用的转移，其中 $d, e \in \mathbb{D}$。由于 $\mathbf{C} \geq \boldsymbol{p}_d + \boldsymbol{q}_e$，我们有 $\mathbf{C}' \geq \boldsymbol{p}_{\rho(d)} + \boldsymbol{q}_{\rho(e)}$。因此，可以在 $\mathbf{C}'$ 中使用 t，得到 $\mathbf{C}' := \mathbf{C}' - (\boldsymbol{p}_{\rho(d)} + \boldsymbol{q}_{\rho(e)}) + (\boldsymbol{p}'_{\rho(d)} + \boldsymbol{q}'_{\rho(e)})$。对所有 $x \in \mathbb{D}$，我们有 $\mathbf{C}(d) \leq \mathbf{C}'(\rho(x))$，因此 $\mathbf{C}' \sqsubseteq \mathbf{C}'$。◀

▶**引理 A2.** 集合 U 向上封闭，且具有有限基。

证明。只需证明 $\mathcal{U}$ 是向上封闭的。事实上，由于 $\sqsubseteq$ 是良拟序（例如，参见 [18]），可知 $\mathcal{U}$ 具有有限基。

设 $\mathbf{C}$ 和 $\mathbf{C}'$ 是配置，满足 $\mathbf{C} \in \mathsf{U}$ 且 $\mathbf{C} \sqsubseteq \mathbf{C}'$。若 $O(\mathbf{C}') = \bot$，则结论成立。因此，假设 $O(\mathbf{C}') = b \in \{0, 1\}$。由于 $\mathsf{act}(\mathbf{C}) \subseteq \mathsf{act}(\mathbf{C}')$，我们有$O(\mathbf{C}) = b$。由于 $\mathbf{C} \in \mathsf{U}$，存在一个配置 $\mathbf{C}$，使得 $\mathbf{C} \xrightarrow{*} \mathbf{C}$ 且 $O(\mathbf{C}) \neq b$。根据引理 A1，存在 $\mathbf{C}'$，满足 $\mathbf{C}' \xrightarrow{*} \mathbf{C}'$ 且 $\mathbf{C} \sqsubseteq \mathbf{C}'$。由于 $\mathsf{act}(\mathbf{C}) \subseteq \mathsf{act}(\mathbf{C}')$，因此我们要么有 $O(\mathbf{C}') = O(\mathbf{C})$，要么有 $O(\mathbf{C}') = \bot$。所以，我们得到 $\mathbf{C}' \in \mathsf{U}$。◀

▶ **引理 A3. </style}设，且设 $\mathbf{C}$,$\mathbf{C}'$ 为配置，满足 $\mathbf{C} \sqsubseteq \mathbf{C}'$。则 $\tau_k(\mathbf{C}) \sqsubseteq \mathbf{C}$，且 $\tau_k(\mathbf{C}) \sqsubseteq \tau_k(\mathbf{C}')$</style}**。

证明。对于每个 $d \in \mathbb{D}$ 和 $q \in Q$，都有 $\tau_k(\mathbf{C})(d, q) = \min(\mathbf{C}(d, q)\ k) \leq \mathbf{C}(d, q)$。因此，$\tau_k(\mathbf{C}) \sqsubseteq \mathbf{C}$ 显然成立（使用恒等函数）。

设 ρ: $\mathbb{D} \to \mathbb{D}$ 是一个单射，使得对于每个 $d \in \mathbb{D}$，$\mathbf{C}(d) \leq \mathbf{C}'(\rho(d))$。对于每个$d \in \mathbb{D}$ 和 $q \in Q$，都有

$$\tau_k(\mathbf{C})(d, q) = \min(\mathbf{C}(d, q), k) \leq \min(\mathbf{C}'(\rho(d), q), k) = \tau_k(\mathbf{C}')(\rho(d), q).$$

因此，$\tau_k(\mathbf{C}) \sqsubseteq \tau_k(\mathbf{C}')$（使用单射 ρ）。◀

▶**引理 14.** 设 ψ 是由无序数据的群体协议计算的谓词。设 S_b 为该协议输出为 b 的稳定配置集合。存在 $k \geq 1$，使得对于所有 $b \in \{0,1\}$, 均有 $\mathbf{C} \in S_b$ 当且仅当 $\tau_k(\mathbf{C}) \in S_b$。

证明。根据引理 A2，集合 $\mathcal{U}$有一个有限基 $\{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_n\}$。令 $k := \max\{\mathbf{C}_i(d, q) : d \in \mathbb{D}, q \in Q\}$。注意，对所有 $i \in [1..n]$，$\tau_k(\mathbf{C}_i) = \mathbf{C}_i$。我们先证明 $\mathbf{C} \in \mathcal{U}$ 当且仅当$\tau_k(\mathbf{C}) \in \mathcal{U}$。

$\Rightarrow$) 由于$\mathbf{C} \in \mathcal{U}$，我们有$\mathbf{C}_i \sqsubseteq \mathbf{C}$，其中某个 $i \in [1..n]$满足条件。由引理 A3，我们有 $\tau_k(\mathbf{C}_i) \sqsubseteq \tau_k(\mathbf{C})$。由于$\mathbf{C}_i = \tau_k(\mathbf{C}_i)$，我们得到$\mathbf{C}_i \sqsubseteq \tau_k(\mathbf{C})$，因此 $\tau_k(\mathbf{C}) \in \mathcal{U}$。

⇐) 由于 $\tau_k(\mathbf{C}) \in \mathcal{U}$，我们有$\mathbf{C}_i \sqsubseteq \tau_k(\mathbf{C})$，其中某个 $i \in [1..n]$满足条件。由引理 A3，我们有$\tau_k(\mathbf{C}) \sqsubseteq \mathbf{C}$。因此，$\mathbf{C}_i \sqsubseteq \mathbf{C}$，这意味着$\mathbf{C} \in \mathcal{U}$。

还需证明主要结论。由于 $\mathcal{S}_0 \cup \mathcal{S}_1$ 是 $\mathcal{U}$ 的补集，因此 $\mathbf{C} \in \mathcal{S}_0 \cup \mathcal{S}_1$ 当且仅当 $\tau_k(\mathbf{C}) \in \mathcal{S}_0 \cup \mathcal{S}_1$。此外，$O(\mathbf{C}) = O(\tau_k(\mathbf{C}))$，因为 $\mathsf{act}(\mathbf{C}) = \mathsf{act}(\tau_k(\mathbf{C}))$。因此，$\mathbf{C} \in \mathcal{S}_b$ 当且仅当 $\tau_k(\mathbf{C}) \in \mathcal{S}_b$。◀

▶**引理 15**。设 ψ 是一个由无序数据即时观察协议计算的谓词。设 k 是引理 *14* 给出的阈值，且$\mathbf{C}_0$ 是一个初始配置，使得对于某个 $d \in \mathbb{D}$ 和某个 $q \in I$，$\mathbf{C}_0(d, q) \geq \ell = |Q| \cdot (k-1) + 1$。那么，$\psi(\mathbf{C}_0 + \boldsymbol{q}_d) = \psi(\mathbf{C}_0)$。

证明。设 $\mathsf{C}_0\mathsf{C}_1 \ldots \mathsf{C}_m$ 是一条公平执行，它到达稳定配置 C_m，其输出为$b \in \{0, 1\}$。设 $s_i(j)$ 是配置 C_i 中数据项 d 的第 j 个代理的状态。设 a 满足 $s_0(a) = q$，且 $\mathsf{C}_m(d, s_m(a)) \geq k$；也就是说，一个初始状态为 q 的代理最终到达状态 $s_m(a) = q'$，并且 C_m 中至少有 k 个代理。注意，由于 $\mathsf{C}_0(d, q) \geq \ell = |Q| \cdot (k-1) + 1$，根据鸽巢原理，这样的状态 q' 必定存在。我们首先用归纳法证明 $\mathsf{C}_0 + \boldsymbol{q}_d \xrightarrow{*} \mathsf{C}_i + \boldsymbol{s_i(a)}_d$。

情形 $i = 0$。由于 $s_0(a) = q$ 且 a 是数据项 d，因此可得 $\mathbf{C}_0 + \boldsymbol{q}_d = \mathbf{C}_0 + \boldsymbol{s_0(a)}_d$，这意味着 $\mathbf{C}_0 + \boldsymbol{q}_d \xrightarrow{*} \mathbf{C}_0 + \boldsymbol{s_0(a)}_d$。

情形 $i > 0$。假设 $\mathbf{C}_0 + \boldsymbol{q}_d \xrightarrow{*} \mathbf{C}_{i-1} + \boldsymbol{s_{i-1}(a)}_d$。若 $s_{i-1}(a) = s_i(a)$，则

$$\begin{aligned}
\mathbf{C}_0 + \boldsymbol{q}_d &\xrightarrow{*} \mathbf{C}_{i-1} + \boldsymbol{s_{i-1}(a)}_d && \text{（根据归纳假设）} \\
&\to \mathbf{C}_i + \boldsymbol{s_{i-1}(a)}_d && (\mathbf{C}_{i-1} \to \mathbf{C}_i) \\
&= \mathbf{C}_i + \boldsymbol{s_i(a)}_d && (s_{i-1}(a) = s_i(a))
\end{aligned}$$

若 $s_{i-1}(a) \neq s_i(a)$，则存在某个 $p \in Q$ 以及某个转移 $t = ((p, s_{i-1}(a)) \sim, (p, s_i(a)))$，使得 $\mathbf{C}_{i-1} \xrightarrow{t} \mathbf{C}_i$。由于该协议是即时观察协议，此转移会使被观察的代理在 $\mathbf{C}_i$中保持不变，即 $\mathbf{C}_i \geq \boldsymbol{p}_{d'}$，其中某个 $d' \in \mathbb{D}$。该代理随后又可被处于状态 $s_{i-1}(a)$、数据项为 d 的代理观察。因此，

$$\begin{aligned}
\mathbf{C}_0 + \boldsymbol{q_d} &\xrightarrow{*} \mathbf{C}_{i-1} + \boldsymbol{s_{i-1}(a)}_d && \text{（根据归纳假设）} \\
&\xrightarrow{t} \mathbf{C}_i + \boldsymbol{s_{i-1}(a)}_d && (\mathbf{C}_{i-1} \to \mathbf{C}_i) \\
&\xrightarrow{t} \mathbf{C}_i + \boldsymbol{s_i(a)}_d && \text{（通过观察处于状态 } p \text{ 的 } d' \text{ 代理）}
\end{aligned}$$

现在，假设 $\mathbf{C}_m + \boldsymbol{q}'_d \in \mathcal{U}$。我们有

$$\begin{aligned}
\mathbf{C}_m + \boldsymbol{q}'_d \in \mathcal{U} &\implies \tau_k(\mathbf{C}_m + \boldsymbol{q}'_d) \in \mathcal{U} && \text{(by Lemma 14)} \\
&\implies \tau_k(\mathbf{C}_m) \in \mathcal{U} && (\mathbf{C}_m(d, q') \geq k) \\
&\implies \mathbf{C}_m \in \mathcal{U} && \text{(by Lemma 14)}
\end{aligned}$$

这与我们的假设 $\mathbf{C}_m \in \mathcal{S}$ 相矛盾。因此，$\mathbf{C}_m + \boldsymbol{q}'_d$ 必须是稳定的。注意，由于 $\mathbf{C}_m(q') > 0$，因此可得 $O(\mathbf{C}_m + \boldsymbol{q}'_d) = b = O(\mathbf{C}_m)$。这意味着从 $\mathbf{C}_0 + \boldsymbol{q}_d$ 开始的执行会到达一个输出为 b 的稳定配置。因此，$\psi(\mathbf{C}_0) = \psi(\mathbf{C}_0 + \boldsymbol{q}_d)$。◀

▶ **引理 16.** 设$\mathbf{C}$ 和$\mathbf{C}'$ 是配置，且满足$\mathbf{C} \xrightarrow{*} \mathbf{C}'$。对于每个 $d \in \mathrm{supp}(\mathbf{C})$和 $d' \in \mathbb{D} \setminus \mathrm{supp}(\mathbf{C})$，都有$\mathbf{C} + (\mathbf{C}(d))_{d'} \xrightarrow{*} \mathbf{C}' + (\mathbf{C}'(d))_{d'}$。

证明. 设 d, d' 如上所述，并设 $\mathbf{C} = \mathbf{C}_0 \xrightarrow{t_1} \mathbf{C}_1 \xrightarrow{t_2} \cdots \xrightarrow{t_i} \mathbf{C}_i = \mathbf{C}'$。我们通过对 i进行归纳法证明该结论。由于该结论在 $i = 0$的情形下显然成立，下面考虑$n > i$的情形。我们进行分类讨论。

分类讨论 $\mathbf{C}_i(d) = \mathbf{C}_{i-1}(d)$。我们有

$$\begin{aligned} \mathbf{C}_0 + (\mathbf{C}_0(d))_{d'} &\xrightarrow{*} \mathbf{C}_{i-1} + (\mathbf{C}_{i-1}(d))_{d'} && \text{(by induction hypothesis)} \\ &= \mathbf{C}_{i-1} + (\mathbf{C}_i(d))_{d'} \\ &\to \mathbf{C}_i + (\mathbf{C}_i(d))_{d'} && (\text{by } \mathbf{C}_{i-1} \to \mathbf{C}_i). \end{aligned}$$

分类讨论 $\mathsf{C}_i(d) \neq \mathsf{C}_{i-1}(d)$。我们证明 $(\mathsf{C}_{i-1}(d))_{d'} \xrightarrow{t} (\mathsf{C}_i(d))_{d'}$。根据假设，存在一个转移 $t = ((p, q) \sim (p, q')) \in \delta$，使得 $\mathsf{C}_{i-1} \xrightarrow{t} \mathsf{C}_i$, $q \in \mathsf{act}(\mathsf{C}_{i-1}(d))$ 且 $q \neq q'$。后者意味着 $(\mathsf{C}_{i-1}(d))_{d'} \geq \mathsf{q}_{\mathsf{d}'}$。下面对 $\sim$进行分类讨论。

- 假设 $\sim$ 是“=”。若 $p = q$，则有 $\mathsf{C}_{i-1}(d, q) \geq 2$。因此，我们有 $(\mathsf{C}_{i-1}(d))_{d'} \xrightarrow{t} (\mathsf{C}_i(d))_{d'}$。否则，有 $\mathsf{C}_{i-1}(d, p) \geq 1$ 且 $\mathsf{C}_{i-1}(d, q) \geq 1$。这也意味着 $(\mathsf{C}_{i-1}(d))_{d'} \xrightarrow{t} (\mathsf{C}_i(d))_{d'}$ 成立。假设 $\sim$ 是“$\neq$”。某个颜色 $d'' \in \mathsf{supp}(\mathsf{C}_0)$ 满足 $p \in \mathsf{act}(\mathsf{C}_{i-1}(d''))$，因此$p \in \mathsf{act}(\mathsf{C}_i(d''))$。
- 注意， $d'' \in \mathsf{supp}(\mathsf{C}_0)$ 意味着 $d'' \neq d'$。因此，我们有 $(\mathsf{C}_{i-1}(d))_{d'} \xrightarrow{t} (\mathsf{C}_i(d))_{d'}$。 ◀

▶**引理 17**。 设 ψ 是由无序数据群体协议计算的谓词。设 $\boldsymbol{f}$ 是满足 $\mathrm{act}(\boldsymbol{f}) \subseteq I$ 的形式。存在 $h(\boldsymbol{f}) \in \mathbb{N}$，使得对于所有初始配置$\mathbf{C}_0$以及满足 $\#_{\boldsymbol{f}}(\mathbf{C}_0) \geq h(\boldsymbol{f})$ 的 $d \in \mathbb{D} \setminus \mathrm{supp}(\mathbf{C}_0)$，都有 $\psi(\mathbf{C}_0 + \boldsymbol{f}_d) = \psi(\mathbf{C}_0)$。

证明。 设 $\boldsymbol{f}$ 是满足 $\mathsf{act}(\boldsymbol{f}) \subseteq I$ 的形式，设 $\{\mathbf{C}_1, \ldots, \mathbf{C}_n\}$ 是由引理 A2 给出的 U 的一组基，并令 $m \coloneqq \max(|\mathsf{supp}(\mathbf{C}_1)|, |\mathsf{supp}(\mathbf{C}_2)|, \ldots, |\mathsf{supp}(\mathbf{C}_n)|)$。我们定义 $h(\boldsymbol{f})$ 如下：

$$h(\boldsymbol{f}) \coloneqq (m-1) \cdot |Q|^{\sum_{q \in Q} \boldsymbol{f}(q)} + 1.$$

设 C_0 是满足 $\#_{\boldsymbol{f}}(\mathsf{C}_0) \geq h(\boldsymbol{f})$ 的初始配置。令 $b \coloneqq \psi(\mathsf{C}_0)$。设 $\mathsf{C} \in \mathsf{S}_b$ 满足 $\mathsf{C}_0 \xrightarrow{*} \mathsf{C}$。根据鸽巢原理，存在 $d \in \mathsf{supp}(\mathsf{C}_0)$，使得 $\mathsf{C}_0(d) = \boldsymbol{f}$，并且

- $\mathbf{C}(d) = \boldsymbol{f}'$ 与 $\#_{\boldsymbol{f}'}(\mathbf{C}) \geq m$。

现在，我们用新的数据项 $d' \in \mathbb{D} \setminus \mathrm{supp}(\mathbf{C}_0)$ 将形式 $\boldsymbol{f}$ 注入。根据引理 16，我们有

$$\mathbf{C}_0 + \boldsymbol{f}_{d'} = \mathbf{C}_0 + (\mathbf{C}_0(d))_{d'} \xrightarrow{*} \mathbf{C} + (\mathbf{C}(d))_{d'} = \mathbf{C} + \boldsymbol{f}'_{d'}.$$

设 $\mathsf{C}' \coloneqq \mathsf{C} + \boldsymbol{f}'_{d'}$。由于 $\mathsf{C}(d) = \boldsymbol{f}'$ 且 $\mathsf{C} \in \mathsf{S}_b$，我们有 $O(\mathsf{C}) = O(\boldsymbol{f}'_{d'}) = b$，因此 $O(\mathsf{C}') = b$。现在证明 $\mathsf{C}' \in \mathsf{S}_b$。为导出矛盾，假设 $\mathsf{C}' \in \mathsf{U}$。存在 $i \in [1..n]$，使得 $\mathsf{C}_i \sqsubseteq \mathsf{C}'$。由此可知，存在一个单射 $\rho\colon \mathbb{D} \to \mathbb{D}$，使得对每个 $x \in \mathbb{D}$，$\mathsf{C}_i(x) \leq \mathsf{C}'(\rho(x))$。由于

$$\#_{\boldsymbol{f}'}(\mathbf{C}') > \#_{\boldsymbol{f}'}(\mathbf{C}) \geq m \geq |\mathrm{supp}(\mathbf{C}_i)|,$$

$d^* \in \mathbb{D}$ 使得 $\mathbf{C}'(d^*) = \boldsymbol{f}'$，并且以下两者之一成立： d^* 不属于 ρ 的像，或者$\mathbf{C}_i(\rho^{-1}(d^*)) = \mathbf{0}$。我们来证明 $\mathbf{C}_i \sqsubseteq \mathbf{C}$，由此得到矛盾 $\mathbf{C} \in \mathcal{U}$。我们进行分类讨论。

情形 $\rho^{-1}(d')$ 未定义。显然，对于所有 $x \in \mathbb{D}$，都有 $\mathbf{C}_i(x) \leq \mathbf{C}'(\rho(x)) = \mathbf{C}(\rho(x))$。

情况 $d^* = d'$。由于 $\mathbf{C}_i(\rho^{-1}(d')) = \mathbf{C}_i(\rho^{-1}(d^*)) = \mathbf{0} = \mathbf{C}(d') = \mathbf{C}(d^*)$，且 $\mathbf{C}$ 与 $\mathbf{C}'$ 仅在 d' 上不同，因此对于每个 $x \in \mathbb{D}$，都有 $\mathbf{C}_i(x) \leq \mathbf{C}(\rho(x))$。因此， $\mathbf{C}_i \sqsubseteq \mathbf{C}$。

情形 $d^* \neq d'$。令 ρ': $\mathbb{D} \to \mathbb{D}$ 为由 ρ给出的单射，其中 d^* 扮演 d'的角色（并且可能需要反过来以确保 ρ' 确实是一个单射），即

$$\rho'(x) \coloneqq \begin{cases} d' & \text{if } \rho^{-1}(d^*) \text{ is defined and } x = \rho^{-1}(d^*), \\ d^* & \text{if } x = \rho^{-1}(d'), \\ \rho(x) & \text{otherwise.} \end{cases}$$

作为 $d^* \neq d'$，我们有 $\mathbf{C}(d^*) = \mathbf{C}'(d^*) = \boldsymbol{f}'$。因此，

$$\mathbf{C}_i(\rho^{-1}(d')) \leq \mathbf{C}'(d') = \boldsymbol{f}' = \mathbf{C}(d^*) = \mathbf{C}(\rho'(\rho^{-1}(d'))).$$

此外，如果 $\rho^{-1}(d^*)$ 有定义，那么我们有 $\mathbf{C}_i(\rho^{-1}(d^*)) = \mathbf{0} = \mathbf{C}(d') = \mathbf{C}(\rho'(\rho^{-1}(d^*)))$。因此，由于 $\mathbf{C}$ 与 $\mathbf{C}'$ 仅在数据项 d' 上不同，对于所有 $x \in \mathbb{D}$，我们都有 $\mathbf{C}_i(x) \leq \mathbf{C}(\rho'(x))$，这意味着 $\mathbf{C}_i \sqsubseteq \mathbf{C}$。◀

C 第4.3节缺失的证明

我们先在一张表中回顾所有转移:

领导者和控制器选举			
rule	状态前置条件	颜色前置条件	状态更新
(1)	领导者(p) 领导者(q)	$d_1 = d_2$	角色(q') $= -\lvert$角色(q)$\rvert$ ¬领导者(q')
(2)	控制器(p) $= 1$ 控制器(q) $= 1$	*none*	控制器(q') $= -1$
按数据项统计元素数量			
rule	状态前置条件	颜色前置条件	状态更新
(3)	元素(p) $=$ 元素(q) 值(q) $=$ 值(p) $< r$	$d_1 = d_2$	值(q') $=$ 值(q) $+ 1$
(4)*	计数$_{\text{elem}(p)}$(q) $<$值(p) 领导者(q)	$d_1 = d_2$	计数$_{\text{elem}(p)}$(q') $=$ 值(p) `if` (角色(q) $> 0 \wedge$ 值(p) $\notin T$(角色(q), 元素(p))): 角色(q') $=$ −角色(q)
角色分布与任务跟踪			
rule	状态前置条件	颜色前置条件	状态更新
(5)	领导者(q) 角色(q) $= 0$ $\bigwedge_{j\in[1..m]}$ 计数$_j$(q) $\in T(i,j)$	*none*	角色(q') $= i$
(6)*	控制器(p) 领导者(q) 角色(q) $= i > 0$ $\bigvee_{i'\in[1..n]\setminus\{i\}}$ (¬任务$_{i'}$(p)$\wedge$ $\bigwedge_{j\in[1..m]}$ 计数$_j$(q) $\in T(i',j)$)	*none*	角色(q') $= -i$
(7)*	角色(p) $\neq 0$ 控制器(q) $= 1$	*none*	task$^{\text{role}(p)\rvert}$(q') $=$ (角色(p) > 0)
(8)*	控制器(p) $= 1$ 角色(q) < 0 ¬任务$^{\lvert\text{role}(q)\rvert}$($p$)	*none*	角色(q') $= 0$
(9)	控制器(p) $= -1$ 控制器(q) $= 1$	*none*	$\bigwedge_{i\in[1..m]}$ ¬任务$_i$(q')
(10)	控制器(p) $= 1$ 控制器(q) $= -1$ $\bigwedge_{i\in[1..m]}$ ¬任务$_i$(p)	*none*	控制器(q') $= 0$
输出传播			
rule	状态前置条件	颜色前置条件	状态更新
(11)*	控制器(p)	*none*	输出(q') $= \bigwedge_{i=1}^{n}$ 任务$_i$(p)

请注意，为简洁起见，在本节余下部分中，我们使用记号$\mathbf{C} \xrightarrow{(n)} \mathbf{C}'$表示$\mathbf{C} \xrightarrow{t} \mathbf{C}'$通过使用由规则（$n$）产生的转移 t。

▶ **引理 21.** 存在 $\tau \in \mathbb{N}$，使得 $|\mathbf{C}_\tau|_{Q_C} = |\mathbf{C}_{\tau+1}|_{Q_C} = \cdots = 1$，并且 $|\mathbf{C}_\tau(d)|_{Q_L} = |\mathbf{C}_{\tau+1}(d)|_{Q_L} = \cdots = 1$ 对每个 $d \in \mathbb{D}$ 都成立。

证明。 我们分别证明该断言的两个部分。取两个阈值 τ中的较大值，即可得到该断言成立。

关于 Q_L 的阈值。注意，$\mathbf{C} \xrightarrow{(1)} \mathbf{C}'$ 意味着 $|\mathbf{C}'|_{Q_L} = |\mathbf{C}|_{Q_L} - 1 > 0$，而 $|\mathbf{C}|_{Q_L} = |\mathbf{C}'|_{Q_L}$则意味着相反的情况。因此，存在 $\tau \in \mathbb{N}$ 和 $k \geq 1$，使得 $|\mathbf{C}_\tau|_{Q_L} = |\mathbf{C}_{\tau+1}|_{Q_L} = \cdots = k$。如果 $k = 1$，则结论成立。因此，为了导出矛盾，假设 $k \geq 2$。规则 (1) 在每个 $\mathbf{C}_\tau$、$\mathbf{C}_{\tau+1}$、..... 中均处于可执行状态。由公平性可知，对某个 $\tau' \geq \tau$，有$|\mathbf{C}_{\tau'}|_{Q_L} > |\mathbf{C}_{\tau'+1}|_{Q_L}$，这与假设矛盾。

的阈值。证明相同，但使用规则 (2)。 ◀

▶ **引理 22。** 存在 $\tau \in \mathbb{N}$，使得对于每个 $\tau' \geq \tau$、$d \in \mathrm{supp}(\mathbf{C}_{\tau'})$ 和 $j \in [1..m]$，如果 $\mathbf{C}_0(d, Q_j) > 0$，则对于某个(d, j)有效状态 q，$\mathbf{C}_{\tau'}(d, q) > 0$ 成立。

证明。注意，规则 (3) 是唯一会修改状态值的规则；它要求两个具有相同值的状态，并将单个状态的值增加 1。因此，对于任意 C、C' 和 $d \in \mathbb{D}$，使得 $\mathsf{C} \to \mathsf{C}'$，且 $\mathsf{C}' = \mathsf{C} - \boldsymbol{q}_d + \boldsymbol{q}'_d$，其中 q、$q' \in Q$ 为某些状态，$\mathsf{val}(q') = \mathsf{val}(q)$ 或 $\mathsf{val}(q') = \mathsf{val}(q) + 1$ 必然成立。进一步注意，没有任何规则会将状态 q 更改为状态 q'，使得 $\mathsf{elem}(q') \neq \mathsf{elem}(q)$，即对于任意 q，$\mathsf{elem}(q)$ 都是不变的。

设 $d \in \mathbb{D}$ 和 $j \in [1..m]$ 满足 $\mathbf{C}_0(d, Q_j) > 0$。我们定义

$$Q_{j,k} \coloneqq \{q \in Q : \mathrm{elem}(q) = j \wedge \mathrm{val}(q) = k\} \text{ and } Q_{j,>k} \coloneqq \bigcup_{k'=k+1}^{r} Q_{j,k'}.$$

我们声称，对于每个 $1 \leq k < \min(\mathbf{C}_0(d, Q_j) + 1, r)$，都存在 $\tau \in \mathbb{N}$，使得对于每个 $\tau' \geq \tau$，以下结论成立：

$$|\mathbf{C}_{\tau'}(d)|_{Q_{j,k}} = 1 \text{ and } |\mathbf{C}_{\tau'}(d)|_{Q_{j,>k}} = \mathbf{C}_0(d, Q_j) - k.$$

在证明该断言之前，先看它如何帮助我们得出结论。如果 $\mathbf{C}_0(d, Q_j) \leq r$，那么这就确认了 d 中存在一个处于 (d, j)-有效状态的代理。注意，如果 $k = r - 1$，那么$Q_{j,>k}$ 实际上是 $Q_{j,r}$。因此，如果 $\mathbf{C}_0(d, Q_j) > r$，那么代理会聚集在 $Q_{j,r}$的某些状态中。在这种情况下，d 中将有多个代理处于 (d, j)-有效状态。现在，让我们对 k应用归纳法来证明该断言。

基础情形（$k = 1$）。由于所有状态都满足 $\mathsf{val}(q) = 1$，根据前面的观察可知，对于 C_i、C'，只要 $\mathsf{C}_i \xrightarrow{*} \mathsf{C}'$，就有 $\mathsf{C}_i(d, Q_j, 1) \geq \mathsf{C}'(d, Q_j, 1)$。注意，规则 (3) 要使 $\mathsf{C}'(d, Q_j, 1)$ 减小，需要两个具有相同值的状态，这意味着 $\mathsf{C}'(d, Q_j, 1) \geq 1$。还要注意，对于任意 $q \in \mathsf{act}(\mathsf{C}_i)$，要么 $\mathsf{val}(q) = 1$，要么 $\mathsf{val}(q) > 1$。根据上述观察，存在 $\ell \in \mathbb{N}$ 和 $\kappa \geq 1$，使得 $|\mathsf{C}_\ell(d)|_{Q_j,1} = |\mathsf{C}_{\ell+1}(d)|_{Q_j,1} = \cdots = \kappa$。如果 $\kappa = 1$，那么 $|\mathsf{C}_\ell(d)|_{Q_j,>1} = \mathsf{C}_0(d, Q_j) - 1$，问题便得以解决。否则，规则 (3) 在 C_ℓ、$\mathsf{C}_{\ell+1}$、..... 中均处于启用状态。根据公平性，这意味着存在 $\ell' \geq \ell$，使得

$$|\mathbf{C}_{\ell'}(d)|_{Q_{j,1}} > |\mathbf{C}_{\ell'+1}(d)|_{Q_{j,1}} \text{ and } |\mathbf{C}_{\ell'}(d)|_{Q_{j,>1}} < |\mathbf{C}_{\ell'+1}(d)|_{Q_{j,>1}},$$

这与假设矛盾。

归纳步骤。假设对于满足 $1 < k < r - 1$ 和 $k \leq \mathbf{C}_0(d, Q_j)$ 的 k，该命题为真。下面证明该命题对 $k+1$ 也成立。如果 $k = \mathbf{C}_0(d, Q_j)$，那么 $k+1 \not< \min(\mathbf{C}_0(d, Q_j)+1, r)$，

因此显然可得结论，因为该命题不必对 $k+1$ 成立。因此，假设 $k < \mathbf{C}_0(d, Q_j)$。根据归纳假设，存在 $v \in \mathbb{N}$，使得对于每个 $v' \geq v$，以下结论成立：

$$|\mathbf{C}_{v'}(d)|_{Q_{j,k}} = 1 \text{ and } |\mathbf{C}_{v'}(d)|_{Q_{j,>k}} = \mathbf{C}_0(d, Q_j) - k.$$

由于值每次递增 1（也就是说，值之间没有间隔），因此至少有一个代理处于状态 $Q_{j,\ k+1}$。假设对于某个 $w \geq v$，存在状态 $p, q \in Q_{j,\ k+1}$，使得$\mathbf{C}_w \geq \boldsymbol{p}_d + \boldsymbol{q}_d$。那么，规则 (3) 已启用，这意味着存在一个配置 $\mathbf{C}'$，使得 $\mathbf{C}_w \xrightarrow{*} \mathbf{C}'$，

$$|\mathbf{C}_w(d)|_{Q_{j,k+1}} > |\mathbf{C}'(d)|_{Q_{j,k+1}} \text{ and } |\mathbf{C}_w(d)|_{Q_{j,>k+1}} < |\mathbf{C}'(d)|_{Q_{j,>k+1}}.$$

根据公平性，这意味着存在某个 w'，使得 $|\mathbf{C}_{w'}(d)|_{Q_{j,\ k+1}} = 1$。由于

$$\begin{aligned} |\mathbf{C}_{w'}(d)|_{Q_{j,>k+1}} &= |\mathbf{C}_{w'}(d)|_{Q_{j,>k}} - |\mathbf{C}_{w'}(d)|_{Q_{j,k+1}} && \text{(by definition of } Q_{j,\cdot}\text{)} \\ &= |\mathbf{C}_{w'}(d)|_{Q_{j,>k}} - 1 \\ &= (\mathbf{C}_0(d, Q_j) - k) - 1 \\ &= \mathbf{C}_0(d, Q_j) - (k+1). \end{aligned}$$

◀

▶ **引理 23.** 存在 $\tau \geq \alpha$，使得对于每个 $\tau' \geq \tau$、$d \in \text{supp}(\mathbf{C}_{\tau'})$，$j \in [1..m]$ 和 $q \in \text{act}(\mathbf{C}_{\tau'}(d)) \cap Q_L$，都有 计数$_j(q) = \min(\mathbf{C}_{\tau'}(d, Q_j)\ r)$。

证明。注意，只有规则（4）会将状态 q 改变为状态 q' 使得对于某个 $j \in [1..m]$，计数$_j(q') \neq$ 计数$_j(q)$。此外，使用规则（4）会导致 计数$_j(q') >$ 计数$_j(q)$。

设 $d \in \mathbb{D}$ 且 $j \in [1..m]$。若 $\mathsf{C}_0(d, Q_j) = 0$，则领导者针对该元素的计数永远不会更新，也就是说，它永远不会遇到其数据项处于状态 q，且 $\text{elem}(q) = j$ 的情况，因此该元素 j的计数始终为 0。否则，设 τ 为引理 22 给出的阈值，设$v \geq \max(\alpha, \tau)$，并设 $q \in \text{act}(\mathsf{C}_v(d))$ 为引理 22 给出的一个 (d, j)-有效状态。若存在$p \in \text{act}(\mathsf{C}_v(d)) \cap Q_L$ 使得 count$_j(p) <$ val(q)，则规则（4）已启用。这意味着存在一个配置 C'，使得 $\mathsf{C}_v \xrightarrow{*} \mathsf{C}'$，并且对于任意 $q' \in \text{act}(\mathsf{C}'(d)) \cap Q_L$，均有 count$_j(q') = \min(\mathsf{C}_0(d, Q_j)\ r)$。因此，根据公平性，存在 v' 使得

$$\forall q' \in \text{act}(\mathbf{C}_{v'}(d)) \cap Q_L, j \in [1..m] : \text{count}_j(q') = \min(\mathbf{C}_0(d, Q_j), r).$$

注意，一旦领导者获得了正确的计数，规则（4）的前置条件便不再满足，因为对于 j 中的 d，没有其他状态能够具有更高的计数。由于这对任意数据项和任意 j 都成立，因此存在一种配置，使得对于任意数据项，该数据项的领导者对所有元素都具有正确的计数。◀

▶ **引理 24.** 对于每个 $\tau \in \mathbb{N}$、$j \in [1..m]$ 和 $q \in$ 活动状态集合$(\mathbf{C}_\tau) \cap Q_L$，使得 角色$(q) > 0$，都有 计数$_j(q) \in T($角色$(q)\ j)$.

证明。初始时，每个状态 $q \in \text{act}(\mathsf{C}_0)$ 都满足 $\text{role}(q) = 0$。注意，只有规则（5）能为状态赋予严格大于零的角色。形式化地，对于数据项 $d \in \mathbb{D}$、角色 $i \in [1..n]$ 以及配置 C_τ、C'，若 $\mathsf{C}_\tau \xrightarrow{t} \mathsf{C}'$ 且 $\mathsf{C}' = \mathsf{C}_\tau - \boldsymbol{q}_d + \boldsymbol{q}'_d$，其中某些 q、$q' \in Q$ 和转移 t 是由规则（5）产生的，则对所有 $j \in [1..m]$，领导者的状态 q 必须满足 count$_j(q) \in T(i, j)$。因此，领导者必须先具备某个角色的正确计数，才能自行赋予该角色。

进一步注意到，规则（4）是唯一会改变领导者计数的规则。形式上，对于数据项 $d \in \mathbb{D}$、角色 $i \in [1..n]$ 以及配置 $\mathbf{C}_\tau$，$\mathbf{C}'$，满足 $\mathbf{C}_\tau \xrightarrow{t} \mathbf{C}'$ 且

$\mathbf{C}' = \mathbf{C}_\tau - \boldsymbol{q}_d + \boldsymbol{q}'_d$，对于某个 q，$q' \in Q$，以及由规则（4）产生的转移 t，该领导者必须进入新状态 q'，使得 role(q') < 0，前提是它不再满足相应区间的计数条件。因此，更新计数后，领导者要么具有严格为负的分配角色，要么仍满足其先前承担的角色。

因此，对于任意 $\tau \in \mathbb{N}$，所有角色严格为正的领导者在 $\mathbf{C}_\tau$ 中都具有其角色对应的正确计数。◀

▶ **引理 A4.** 对于每个 $\tau \in \mathbb{N}$ 和 $q \in$ 活动状态集合($\mathbf{C}_\tau$) ∖ Q_L，都有 角色(q) ≤ 0。

证明。注意，只有规则（5）能够为某个状态赋予严格为正的角色。形式化地，对于数据项 $d \in \mathbb{D}$ 以及配置 $\mathbf{C}_\tau, \mathbf{C}'$，若满足 $\mathbf{C}_\tau \xrightarrow{(5)} \mathbf{C}'$ 且 $\mathbf{C}' = \mathbf{C}_\tau - \boldsymbol{q}_d + \boldsymbol{q}'_d$，其中某个 $q, q' \in Q$，则状态 q 必须满足 lead(q)。因此，非领导者永远不会被直接赋予严格为正的角色。

还要注意，只有规则 (1) 会移除某个状态的领导者身份。形式化地，对于数据项$d \in \mathbb{D}$ 以及配置 $\mathbf{C}_\tau, \mathbf{C}'$，若满足 $\mathbf{C}_\tau \xrightarrow{(1)} \mathbf{C}'$ 且 $\mathbf{C}' = \mathbf{C}_\tau - \boldsymbol{q}_d + \boldsymbol{q}'_d$，其中 $q, q' \in Q$，则状态 q' 必须满足 role(q') < 0。因此，非领导者在失去领导者身份时会被赋予严格为负的角色。注意，领导者身份只会被移除一次。

因此，非领导者永远不可能具有严格为正的分配的角色。◀

▶ **引理 26**。存在某个 $\tau \in \mathbb{N}$，使得对于每个 $\tau' \geq \tau$ 和 $q \in$ 活动状态集合($\mathbf{C}_{\tau'}$) ∖ Q_L，都有 角色(q) $= 0$。

证明。由引理 A4 可知，非领导者不可能具有严格为正的分配的角色。令 $Q_0 := \{q \in Q\colon \neg$ 领导者(q) $\wedge$ 角色(q) $= 0\}$。注意，只有规则（8）会改变非领导者的角色，并且会将其改为 0。再结合移除领导权是不可逆的这一事实，可知 $|\mathbf{C}_j|_{Q_0} \geq |\mathbf{C}_i|_{Q_0}$ 对每个 $j > i$均成立。

假设对于某个 $v \in \mathbb{N}$，存在某个状态 $q \in$活动状态集合(C_v)，使得 ¬领导者(q) 成立，且角色(q) < 0。令 $i := |$角色(q)$|$。于是，规则（7）在 C_v 中已启用。这意味着存在一个配置 C' 和一个状态 $q \in$ 活动状态集合(C') $\cap\, Q_C$，使得 $\mathsf{C}_v \xrightarrow{(7)} \mathsf{C}'$，且任务$_i$($q$) $=$ 为假。规则（8）在 C' 中已启用。这意味着存在一个配置 C''，使得 $\mathsf{C}' \xrightarrow{*} \mathsf{C}''$，且 $|\mathsf{C}_v|_{Q_0} < |\mathsf{C}''|_{Q_0}$。

根据公平性，某个配置 $\mathbf{C}_{v'}$ 也满足此条件。因此，对于某个 $\tau \geq v'$，有 $|\mathbf{C}_\tau|_{Q_0} = |\mathbf{C}_\tau|_{Q \setminus Q_L}$。既然已经证明，没有规则会将非领导者的角色改为非 0 值，这就意味着对于任意 $\tau' \geq \tau$，非领导者都处于状态 $q \in Q_0$。◀

▶ **引理 27.** 存在 $\tau \geq \alpha$，使得对于每个 $\tau' \geq \tau$ 和 $q \in$ 活动状态集合($\mathbf{C}_{\tau'}$)，都有 控制器(q) $\in \{0,1\}$。

证明。注意，规则 (2) 是唯一赋予严格负控制器值的规则。我们知道，从 $\mathbf{C}_\alpha$出发，存在唯一的控制器。令 $Q_{-1} := \{q \in Q : \text{控制器}(q) = -1\}$。注意，对于任意 $j > i \geq \alpha$，由于规则 (2) 未启用，因此有 $|\mathbf{C}_j|_{Q_{-1}} \leq |\mathbf{C}_i|_{Q_{-1}}$。

假设对于某个 $v \geq \alpha$，存在一个状态 $q \in$ 活动状态集合($\mathbf{C}_v$) 满足 ctrl(q) $= -1$。那么，我们知道规则 (9) 在 $\mathbf{C}_v$ 中已启用，这意味着存在一个配置 $\mathbf{C}'$，使得 $\mathbf{C}_v \xrightarrow{(9)} \mathbf{C}'$ 并且

$$\bigwedge_{i \in [1..m]} \neg \mathrm{task}_i(q) \qquad \text{for some } q \in \mathrm{act}(\mathbf{C}') \cap Q_C$$

这意味着，对于$\mathbf{C}'$，规则 (10) 已启用。因此，存在一个配置$\mathbf{C}''$，使得$\mathbf{C}' \xrightarrow{(10)} \mathbf{C}''$ 且 $|\mathbf{C}_v|_{Q_{-1}} > |\mathbf{C}''|_{Q_{-1}}$。因此，根据公平性，存在 $\tau \geq v$，使得 $|\mathbf{C}_\tau|_{Q_{-1}} = 0$。由于没有规则能将控制器为 ctrl($q$) $= 0$ 的状态 q 转变为控制器为 ctrl(q) $\neq 0$ 的状态 q'，因此这对任意 $\tau' \geq \tau$ 都成立。◀

▶ **引理 28.** 对于每个 $\tau \geq \gamma$、 $i \in [1..n]$ 和 $q \in$ 活动状态集合$(\mathbf{C}_\tau) \cap Q_C$，只要任务$_i(q)$ 成立，就存在 $q' \in$ 活动状态集合$(\mathbf{C}_\tau)$，使得 |角色$(q')| = i$。

证明。注意，只有规则 (7) 能将 真 分配给控制器的某个任务 i。令$R_i := \{q \in Q:$ |角色$(q)| = i\}$。

令 $i \in [1..n]$，并令 $q \in Q_C$ 满足 ¬任务$_i(q)$，且 $q' \in Q_C$ 满足任务$_i(q')$。令某个 $v \geq \gamma$ 满足 $\mathbf{C}_v \xrightarrow{(7)} \mathbf{C}_{v+1}$、$\mathbf{C}_{v+1} = \mathbf{C}_v - \boldsymbol{q}_d + \boldsymbol{q'}_d$。规则 (7) 蕴含存在某个状态 $p \in$ 活动状态集合$(\mathbf{C}_{v+1}) \cap R_i$。注意，对于 $\mathbf{C}_\gamma$，已经永久选出唯一的控制器。

令 $w \geq v + 1$。对于 $\mathbf{C}_w \to \mathbf{C}_{w+1}$ 满足 $|\mathbf{C}_w|_{R_i} > |\mathbf{C}_{w+1}|_{R_i}$，我们必须有 $\mathbf{C}_w \xrightarrow{(8)} \mathbf{C}_{w+1}$。要实现这一点，必须存在某个状态 $q'' \in$ 活动状态集合$(\mathbf{C}_w) \cap Q_C$，使得¬任务$_i(q'')$成立。因此，对于任意 $\mathbf{C}_w$，只要存在一个状态 $q^* \in$ 活动状态集合$(\mathbf{C}_w) \cap Q_C$满足任务$_i(q^*)$，就存在一个状态 $p \in$ 活动状态集合$(\mathbf{C}_w) \cap R_i$。 ◀

▶ **引理 29.** $\psi(\mathbf{C}_0)$ 成立，当且仅当存在某个 $\tau \geq \gamma$，使得对于每个$\tau' \geq \tau$，都存在 $q \in$ 活动状态集合$(\mathbf{C}_{\tau'}) \cap Q_C$，使得 $\bigwedge_{i \in [1..n]}$ 任务$_i(q)$ 成立。

证明 $\Rightarrow$)。根据定义，$\mathbf{C}_\beta$ 有一个唯一当选的控制器，并且每个数据项都有一个唯一的领导者，该领导者携带其数据项初始状态的正确计数。由于 $\psi(\mathbf{C}_0)$ 为 真，这些领导者中的一个子集可以同时履行这些 n 角色。虽然在$\mathbf{C}_\beta$中角色的分配可能不正确，但我们将证明，存在一个从$\mathbf{C}$ 可达的配置 $\mathbf{C}_\beta$，使得控制器的所有任务都为真。下面考虑以下引理及其推论：

- 根据引理 24，具有分配角色的领导者总是认为自己能够履行该角色；根据引理 26，所有非领导者最终都会将自己的角色设为 0，只留下当选的领导者（可能）具有分配的角色；根据引理 27 和引理 28，最终，唯一的控制器只有在某个领导者的角色被设为 $\pm i$ 时，才会有一个设为真的任务 i。

因此，对于配置$\mathbf{C}_\gamma$，只有领导者具有非零角色。这些领导者也正确认为 自己能够履行分配给自己的角色。最后，存在唯一的控制器，且没有代理处于负控制器状态。

现在，我们将有效赋值 A 定义为单射函数 A: $[1..n] \to \mathbb{D}$，满足：若 $A(i) = d$，则 d 与角色 i相匹配。对于满足$\mathbf{C}_0 \xrightarrow{*} \mathbf{C}$ 的配置$\mathbf{C}$ 以及有效赋值 A，令 $s_A(\mathbf{C})$ 表示相对于 A，配置$\mathbf{C}$ 中正确分配的颜色集合，即

$$s_A(\mathbf{C}) := \{d \in \mathbb{D} \colon (\exists i \in [1..n] : A(i) = d) \wedge (\exists q \in Q_L : \mathbf{C}(d, q) > 0 \wedge \mathrm{role}(q) = i)\}.$$

此外，若配置$\mathbf{C}$ 满足以下条件，我们称其为固定的：

- $\mathbf{C}_\gamma \xrightarrow{*} \mathbf{C}$,
- 存在 $q \in$ 活动状态集合$(\mathbf{C}) \cap Q_C$，使得 $\bigwedge_{i=1}^{n}$任务$_i(q)$，并且
- $\mathrm{role}(q) \geq 0$ f$^{\text{或对于每个 } q \in \text{活动状态集合}}(\mathbf{C})$.

请注意，严格为负的角色只能通过规则 (1)（领导者选举）、规则（4）（计数变化）和规则（6）（自身重新分配）进行分配。注意，规则 (1)、(4) 和 (6) 对固定配置均不可用。我们还知道，控制器只有在通过规则 (7) 观察到负角色，或通过规则 (9) 观察到负控制器时，才会将其某个任务更改为$false$。由于固定配置不存在严格为负的角色，也无法生成此类角色，因此规则 (7) 无法将控制器的任何任务更改为$false$。此外，规则 (9) 对固定配置不可用。因此，固定配置只能到达固定配置。

现在，设 A 为一个有效赋值。对于任意 $v \geq \gamma$，我们观察到两种情况。

情况： $\exists A' : |s_{A'}(\mathbf{C}_v)| = n$。 *存在一个固定的配置 C'*，可从 $\mathbf{C}_v$ 到达。

情况： $\forall A'$：$|s_{A'}(\mathbf{C}_v)| < n$。这意味着某个角色 i 没有被任何领导者担任。如果 $A(i)$ 的领导者没有分配的角色，那么根据规则（5），存在一个配置 $\mathbf{C}'$，可从$\mathbf{C}_v$到达，使得 $|s_A(\mathbf{C}_v)| < |s_A(\mathbf{C}')|$，并且对于某个 $q \in \mathsf{act}(\mathbf{C}'(A(i))) \cap Q_L$，$\mathsf{role}(q) = i$。

否则，我们观察到两种情况：控制器的任务 i 被设为假，因此可以通过规则（6）自行

- 重新分配 $A(i)$ 的领导者。控制器的任务 i 被设为真。由于没有任何领导者担任角色 i，根据引理 28，若要使控制器的任务 i 被设为真，则必定存在某个 $q \in \mathsf{act}(\mathsf{C}_v) \cap Q_L$ 满

- 足 $\mathsf{role}(q) = -i$。因此，根据规则（7），存在一个可从 C_v 到达的配置 C'，使得对于某个 $q \in \mathsf{act}(\mathsf{C}') \cap Q_C$，$\neg \mathsf{task}_i(q)$。由此可通过规则（6）自行重新分配 $A(i)$ 的领导者。

因此，还存在一个可从 $\mathbf{C}'$ 到达的配置 $\mathbf{C}''$，使得对于某个 $q \in \mathsf{act}(\mathbf{C}''(A(i))) \cap Q_L$，$|s_A(\mathbf{C}_v)| \lhd s_A(\mathbf{C}'')|$，并且 $\mathsf{role}(q) = i$。

根据公平性，由于 $\mathsf{supp}(\mathbf{C}_0)$ 的配置数量有限，且有效赋值数量有限，因此要么对于某个 v, $|s_A(\mathbf{C}_v)| = n$ 成立，并且根据公平性，$\mathbf{C}_v$ 会到达一个固定配置 $\mathbf{C}_\tau$；要么对于某个 v 和某个有效赋值 $A' \neq A$,$|s_{A'}(\mathbf{C}_v)| = n$ 成立，而根据公平性，这同样会到达一个固定配置 $\mathbf{C}_\tau$。我们知道，固定配置只能到达固定配置，而且还保证控制器的任务列表已完全赋值，因此对于任意 $\tau\prime \geq \tau$，这都成立。◀

⇐ **的证明**⇐）。我们证明逆否命题。假设 $\psi(\mathbf{C}_0)$ = 为假。令 $W(\mathbf{C})$ 为集合 of 数据 $d \in \mathsf{supp}(\mathbf{C}_0)$，使得对于某个 $q \in \mathsf{act}(\mathbf{C}(d)) \cap Q_L$，

- $|\mathsf{role}(q)| = i$ 对某个 $i \in [1..n]$ 成立，并且
- $\mathbf{C}(d, Q_j) \notin T(i, j)$ 对某个 $j \in [1..m]$ 成立。

非正式地说，$W(\mathsf{C})$ 是 C 中具有角色分配错误（正向或负向）的领导者的数据集合，也就是说，该角色是该数据无法匹配的角色。由于 $|\mathsf{C}_\beta|_{Q_L} = |\mathsf{supp}(\mathsf{C}_0)|$，且领导者的数量永远不会增加，因此对于任何满足 C' 的 $\mathsf{C}_\beta \xrightarrow{*} \mathsf{C}'$，都有 $W(\mathsf{C}') \leq |\mathsf{supp}(\mathsf{C}_0)|$。此外，根据引理 23，在 C_β之后，领导者的计数是稳定且正确的。这意味着，对于任何 $t \geq \beta$ 和数据项 d，如果规则（5）将角色 i 分配给 d 的领导者，那么d 就与角色 i匹配。因此，对于任何 i、$j \geq \beta$，只要满足 $i \leq j$，就有 $W(\mathsf{C}_j) \leq W(\mathsf{C}_i)$。

假设存在某个 $\pi \geq \beta$，使得 $W(\mathbf{C}_\pi) > 0$。那么，对于某个 $d \in \mathsf{supp}(\mathbf{C}_0)$，存在状态 $q \in \mathsf{act}(\mathbf{C}_\pi(d)) \cap Q_L$，满足 $|\mathsf{role}(q)| = i$，其中 i 是一个 d无法匹配的角色。由于 $\pi \geq \beta$，根据引理 23，状态 q 必须满足 $\bigwedge_{j \in [1..m]} \mathsf{count}_j(q) = \min(\mathbf{C}_0(d, Q_j)\ r)$，也就是说，对于 d，每个元素的计数都必须正确。

因此，根据引理 24，$\mathsf{role}(q)$ 必须严格为负。这意味着规则（7）在 $\mathbf{C}_\pi$ 中已启用。因此，存在 $\mathbf{C}'$ 和 $q \in$ 活动状态集合$(\mathbf{C}') \cap Q_C$，使得 $\mathbf{C}_\pi \xrightarrow{*} \mathbf{C}'$，且 $\neg$任务$_i(q)$ 成立。注意，规则（8）现在在 $\mathbf{C}'$ 中已启用。因此，存在一个配置 $\mathbf{C}''$，使得 $\mathbf{C}' \xrightarrow{(8)} \mathbf{C}''$，且 $W(\mathbf{C}'') < W(\mathbf{C}_\pi)$。因此，根据公平性，存在 $u \geq \pi$，使得$W(\mathbf{C}_u) = 0$。

根据引理 28，对于任意 $v \geq \beta$，对于所有 $q \in \mathsf{act}(\mathbf{C}_v) \cap Q_C$，如果任务$_i(q)$ 成立，则存在某个状态 $q' \in \mathsf{act}(\mathbf{C}_v) \cap Q_L$，使得角色$(q') = \pm i$。回顾引理 26，任何非领导者最终都会具有角色 0。此外，请注意，

- 对于上文所述的$\mathbf{C}_u$，对于任意 $d \in \mathbb{D}$，如果 d 的领导者具有分配的角色 i，那么 d 匹配角色 i；并且对于每个 $w \geq \max(u, \gamma)$，由于 $\psi(\mathbf{C}_0)$ 为假，不存在有效赋值 A 使

- 得$|s_A(\mathbf{C}_w)| = n$。

因此，不存在满足以下条件的 $\mathbf{C}_w$，对所有 $i \in [1..n]$ 均成立：

$\exists d \in \mathsf{supp}(\mathbf{C}_0)\ q \in$ 活动状态集合$(\mathbf{C}_w(d)) \cap Q_L$：|角色$(q)| = i$.

因此，存在 $\tau \in \mathbb{N}$，使得对于任意满足 $\mathbf{C}'$ with $\mathbf{C}_\tau \xrightarrow{*} \mathbf{C}'$ 的对象，都存在 $i \in [1..n]$，使得对于每个 $q \in$ 活动状态集合$(\mathbf{C}') \cap Q_C$，¬任务$_i(q)$ 均成立。◄

► **推论30**。存在 $\tau \in \mathbb{N}$ 使得 $O(\mathbf{C}_\tau) = O(\mathbf{C}_{\tau+1}) = \cdots = \psi(\mathbf{C}_0)$。

证明。情形：$\psi(\mathsf{C}_0) =$ 真。令 $Q_{true} :=$ {$q \in Q$: out(q)}。注意，对于从 C_α可达的任何配置，都只存在一个控制器。回顾引理 29 的证明中固定配置的概念。对于固定配置 C，如果存在一个状态 $q \in$act(C) 使得 ¬out(q成立，由于控制器的任务列表已完全分配，因此存在一个可从 C′ 到达的配置 C，使得 $|\mathsf{C}|_{Q_{true}} < |\mathsf{C}'|_{Q_{true}}$。注意，控制器在固定配置中绝不会传播假输出；由此可知，对于任意可从 C* 到达的配置 C_v，都有 $|\mathsf{C}_v|_{Q_{true}} \leq |\mathsf{C}^*|_{Q_{true}}$。利用公平性可知，还存在一个可从 C′ 到达的配置 C″，使得 $|\mathsf{C}''|_{Q_{true}} = |\mathsf{C}''|$。从引理 29 的证明中，我们还知道，如果 $\psi(\mathsf{C}_0) =$ 为真，那么最终会到达一个固定配置。因此，会到达某个配置 C_τ，使得 $O(\mathsf{C}_\tau) =$ 为真；对于任意满足 $\mathsf{C}_\tau \xrightarrow{*} \mathsf{C}_{\tau'}$的配置 $\mathsf{C}_{\tau'}$，它仍为真。

情况：$\psi(\mathsf{C}_0) =$假。根据引理 29，我们知道，对于某个 $u \in \mathbb{N}$，以及任意 $v \geq u$，控制器在 C_v 中至少有一个未分配的任务。假设存在某个 $v \geq u$，使得某个 $q \in$活动状态集合(C_v) 满足 out(q)。那么，由于控制器至少有一个未分配的任务，规则（11）处于启用状态（控制器传播假输出），并且存在一个从 C_v 可达的配置 C′，使得 $|\mathsf{C}_v|_{Q_{true}} > |\mathsf{C}'|_{Q_{true}}$。注意，对于 C_v，控制器从不传播真输出；因此，对于任意从 C_v 可达的配置 C*，都有 $|\mathsf{C}_v|_{Q_{true}} \geq |\mathsf{C}^*|_{Q_{true}}$。因此，根据公平性，这意味着存在一个从 C′ 可达的配置 C″，使得 $|\mathsf{C}''|_{Q_{true}} = 0$。于是，某个满足 $O(\mathsf{C}_\tau) =$ 假 的配置 C_τ 将会被到达。任何从 C_τ 可达的配置 $\mathsf{C}_{\tau'}$ 也满足 $O(\mathsf{C}_\tau) =$ 假。◄