具有遗漏故障的最优最终拜占庭一致协议

KAYA ALPTURER,美国康奈尔大学JOSEPH Y.
HALPERN,美国康奈尔大学RON VAN DER
MEYDEN,澳大利亚悉尼新南威尔士大学

arXiv:2305.06271v1 [cs.DC] 10 May 2023

关于最优最终拜占庭一致性（EBA）协议的研究——这类协议以精确意义上的方式，在每次运行中尽可能早地作出决定，并保证所有非故障代理作出相同的决定——一直聚焦于全信息协议（FIP）：代理反复向其他每个代理发送消息，完整描述其过去的观察结果。虽然可以证明，不失一般性，我们可以将最优协议视为 FIP，但由于所需的消息规模，完全信息交换在许多应用中难以实现。我们将协议分为两部分：信息交换协议和行动协议，以便考察信息交换受限所带来的影响。随后，我们定义相对于信息交换协议的最优性概念。粗略地说，若信息交换协议为 E，且 P，那么行动协议 P 相对于 E 是最优的，即在遵循 E 进行信息交换的行动协议中，代理能尽可能早地作出决定。我们提出了一个适用于遗漏故障的基于知识的 EBA 程序，其所有实现都保证正确；如果信息交换满足某个安全条件，这些实现也都是最优的。接着，我们构造了在两个重要场景中实现这一基于知识的程序的具体程序，并证明这两个场景都满足该安全条件。最后，我们证明，只需对程序稍作修改，就能得到一个既最优又可高效实现的 FIP，从而解决 Halpern、Moses 和 Waarts 提出的一个开放问题（SIAM J. Comput., 2001）。

CCS Concepts: • **Theory of computation → Distributed algorithms**; • **Computing methodologies** → *Reasoning about belief and knowledge*; • **Computer systems organization → Dependable and fault-tolerant systems and networks**.

Additional Key Words and Phrases: Distributed algorithms, Epistemic logic, Reasoning about knowledge, Byzantine Agreement, Consensus, Fault tolerance

1 INTRODUCTION

Logics of knowledge (epistemic logics) have been shown to provide useful abstractions for reasoning about distributed systems [5, 7], enabling a focus on the information that needs to be attained in order to perform certain actions, independent of how that information is encoded in the states of the system. The approach has, in particular, been used fruitfully to analyze agreement protocols, where agents are required to make consistent decisions on some value, based on their initial preferences [1, 4, 8, 12].

本研究的一个重点是开发能够最优利用信息的协议。在一致性协议的语境下，这意味着协议应尽可能早地作出决定。粗略地说，如果对于每一种可能的故障模式和输入，P_1 支配协议 P_2 作出决定的时间都不晚于 P_1，我们就称协议 P_1 支配协议 P_2。最优协议是支配所有其他协议的协议；最优协议则是不被任何其他协议支配的协议。针对存在崩溃故障的同步拜占庭一致性（SBA），已有最优的基于知识的程序（即显式测试知识的程序）

Authors' addresses: Kaya Alpturer, ki78@cornell.edu, Cornell University, Ithaca, NY, USA; Joseph Y. Halpern, halpern@cs.cornell.edu, Cornell University, Ithaca, NY, USA; Ron van der Meyden, R.VanderMeyden@unsw.edu.au, UNSW Sydney, Sydney, NSW, Australia.

[1, 2, 4] 以及存在遗漏故障的情形，针对存在遗漏故障的一致同步拜占庭一致性（SBA）也已有此类程序 [13] （即所有作出动作决定的代理都必须执行相同的动作，而不仅仅是非故障代理）。此外，这些程序还已有多项式时间实现。对于最终拜占庭一致性（EBA）问题，众所周知，不存在最优协议 [12]，尽管存在最优协议。针对存在崩溃故障 [1] 和发送遗漏故障 [8]的情况，已有适用于 EBA 的最优基于知识的程序。虽然针对崩溃故障情况，已有该基于知识程序的多项式时间实现 [1]，但针对遗漏故障情况则尚无此类实现。事实上，寻找适用于存在遗漏故障时 EBA 的多项式时间最优算法这一问题，已悬而未决二十多年。本文解决了这一问题，此外还取得了其他成果。

The work on optimal (and optimum) protocols has focused on *full-information protocols* (FIPs), ones where each agent repeatedly sends all other agents its complete state, containing its initial state and all messages that it has received up to the present time. As far as optimal protocols go, there is no loss of generality in considering only FIPs. As is well known [3], any protocol can be simulated by an FIP, so for any protocol, there is an FIP that decides at least as soon. However, while FIPs do provide optimality, they are expensive to implement due to their space requirements and the length of messages sent, their analysis may be complex, and in some failure environments, they may require that intractable properties be computed at each step of the protocol to attain optimality. They are therefore not necessarily practical. The present paper is part of a program of research in which we seek to overcome these difficulties with the full-information paradigm by considering protocols in which less than full information is exchanged between the agents.

本文旨在考察信息交换受到更多限制时所产生的影响。为此，我们将协议分为两部分：信息交换协议规定代理在其局部状态中保留哪些信息，以及每一步交换什么消息；行动协议则在一致性协议的情况下规定代理作出什么决定。接着，我们定义相对于信息交换协议的最优性概念。粗略地说，对于某个特定规范，若行动协议 P 能使代理尽可能早地作出决定，且与任何满足该规范并按照 E 交换信息的其他协议相比都不逊色，那么该行动协议相对于信息交换协议 E 就是最优的。全信息协议是信息交换协议的一种特殊情形，但我们也允许限制更多的协议，在这些协议中，代理跟踪的信息较少，发送的消息也较少。本文关注二元 EBA 的最优协议，其中代理只有两种可能的初始偏好；我们假设存在发送遗漏故障。对于 EBA，有一组代理，每个代理的初始偏好为 0 或 1。要求无故障代理就某个值达成一致，但不同代理可以在不同时间作出决定。解还必须满足非平凡性，即如果所有初始偏好都是 $x \in \{0,1\}$，那么无故障代理必须作出决定 x。

由于在 EBA 中，代理不必同时作出决定，因此文献通常会考虑偏向 0 的最优协议：代理一旦得知某个代理的初始偏好为 0，就立即作出 0 的决定；也会考虑偏向 1 的协议。在崩溃故障的情况下，已知存在按此意义偏向 0（或 1）的最优 EBA 协议 [1]。但很容易看出，在遗漏故障的情况下，不可能存在偏向 0（或 1）的最优 EBA 协议。考虑这样一种情形：至少有三个代理，且某次运行 r 中恰有一个代理发生故障，例如代理 1，其余代理的

[1]Recall that with *crash failures*, a faulty process behaves according to the protocol, except that it might crash at some round (possibly after sending some messages), after which it sends no messages; with sending-omission failures, a faulty process may omit to send an arbitrary set of messages in any given round, but otherwise behaves according to the protocol.

初始偏好均为 1，且故障代理不发送任何消息。在运行 r 中，非故障代理最终必须作出 1 的决定，因为故障代理的初始偏好可能为 1，而在这种情况下，EBA 要求作出 1 的决定。假设在运行 r 中，第一个作出决定的非故障代理在第 n 轮作出决定；不失一般性地，设代理 2 是在第 n 轮作出 1 的决定的非故障代理。现在考虑运行 r'：代理 1 的初始偏好为 0，且发生故障；其余代理均无故障且初始偏好为 1；代理 1 直到第 $n-1$ 轮都不发送任何消息，并在第 $n-1$ 轮只发送一条消息给代理 3，真实地告知代理 1 的初始偏好为 0。由于代理 2 无法区分 r 与 r'，代理 2 必须在 r' 的第 n 轮作出 1 的决定。由于代理 3 在 r 的前 $n-1$ 轮没有作出决定，因此它在 r' 的前 $n-1$ 轮也不会作出决定。代理 3 在第 $n-1$ 轮得知某个代理的初始偏好为 0，因此在 r' 的第 n 轮作出 0 的决定。于是，在 r' 中，两个非故障代理作出了不同的决定，因此 EBA 未能实现。（注意，类似 r' 的运行与崩溃故障不相容；它确实需要遗漏故障。）

为了解决这个问题，在偏向 0 的协议中，我们不再要求代理一听到有关 0 的消息就立即对 0 作出决定，而只要求代理仅在通过一条代理链听到有关 0 的消息时才对 0 作出决定（链中的第一个代理的初始偏好为 0，并且在第 k 轮，链中的第 k 个代理对 0 作出决定，并将此事告知第 $(k+1)$ 个代理）。请注意，在崩溃故障的情况下，代理只能通过这样一条链听到有关 0 的消息。接着，我们基于这一（众所周知的）想法提出一个基于知识的行动协议 $\mathbf{P}^0$，并证明该协议是正确的（也就是说，无论使用何种信息交换协议，其所有实现都满足 EBA 规范），且在满足某一安全条件的上下文中是最优的。然后，我们考察两种代理之间交换相对短（且数量相对少）的消息的信息交换协议，并证明它们满足该安全条件。最后，我们给出具体的多项式时间行动协议，这些协议针对上述两种信息交换协议实现 $\mathbf{P}^0$。

The knowledge-based program $\mathbf{P}^0$ is not optimal in the case of full-information contexts, but, as we show, a small modification $\mathbf{P}^1$ of $\mathbf{P}^0$ is optimal. Moreover, $\mathbf{P}^0$ and $\mathbf{P}^1$ are equivalent in the two limited-information contexts that we considered, so $\mathbf{P}^1$ is also optimal in these contexts. Roughly speaking, $\mathbf{P}^1$ allows agents to decide if they get common knowledge of who the nonfaulty agents are. Such common knowledge can't be obtained in the limited-exchange contexts that we consider, but it can be obtained if agents use a full-information protocol and keep track of everything they have learned. The knowledge-based program $\mathbf{P}^1$ does not involve the (rather complicated) *continual common knowledge operator* used by Halpern, Moses, and Waarts; rather, it uses more standard knowledge and common knowledge operators. This allows us to provide a polynomial-time implementation of it.

本文其余部分组织如下。第 2 节回顾[5]的认知框架。第 3 节介绍信息交换协议与动作协议的分离，以及故障模型的表示。第 4 节定义基于知识的程序。EBA 的规范以及相对于信息交换协议的最优性的形式化定义见第 5 节。在第 6 节中，我们定义基于知识的程序 $\mathbf{P}^0$,，证明它满足 EBA，定义足以使 $\mathbf{P}^0$ 达到最优性的安全条件，描述两个满足该安全条件的自然的有限信息交换协议，并给出相对于这两个信息交换协议实现 $\mathbf{P}^0$ 的具体动作协议。在第 7 节中，我们定义 $\mathbf{P}^1$,，即 $\mathbf{P}^0$ 的一个修改版本，该版本对于全信息交换协议是最优的，并给出其多项式时间实现。第 8 节讨论有限信息交换的代价。本文所有结果的证明均见附录。

2 语义模型

我们假设一个代理 集合通过消息传递网络进行通信，该网络可能会遭受各种类型的故障。为了对这类系统进行语义建模，我们使用标准的运行与系统模型 [5, 6], , 下面简要回顾该模型。

解释系统 [5] 对多代理场景进行建模，其中一定数量的代理 n 会随时间改变其状态。解释系统是一个二元组 $\mathrm{I} = (\mathrm{R}\ \pi)$，其中 R 是运行的集合，用于描述系统如何随时间演变； $\pi : \mathrm{R} \times \mathbb{N} \to \mathrm{P}(Prop)$ 是一个解释函数，用于指出系统的每个点上哪些原子命题为真。点是一个有序对 (r, m)，由一个运行 $r \in \mathrm{R}$ 和一个时间 $m \in \mathbb{N}$组成。集合 R 称为系统。形式化地，运行 $r \in \mathrm{R}$ 是一个函数 $r : \mathbb{N} \to L_e \times \Pi_{i \in \mathrm{Agt}} L_i$，其中 L_e是代理所处环境的可能局部状态的集合，而每个 L_i 是代理 i的可能局部状态的集合。$L_e \times \Pi_{i \in \mathrm{Agt}} L_i$中的元素称为全局状态。给定一个运行 r、代理 i和时间 m，我们用 $r_i(m)$ 表示 $r(m)$ 的第 $i+1$ 个分量，它是全局状态 $r(m)$ 中代理 i的局部状态；用 $r_e(m)$ 表示 $r(m)$ 的第一个分量，它是环境的局部状态。

为了推理解释系统中代理的知识，我们使用一种用于推理知识和时间的标准语言。我们从原子命题的集合 Φ 开始，并在 $\wedge$、 $\neg$、认知算子 K_i（分别对应$, \ldots, n$，每个代理各有一个）和 $C_{\mathcal{S}}$（解释系统中索引集合 S内代理之间的共同知识；见下文），以及时态算子 $\Box$、$\boxdot$、$\bigcirc$和 $\ominus$下闭合。公式 $K_i\phi$ 表示代理 i 知道公式 ϕ 成立； $\Box\phi$ 表示 ϕ 在未来所有时刻都成立；$\boxdot\phi$ 表示 ϕ 始终成立；$\bigcirc\phi$ 表示 ϕ 在下一时刻成立；而 $\ominus\phi$表示 ϕ 在上一时刻成立。

该逻辑的语义由关系 $\mathrm{I}, (r, m) \models \phi$给出，其中 I 是一个解释系统，$(r, m)$ 是 I 中的一个点，而 ϕ是一个公式。对于不具有 $C_{\mathcal{S}}\phi$形式的公式，关系 $\models$按如下方式归纳定义（我们省略命题算子的显然情形）：

- $\mathcal{I}, (r, m) \models p$ if $p \in \pi(r, m)$,
- $\mathcal{I}, (r, m) \models K_i\phi$ if $\mathcal{I}, (r'\ m') \models \phi$ for all points $(r'\ m')$ of $\mathcal{I}$ such that $r_i(m) = r'_i(m')$,
- $\mathcal{I}, (r, m) \models \Box\phi$ if $\mathcal{I}, (r, m') \models \phi$ for all $m' \geq m$,
- $\mathcal{I}, (r, m) \models \boxdot\phi$ if $\mathcal{I}, (r, m') \models \phi$ for all $m' \in \mathbb{N}$,
- $\mathcal{I}, (r, m) \models \bigcirc\phi$ if $\mathcal{I}, (r, m+1) \models \phi$
- $\mathcal{I}, (r, m) \models \ominus\phi$ if $m > 0$ and $\mathcal{I}, (r, m-1) \models \phi$.

The intuition for the definition of the knowledge operator $K_i\phi$ is that $r'_i(m) = r_i(m)$ says that agent i considers it possible, when in the actual situation (r, m), that it is in situation $(r'\ m')$, since it has the same local state there. An agent then *knows* ϕ if ϕ is true in all the situations that the agent considers to be possible.

We can now define the modal operator $C_{\mathcal{S}}$. Intuitively, $C_{\mathcal{S}}\phi$ is true at a point (r, m) if ϕ is common knowledge among the agents in S; each of the agents in $\mathcal{S}$ knows that each of the agents in S knows …that ϕ is true. The fact that $\mathcal{S}$ is an indexical set means that its membership can depend on the point; that is, semantically, $\mathcal{S}(r, m)$ is a set of agents for each point (r, m). We define $E_{\mathcal{S}}\phi$ (everyone in $\mathcal{S}$ knows ϕ) as an abbreviation for $\bigwedge_{i \in \mathcal{S}} K_i\phi$. That is,

$$\mathcal{I}, (r, m) \models E_{\mathcal{S}}\phi \text{ if, for all } j \in \mathcal{S}(r, m), \text{ we have } \mathcal{I}, (r, m) \models K_j\phi.$$

Taking $E^1_{\mathcal{S}}\phi$ to be an abbreviation of $E_{\mathcal{S}}\phi$, and $E^{m+1}_{\mathcal{S}}\phi$ to be an abbreviation of $E_{\mathcal{S}}(E^m_{\mathcal{S}}\phi)$, we define

$$\mathcal{I}, (r, m) \models C_{\mathcal{S}}\phi \text{ if, for all } m \geq 1, \mathcal{I}, (r, m) \models E^m_{\mathcal{S}}\phi.$$

As usual, 我们称 ϕ在 I 中有效，并记作 $\mathrm{I} \models \phi$，若对于 I 中的所有点 $(r,\ m)$，均有 $\mathrm{I}, (r,\ m) \models \phi$.

3 通信与故障模型

现在，我们将上一节的一般模型具体化，用以表示遗漏故障模型。在我们的表示中，我们将信息交换协议与行动协议分开：前者刻画代理在其局部状态中保留的信息，以及发送哪些消息、何时发送；后者刻画除发送消息之外执行其他动作的规则。（在我们的情形中，这些动作将是决定。）在相关文献中，信息交换协议通常采用全信息协议，即在每个步骤中，每个代理都会向所有其他代理发送一份完整说明，列出截至该时刻它所获知的一切。不过，我们关注的是信息交换较少的协议，因此，将信息交换协议作为我们所构造的解释系统的一个参数单独处理会很有帮助。另一个参数是故障模型 F，定义如下。

我们假设信息通过发送消息来交换。我们的重点将放在同步的消息传递上，在这种机制中，代理按照一系列同步轮次运行。在每一轮中，每个代理都会执行一些动作，向其他代理发送一组消息，接收其他代理在同一轮中发送的部分消息，并根据这些事件更新自身的状态。信息交换协议描述了代理可能的初始状态（其中可能包含诸如代理对待作出共识决策结果的偏好等信息）、代理如何选择在每个时间点发送的消息，以及代理如何根据接收到的消息更新自身状态。

我们假设每个代理都有一个动作集合 A_i，其中包含它可以执行的动作。在我们的应用中，$A_i = \{\text{decide}_i(x) \mid x \in \{0,1\}\} \cup \{\text{noop}\}$，但一般而言，$A_i$可以是任意的。形式上，对于代理 $Agt = \{1,\ldots,n\}$，一个信息交换协议 E 由一个元组 $E_1,\ldots,E_n$ 给出，该元组由每个代理 i各自的本地信息交换协议 E_i 组成。每个本地信息交换协议 E_i是一个元组 $L_i, I_i, A_i, M_i, \mu_i, \delta_i$，其中

- L_i 是局部状态的集合。
- $I_i \subseteq L_i$ 是初始状态的集合。
- M_i 是代理 i可以发送的消息集合。
- μ_i：$L_i \times A_i \to \Pi_{j\in\text{代理}}(M_i \cup \{\perp\})$ 是一个函数，它将局部状态 s和动作 a映射为当前轮次要发送的消息，每个代理 j各一条。直观地说，$\mu_i(s, a) = \sigma$表示，当在状态 s下执行动作 a时，信息交换协议会向每个代理 j传送消息 σ_j。此处，$\sigma_j = \perp$ 表示 i 没有向 j发送消息。令$\mu_{ij}(s, a)$ 表示此元组中 i发送给 j的消息。
- δ_i：$L_i \times A_i \times \Pi_{j\in\text{代理}}(M_j \cup \{\perp\}) \to L_i$是一个函数，它根据一个动作和消息元组 $(m_1, \ldots\ldots, m_n)$ 更新局部状态，其中消息为 $m_j \in M_j \cup \{\perp\}$（如果 i没有收到来自 j的消息，则 $m_j = \perp$）。

故障模型描述可能发生哪些故障。通常，故障模型带有一个参数 t，用于表示可能发生故障的代理数量上限。故障模式或对手定义了在符合故障模型的特定运行中实际发生的故障。形式上，故障模式 α是一个二元组 $(N\ F)$，其中 $N \subseteq Agt$，且 F是一个映射 $F:\mathbb{N}\times Agt \times Agt \to \{0,1\}$。这里，N 是非故障代理的集合，而 $F(m,i,j)$描述代理 i 在第 $m+1$ 轮发送给代理 j 的消息是否送达。（如果消息未送达，我们假设改为送达消息 $\perp$。）故障模型是故障模式的集合。对于代理集合 Agt，发送遗漏模型 $SO(t)$是所有满足以下条件的故障模式 $(N\ F)$ 的集合：$|Agt - N| \le t$，因此故障代理至多有 t个；并且对于所有 $m \in \mathbb{N}$和 $j \in Agt$，如果 $F(m,i,j) = 0$，那么$i \in Agt - N$。崩溃故障模型是一个特殊情况：如果 $F(m,i,j) = 0$，那么对于所有 $m' > m$和代理 j'，$F(m',i,j') = 0$。

针对信息交换协议 E 的行动协议 P是一个元组 $(P_1, \ldots, P_n)$，其中为每个代理 $i = 1 \ldots n$ 指定一个局部行动协议 P_i： $L_i \to A_i$一个将 E 中代理 i的局部状态 L_i映射到 A_i中动作的映射。

为了将这些定义与第 2 节的语义模型联系起来，我们描述信息交换协议 E、故障模型 F 和动作协议 P 如何确定一个系统 $\mathrm{R}_{\mathcal{E},\mathcal{F},P}$。在该系统中，环境可能局部状态的集合 L_e 由所有可能的故障模式组成。代理的局部状态 L_i 是代理 i的信息交换协议的状态。一个初始状态（$s_e, s_1, \ldots, s_n$）是一个全局状态，其中 $s_i \in I_i$是每个代理i的信息交换协议的初始状态。对于每个初始状态，由信息交换协议 E、故障模型 F 和动作协议 P唯一确定一个具有该初始状态的运行 r。在此运行中，协议 E、故障模式 α和 P依次确定每个步骤中执行哪些动作、发送哪些消息以及接收哪些消息。每个代理根据本轮执行的动作和接收到的消息更新其局部状态。形式上，时间 $k+1$的全局状态 $r(k+1) = (s'_e, s'_1, \ldots, s'_n)$由时间 k 的全局状态 $r(k) = (s_e, s_1, \ldots, s_n)$按如下方式确定：

- $s'_e = s_e$ (so the failure pattern remains unchanged throughout the run).
- For each pair of agents i and j, let $m_{i,j}$ be the message that agent i sends to j, given that it performs action $P_i(s_i)$ in state s_i, that is, $m_{i,j} = \mu_i(s_i, P_i(s_i))(j)$.
- For each pair of agents i and j, let $m'_{i,j}$ be the result of applying the failure pattern to the messages sent. Specifically, suppose that $s_e = (\mathcal{N}\ F)$. If $F(k, i, j) = 0$ then $m'_{i,j} = \perp$ and if $F(k, i, j) = 1$ then $m'_{i,j} = m_{i,j}$.
- Finally, for each agent i, $s'_i = \delta_i(s_i, P_i(s_i), (m'_{1,i}, \ldots, m'_{n,i}))$.

The system $\mathcal{R}_{\mathcal{E},\mathcal{F},P}$ consists of all runs generated from some initial state.

4 基于知识的程序

基于知识的程序规定了在代理所知信息的基础上如何确定其动作。正如 Fagin 等人所定义的那样， [5], 这些程序是在相对于一个解释上下文的情况下进行解释的；该上下文定义了全局状态、如何根据动作更新这些状态，以及对原子命题的解释。在我们的设定中，可以将解释上下文取为一个元组 (E， F π)，其中包括信息交换协议 E、故障模型 F，以及对由 E 和 F 构造的全局状态上所有运行的集合中的原子命题的解释 π。

对我们的目的而言，将基于知识的程序取为如下形式是很方便的：$\mathbf{P} = (\mathbf{P}_1, \ldots, \mathbf{P}_n)$，其中对于每个代理 i，局部基于知识的程序 $\mathbf{P}_i$属于具有如下语法的语言

$$\mathbf{P}_i ::= a_i \mid \text{if } \phi_i \text{ then } \mathbf{P}_i \text{ else } \mathbf{P}_i,$$

其中， a_i 表示代理 i的动作集合 A_i 中的动作，而 ϕ_i 是形如 $K_i\psi$ 的公式的布尔组合。也就是说，代理 i 的局部基于知识的程序中的测试涉及代理 i 的知识。注意，在解释系统 I 中的点 (r, m) 处，这类公式 ϕ_i 的真值仅取决于代理 i 在该点的局部状态。也就是说，对于点 (r, m)、$(r'm')$，若$r_i(m) = r'i(m')$，则 $\mathrm{I}, (r, m) \models \phi_i$ 当且仅当 $\mathrm{I}, (r'\, m') \models \phi_i$。因此，给定代理 i 的局部状态 s，我们可以写作 $\mathrm{I}\ s \models \phi_i$，以表示对于 I 中所有满足 $r_i(m) = s$ 的点 (r, m)，均有 $\mathrm{I}, (r, m) \models \phi_i$。

为了从语义上解释基于知识的程序，我们首先定义：给定解释系统 I，基于知识的程序 $\mathrm{P} = (\mathrm{P1}, \ldots, \mathrm{P}n)$ 如何确定一个具体动作协议 P^I。对于 I 中的每个代理 i 及其局部状态 s（属于代理 i），我们将 $\mathrm{P}_i^I(s)$ 定义为在 I 中的局部状态 s 下执行程序 P_i 所得的动作，其中程序中的测试均按该状态解释。形式上，我们通过对 P 的结构进行归纳法来定义 P^I， 令 $(a_i)^I(s) = a_i$；对于 P_i =“if ϕ_i then Q_i else R_i”，若 $\mathrm{I}\ s \models \phi_i$，则定义 $\mathrm{P}_i^I(s) = \mathrm{Q}_i^I(s)$；否则，定义 $\mathrm{P}_i^I(s) = \mathrm{R}_i^I(s)$。

给定一个基于知识的程序 $\mathbf{P} = (\mathbf{P}_1, \ldots, \mathbf{P}_n)$，以及信息交换协议 E 的具体动作协议 $P = (P_1, \ldots, P_n)$，如果对于 $\mathrm{I} = (\mathrm{R}_{\mathcal{E},\mathcal{F},P}, \pi)$，对于 I 中出现的所有代理 $i = 1, \ldots, n$ 及其局部状态 s（属于代理 i），均有 $P_i(s) = \mathbf{P}_i^{\mathcal{I}}(s)$，则称 P在上下文 $\gamma = (\mathrm{E}, \mathrm{F}\ \pi)$ 中P得以实现。

5 EVENTUAL BYZANTINE AGREEMENT

We briefly review the specification of the eventual Byzantine agreement problem that we consider in this paper. The specification assumes that each agent starts with an independently selected value $init_i \in \{0, 1\}$. The actions in A_i have the form $\mathtt{decide}_i(v)$, where $v \in \{0, 1\}$, as well as a "do-nothing" action noop. We seek protocols (i.e., an information-exchange protocol and an action protocol) such that every run satisfies the following four properties:

- **Unique Decision:** If agent i performs an action $\mathtt{decide}_i(v)$ (for some v), then it does not later perform $\mathtt{decide}_i(1 - v)$.
- **Agreement:** If agents i and j are both nonfaulty, i performs $\mathtt{decide}_i(v)$, and j performs $\mathtt{decide}_j(v')$, then $v = v'$.
- **Validity:** If a nonfaulty agent i performs $\mathtt{decide}_i(v)$ then $init_j = v$ for some agent j.
- **Termination:** For all nonfaulty agents i, eventually i performs $\mathtt{decide}_i(v)$ for some value $v \in \{0, 1\}$.

To relate this specification to our formal model, we define an *EBA context* to be a tuple $(\mathcal{E}, \mathcal{F}\ \pi)$ consisting of an information-exchange protocol $\mathcal{E}$, a failure model $\mathcal{F}$, and an interpretation π of a set *Prop* of propositions, such that the following conditions hold:

- The local states in $\mathcal{E}_i$ have the form $\langle time_i, init_i, decided_i, jd_i, \ldots \rangle$, where $time_i \in \mathbb{N}$, $init_i \in \{0, 1\}$, $decided_i \in \{0, 1, \bot\}$ and $jd_i \in \{0, 1, \bot\}$. Intuitively, $jd_i = v$ if i learned that some agent just decided v, for $v \in \{0, 1\}$.
- The initial local states in $\mathcal{E}_i$ have the form $\langle 0, init_i, \bot, jd_i^0, \ldots \rangle$, where $jd_i = \bot$.
- 消息选择值 $\mu_i(s,\ a)$ 满足以下约束：i在以下三种情况下发送不同的消息：(a) $a =$ 作出决定$_i$(0)，(b) $a =$ 作出决定$_i$(1)，以及 (c) 其余情况。这意味着，j可以根据其从 i收到的消息，判断 i在当前轮次是要决定 0 还是 1。形式化地说，这意味着存在三个互不相交的集合 M^0、M^1和 M^2，且 $\bot \notin M^0 \cup M^1$，使得如果 $a =$ 作出决定$_i$(0)，那么 i向每个代理 j发送一条属于 M^0的消息；如果 $a =$ 作出决定$_i$(1)，那么 i向每个代理 j发送一条属于 M^1的消息；否则，i向每个代理 j发送一条属于 M^2的消息。
- 转移函数 δ_i以状态 s、动作 a和消息元组 $(m_1, \ldots\ldots, m_n)$ 为输入时，会将 s的时间分量 $time_i$增加 1；如果 $a =$ 作出决定$_i(v)$，则将 $decided_i$设为 v，否则保持 $decided_i$ 不变；此外，如果 i 在第 m 轮收到了来自某个代理的消息，且该代理在该轮执行动作"决定$_i$(0)"，则将 $jd_i = 0$ 设为 i；如果 i 在第 m 轮收到了来自某个代理的消息，且该代理在该轮执行动作"决定$_i$(1)"，则将 $jd_i = 1$ 设为 i；否则将 $jd_i = \bot$ 设为 μ_i（关于 μ_i 的假设确保此类消息可与其他消息区分）。请注意，*time* 分量在每一步都会增加 1，这保证了系统是同步的；在时间 m，所有代理 i 的$time_i = m$ 都为 m。

- *Prop* 包含以下至少这些命题集（它们都是定义下述规范所必需的），针对每个代理 $i \in Agt$：$-\ init_i = v$，适用于 $v \in \{0,1\}$；$-\ decided_i = v$，适用于 $v \in \{\bot, 0, 1\}$；$-\ time_i = k$，适用于 $k \in \mathbb{N}$；以及$-\ i \in \mathrm{N}$；

- π 根据 i的局部状态，以显而易见的方式解释$init_i = v$、$decided_i = v$和$time_i = k$（例如，$\pi(r, m)$使$init_i = v$为真，当且仅当 $r_i(m)$的第二个分量$init_i$等于 v）；并根据 F，以显而易见的方式解释 $i \in$N（即，$\pi(r, m)$使 $i \in$N为真，当且仅当 $i \in$N (r)成立，其中 N (r)是运行 r中的非故障代理集合。

An EBA context satisfies some minimal properties that we expect all contexts that arise in the analysis of EBA to satisfy.

We define $decided_i$ to be an abbreviation of $decided_i = 0 \vee decided_i = 1$, take $jdecided_i = v$ to be an abbreviation for $decided_i = v \wedge \ominus decided_i = \bot$ (intuitively, i just decided v), take $deciding_i = v$ to be an abbreviation for $decided_i = \bot \wedge \bigcirc decided_i = v$ (intuitively, i is deciding v in the current round) and take $\exists v$, for $v \in \{0, 1\}$ to be an abbreviation of $\bigvee_{i \in Agt} init_i = v$.

Given an EBA context $\gamma = (\mathcal{E}, \mathcal{F}\ \pi)$ and an action protocol P for $\mathcal{E}$, we define the system $\mathcal{I}_{\gamma,P} = (\mathcal{R}_{\mathcal{E},\mathcal{F},P}, \pi)$. To satisfy the informal specification above, we now seek an EBA-context $\gamma = (\mathcal{E}, \mathcal{F}\ \pi)$ and an action protocol P for $\mathcal{E}$ such that the following are valid in the system $\mathcal{I}_{\gamma,P}$ for all agents i and j:

- **唯一决策：** 已决定$_i = v \Rightarrow \Box\neg$(已决定$_i = 1 - v$)，对于 $v \in \{0,1\}$。
- **一致性：** $\neg$($i \in$N $\wedge\ j \in$N $\wedge$已决定$_i = v \wedge$已决定 $_j = v'$), 对于 $v \neq v'$。
- **有效性：**(已决定$_i = v \wedge i \in \mathcal{N}) \Rightarrow \exists v$
- **终止性：** $i \in \mathcal{N} \Rightarrow \Diamond$(已决定$_i \neq \bot$)。

如果满足这些条件，我们将 P 称为适用于上下文 γ 的 *EBA* 决策协议 。元组 (E P, π) 是故障模型 F 的 *EBA* 协议 。也就是说，在该故障模型中解决 EBA 的协议由信息交换协议、做出决定的动作协议以及对基本命题集的解释组成。

我们关注的是根据由信息交换协议维护的信息而言最优的协议。以下定义将这一概念形式化。两个行动协议 P、P' 的运行 r、r' 若满足 $r(0) = r'(0)$，则称它们相对应。也就是说，这两个运行具有相同的故障模式，且所有代理的初始状态都相同。回顾一下，运行的初始全局状态、信息交换协议和行动协议共同决定了完整运行。相对于上下文 γ =(E, F π)，若对所有相对应的运行 $r \in \mathrm{R}_{\mathcal{E},\mathcal{F},P}$ 和 $r' \in \mathrm{R}_{\mathcal{E},\mathcal{F},P'}$、所有在 r 中无故障的代理 i 以及所有时间 m，只要 $P_i(r_i(m))$ = 对 $v \in \{0, 1\}$ 作出决定 $_i(v)$，就有 $P'_i(r'_i(m'))$ $\neq$ 对任意 $m' < m$ 和 $w \in \{0, 1\}$ 作出决定 $_i(w)$，则称行动协议 P 支配行动协议 P'，记作 $P' \leq_\gamma P$。也就是说，P 作出决定的时间不晚于 P'。若 $P' \leq_\gamma P$ 且 $P \leq_\gamma P'$ 不成立，则称 P 相对于 γ 严格支配 P'。若对于 EBA 上下文 γ，不存在任何针对 γ 的 EBA 决策协议能够严格支配 P，则称 EBA 决策协议 P 相对于该上下文是最优的。

6 OPTIMAL EBA WITH RESPECT TO LIMITED INFORMATION EXCHANGE

In this section, we describe a knowledge-based program for EBA that is somewhat biased towards 0, show that it is correct, and show that it is optimal with respect to all information-exchange protocols that satisfy a certain safety condition. As discussed in the introduction, there is no protocol for EBA in the presence of omission failures where an agent decides 0 as soon as it hears that some agent had an initial preference of 0. So we consider instead a program where an agent decides 0 if it hears that some agent had an initial preference of 0 via a chain of agents; this is essentially the condition used to decide 0 in the crash-failure case.

代理序列 $i_0, \ldots, i_m$ 由互不相同的代理组成，若满足以下条件，则称其为解释系统 I 的运行 r 中长度为 m 的 *0*-链：(a) I, (r,0) $\models$ 初始$_{i_0} = 0$；(b) 对所有满足 $0 \leq m' \leq m$ 的 m'，代理 $i_{m'}$ 首次在 r 的第 $m' + 1$ 轮作出 0 的决定；(c) 对所有满足 $1 \leq m' \leq m$ 的 m'，$i_{m'}$ 在点 (r, m') 知道 $i_{m'-1}$ 刚刚作出了 0 的决定。若在运行 r 中存在一条长度为 m、以代理 i 结尾的 0-链，则称代理 i在运行 m 的第 m 轮收到一条 0-链 m。

文献中存在我们所称的 0-链的变体 [1, 8]。 [1], 中针对崩溃故障定义的 0-链只要求 $i_{m'}$ 在第 m' 轮从 $i_{m'-1}$ 收到一条消息，以便 $m' \geq 1$。 [8] 中的 0- 链定义要求 i_m 在第 m' 轮从 i_{m-1} 收到一条消息，以便 $m' \geq 1$，并且 i_m 认为在收到消息时 i_{m-1} 可能是无故障的（对于崩溃故障，这一点会自动成立，因为 [1], 中只考虑崩溃故障；但对于遗漏故障则不一定成立，因为 [8] 中考虑了遗漏故障）；此外， i_m 必须是无故障的。

令 $\mathbf{P}_i^0$ 为代理 i的如下基于知识的程序:

程序：$\mathbf{P}_i^0$

if 已决定$_i \neq \perp$**then** 空操作**else if** 初始偏好$_i = 0 \ \vee \ K_i($
$\bigvee_{j\in$代理 已决定$_j = 0)$ **then** 作出决定$_i(0)$**else if** $K_i(\bigwedge_{j\in$代理
$\neg($正在决定$_j = 0))$ **then** 作出决定$_i(1)$**else** 空操作

换言之，只要 i 尚未作出决定，那么如果 i 的初始偏好为 0，或者知道有人刚刚决定了 0， i 就作出决定 0；如果 i 知道没有代理当前正在决定 0，就作出决定 1；否则不采取任何操作。$\mathbf{P}^0$ 本质上与 Castañeda 等人 [1] 在崩溃故障情形下使用的基于知识的程序相同。我们将证明，作出决定 0 的第二个条件——即 i 知道有人刚刚决定了 0——当且仅当 i 收到一条 0-链时成立。因此，如果 i 尚未作出决定且没有正在决定 0，那么 i 当且仅当 i 知道没有代理正在接收 0-链时，才作出决定 1。后一条件的思想与 Castañeda 等人关于不存在隐藏路径的概念十分接近。

$\mathbf{P}^0$ satisfies all the EBA properties in all EBA contexts.

Proposition 6.1. *If $\gamma = (\mathcal{E}, \mathcal{F}\ \pi)$ is an EBA context, then all implementations of the knowledge-based program $\mathbf{P}^0$ with respect to γ are EBA decision protocols for γ. Indeed, all implementations of $\mathbf{P}^0$ terminate after at most $t+1$ rounds of message exchange and Validity holds even for faulty agents.*

We next prove that some implementations of $\mathbf{P}^0$ are actually optimal EBA decision protocols in certain EBA contexts. Instead of individually proving optimality of $\mathbf{P}^0$ with respect to specific information exchanges, we first give a sufficient condition for optimality in EBA contexts. We then look at two specific contexts that satisfy this safety property and show that implementations of $\mathbf{P}^0$ in them are optimal.

Definition 6.2 (safety). A knowledge-based program $\mathbf{P}$ is *safe with respect to an EBA context $\gamma = (\mathcal{E}, \mathcal{F}\ \pi)$* if, for all implementations P of $\mathbf{P}$ with respect to γ and all points (r, m) of $\mathcal{I} = (\mathcal{R}_{\mathcal{E},\mathcal{F},P}, \pi)$, the following two conditions hold:

(1) 如果 i 在 (r, m) 前未收到 0-链，则存在一个点 $(r'm)$，使得 $r_i(m) = r'_i(m)$，且所有代理在 r' 中的初始偏好为 1。(2) 如果 $\mathrm{I}, (r, m) \models \neg K_i(\bigwedge_{j\in\mathrm{Agt}} \neg($决定$_j = 0))$，且 i 在 r 中的第 $m+1$ 轮之前未作出决定，则存在一个点 $(r'm)$，使得：(a) $r_i(m) = r'_i(m)$，(b) i 在 r' 中无故障，(c) 某个在 r' 中无故障的代理 j 在 r 的第 $m+1$ 轮决定 0；此外，如果 $m \geq 1$，则存在一个运行 r'' 和一个代理 j'，使得 j 和 j' 在 r'' 中无故障， $r'_j(m) = r''_j(m)$，且 j' 在 r'' 的第 m 轮决定 0。

直观地说，(1) 表明，代理获知某个代理的初始偏好为 0 的唯一途径是通过一条 0-链；因此，(1) 意味着，如果 i 在 (r,m) 之前尚未收到一条 0-链，那么 $I,(r,m) \models \neg K_i \exists 0$。子句 (2) 表明，代理无法决定 1（即，$\mathbf{P}^0$ 中用于决定 1 的检验条件不成立）的唯一原因，是它认为某个无故障代理可能正在决定 0；因此，(2) 意味着，如果 $\mathrm{I},(r,m) \models \neg K_i(\bigwedge_{j \in Agt} \neg(\text{正在决定}_j = 0))$，并且 i 到第 m 轮时尚未作出决定，那么 $\mathrm{I},(r,m) \models \neg K_i(\bigwedge_{j \in \mathcal{N}} \neg(\text{正在决定}_j = 0))$。

Note that a knowledge-based program will in general not be safe with respect to an FIP, since an agent i may learn that some agent j had an initial preference of 0 without receiving a 0-chain. As a result, the first condition will not hold, since at all points that i considers possible, j has an initial preference of 0. (In Section 7, we show that a small modification of $\mathbf{P}^0$ is optimal even with full information exchange.) But, as we shall see, $\mathbf{P}^0$ is safe with respect to two EBA contexts of interest, where agents do not keep track of who is faulty.

Theorem 6.3. *If γ is an EBA context and $\mathbf{P}^0$ is safe with respect to γ, then all implementations of $\mathbf{P}^0$ are optimal with respect to γ.*

We now describe two families of concrete EBA contexts with respect to which $\mathbf{P}^0$ is safe, parameterized by the number n of agents involved. For the first, let $\mathcal{E}_{\min}(n)$ be the *minimal* information-exchange protocol for n agents, where for each agent i, the following hold:

- The local states have the form $\langle time_i, init_i, decided_i, jd_i \rangle$. Thus, the local states include just what is required in an EBA context
- The initial local states have the form $\langle 0, init_i, \bot, \bot \rangle$.
- $M_i = \{0,1\}$, $M^0 = \{0\}$, $M^1 = \{1\}$ and $M^2 = \{\bot\}$.
- For each agent j, if $a = \mathtt{decide}_i(v)$ then $\mu_{ij}(s,a) = v$; otherwise, $\mu_{ij}(s,a) = \bot$. Note μ_{ij} satisfies the constraint we imposed for EBA contexts. Intuitively, if $\mu_{ij}(s,a) = v \neq \bot$, then i is about to decide v.
- The state-update component δ_i is defined on $time_i$, $init_i$, $decided_i$, and jd_i just as in EBA contexts.

For the second, let $\mathcal{E}_{basic}(n)$ be the *basic* information-exchange protocol for n agents, where the local states of agents are like those in a minimal information-exchange protocol, except that, in addition to the other messages allowed in a basic information-exchange protocol, each agent i can send a message of the form $(init, 1)$, and their local states have one additional component, $\#1_i$, that intuitively counts how many messages of the form $(init, 1)$ i received in the last round. In more detail, for each agent i, the following hold:

- The local states have the form $\langle time_i, init_i, decided_i, jd_i, \#1_i \rangle$, where $\#1_i \in \{0, \ldots, n\}$.
- The initial local states have the form $\langle 0, init_i, \bot, \bot, 0 \rangle$.
- $M_i = \{0, 1, (init, 1)\}$, $M^0 = \{0\}$, $M^1 = \{1\}$, and $M^2 = \{(init, 1)\ \bot\}$.
- For all agents j, if $a = \mathtt{decide}_i(v)$ then $\mu_{ij}(s,a) = v$ for $v \in \{0,1\}$; if $a = \mathtt{noop}$ and s has the form $\langle m, 1, \bot, \bot, k \rangle$, then $\mu_{ij}(s,a) = (init, 1)$; otherwise, $\mu_{ij}(s,a) = \bot$.
- The state-update component δ_i is defined as in EBA contexts, with the added constraint that $\#1_i$ is updated to the number of messages of the form $(init, 1)$ that i receives in the current round if $decided_i = \bot$ and i does not receive a message $v \in \{0,1\}$ from some agent j; otherwise, $\#1_i$ is set to 0 (essentially, for technical reasons, once a decision is made, $\#1_i$ is ignored).

设 $\gamma_{\min,n,t} = (\mathrm{E}_{\min}(n)\ SO(t)\ \pi_{\min,n})$ 表示最小上下文族，其中有 n 个代理，至多有 $t < n$ 个故障代理，语言除时间 $\mathtt{init}_i = k$、$\mathtt{init}_i = v$、$\mathtt{decided}_i(v)$ 和 $i \in \mathsf{N}$（其中 $i \in \{1, \ldots, n\}$）之外，还包括 $\mathtt{jd}_i = v$，并且 $\pi_{\min,n}$ 以显而易见的方式解释这些原子命题。类似地，令 $\gamma_{basic,n,t} = (\mathrm{E}_{basic}(n)\ SO(t)\ \pi_{basic,n})$ 表示基本上下文族。

有 n 个代理和 $t < n$ 个故障代理，其中语言除了最小上下文中使用的所有原子命题外，还包括 $\#1_i = k$；其中，同样，$\pi_{basic,n}$ 以显而易见的方式解释所有原子命题。

Proposition 6.4. *$\mathbf{P}^0$ is safe with respect to all contexts $\gamma_{\min,n,t}$ and $\gamma_{basic,n,t}$ such that $n - t \geq 2$.*

Finally, we provide EBA decision protocols that implement $\mathbf{P}^0$ in the two contexts of interest. Let P^{min} be the protocol implemented by the following (standard) program:

Program: P_i^{min}

if $decided_i \neq \bot$ **then** noop
else if $init_i = 0 \vee jd_i = 0$ **then** $\mathtt{decide}_i(0)$
else if $time_i = t + 1$ **then** $\mathtt{decide}_i(1)$
else noop

Intuitively, this EBA decision protocol decides 0 if the agent has initial value 0 or hears of a 0-decision by another agent. If the agent does not hear about a 0-decision by time $t + 1$, then it decides 1.

Theorem 6.5. *If $t \leq n - 2$, then P^{min} implements $\mathbf{P}^0$ in the EBA context $\gamma_{\min,n,t}$.*

Finally, let P^{basic} be the EBA decision protocol that implements the following program:

Program: P_i^{basic}

if $decided_i \neq \bot$ **then** noop
else if $init_i = 0 \vee jd_i = 0$ **then** $\mathtt{decide}_i(0)$
else if $\#1_i > n - time_i \vee jd_i = 1$ **then** $\mathtt{decide}_i(1)$
else noop

Theorem 6.6. *If $t \leq n - 2$, then P^{basic} implements $\mathbf{P}^0$ in the EBA context $\gamma_{basic,n,t}$.*

We get the following immediate corollary to Theorems 6.3, 6.5, and 6.6.

Corollary 6.7. *P^{basic} is optimal with respect to $\gamma_{basic,n,t}$ and P^{min} is optimal with respect to $\gamma_{\min,n,t}$.*

7 AN OPTIMAL FULL-INFORMATION PROTOCOL FOR EBA

While $\mathbf{P}^0$ is optimal with respect to the basic and minimal information-exchange settings, it is not optimal in the full-information setting, as the following example shows.

例 *7.1.* 假设 $t = 10$ 和 $n = 20$。考虑这样一次运行：所有代理的初始值均为 1，代理 1–10 故障，并且没有故障代理在任何轮次发送消息。这意味着，在第一轮结束时，每个非故障代理都知道哪些代理发生了故障。在第二轮结束时，哪些代理发生了故障成为非故障代理之间的共同知识：此时每个非故障代理 i 都知道，其他每个非故障代理 j 都知道哪些代理是非故障代理，而且（根据同样的推理）i 知道 j 知道所有非故障代理都知道哪些代理发生了故障，依此类推。此外，非故障代理之间的共同知识是，没有非故障代理已经作出决定；而某个代理的初始偏好为 0，并非非故障代理之间的共同知识，而某个代理的初始偏好为 1 则是共同

知识。如下文所示，这意味着非故障代理可以在第 3 轮对 1 作出决定。另一方面，在 P^{basic} 和 P^{min}的情况下，非故障代理在这次运行中要到第 12 轮才会作出决定。

直观地说，当非故障代理共同知道哪些代理是非故障代理时，我们认为 $C_{\mathcal{N}}(t\text{-个故障})$ 成立。事实证明，只有当非故障代理共同知道 t 个代理发生故障时，这种情况才会出现。因此，对于每个公式 ϕ，我们将 $C_{\mathcal{N}}(t\text{-个故障} \wedge \phi)$ 定义为 $\exists A \subseteq \text{代理集}\ (|A| = t \wedge C_{\mathcal{N}}(\bigwedge_{i \in A}(i \notin \mathrm{N}) \wedge \phi))$ 的缩写。我们还将未-作出决定$_{\mathcal{N}}(x)$ 定义为 $\bigwedge_{j \in \mathcal{N}} \neg(\text{已作出决定}_j = x)$ 对于 $x \in \{0, 1\}$ 的缩写。现在我们可以将示例中的情形形式化。

命题 7.2. 如果 P 是上下文 $\gamma_{fip,n,t}$ 和 $I_{P,\gamma_{fip},n,t}$ 中的一个最优协议，且 $(r, m) \models \text{已决定}_i = \bot \wedge K_i(C_{\mathcal{N}}(t \text{ 个故障} \wedge \text{未作出决定}_{\mathcal{N}}(1) \wedge \exists 0))$，那么 $N(r)$ 中所有尚未作出决定的代理都会在第 $m+1$ 轮作出决定；对于 $I_{P,\gamma}$，$(r, m) \models \text{已决定}_i = \bot \wedge K_i(C_{\mathcal{N}}(t \text{ 个故障} \wedge \text{未作出决定}_{\mathcal{N}}(0) \wedge \exists 1))$ 的情形，结论也类似。

事实证明，如果我们给$\mathbf{P}^0$ 添加一个条件，用来测试这种共同知识，并在该条件成立时作出相应决定，那么即使在完全信息交换的情况下，我们也能得到一个最优程序。具体而言，令$\mathbf{P}^1_i$ 为代理 i的以下基于知识的程序：

Program: $\mathbf{P}^1_i$

如果已决定$_i \neq \bot$，**则**空操作
else if $K_i(C_{\mathcal{N}}(t\text{-}faulty \wedge no\text{-}decided_{\mathcal{N}}(1) \wedge \exists 0))$ **then** $\mathtt{decide}_i(0)$
else if $K_i(C_{\mathcal{N}}(t\text{-}faulty \wedge no\text{-}decided_{\mathcal{N}}(0) \wedge \exists 1))$ **then** $\mathtt{decide}_i(1)$
else if $init_i = 0 \ \vee \ K_i(\bigvee_{j \in Agt} jdecided_j = 0)$ **then** $\mathtt{decide}_i(0)$
else if $K_i(\bigwedge_{j \in Agt} \neg(deciding_j = 0))$ **then** $\mathtt{decide}_i(1)$
else noop

请注意，在基本上下文和最小上下文中，代理永远无法得知哪些代理存在故障，因此，无故障代理之间永远不会形成关于哪些代理存在故障的共同知识。因此，在上下文$\gamma_{\min,n,t}$和 $\gamma_{basic,n,t}$中，$\mathbf{P}^1$ 等价于 $\mathbf{P}^0$，，因此$\mathbf{P}^1$ 在这些上下文中是正确且最优的。正如我们即将证明的那样，$\mathbf{P}^1$ 在完全信息交换的情况下同样是正确且最优的。

为证明正确性和最优性，我们采用 Halpern、Moses 和 Waarts [8]的方法，并考虑一种略有不同的全信息上下文。我们假设每个代理 i的局部状态不包含变量已决定$_i$ 和jd_i，但包含一个或多个变量，用于记录从所有代理接收到的全部消息。如果代理在其局部状态中记录所有收到的消息，那么对于给定的决策协议 P，变量已决定$_i$ 和jd_i 就是冗余的；它们的值可以根据收到的消息推断出来。令$\gamma_{fip、n、t}$ 表示上述全信息上下文的族。在局部状态中不包含已决定$_i$ 和jd_i 的优点是，对于所有决策协议 P和 P'，P和 $P_{\prime}$在 $\gamma_{fip、n、t}$中的相应运行实际上是相同的；尽管代理可能作出不同的决定，但它们在任何时刻的局部状态都相同。（如果局部状态包含有关决定的信息，尤其是包含变量已决定$_i$ 和jd_i，情况就不是这样了。）这里至关重要的是，我们讨论的是 FIP；对于任意信息交换协议，该结论并不成立。

Proposition 7.3. *All implementations of* $\mathbf{P}^1$ *with respect to* $\gamma_{fip,n,t}$ *are EBA decision protocols for* $\gamma_{fip,n,t}$.

为了使用相对于 $\gamma_{fip,n,t}$ 的最优性刻画，我们需要回顾 [8] 中的一些内容。给定一个由代理构成的指示集 S，点 $(r_{\prime}m')$ 是从 (r, m) -⊡可达的（相对于 S）。

如果存在运行 r^0、. . .、 r^k，时间 m_0、 m'_0、 . . .、 m_k、 m'_k，以及代理 i_0、 . . .、 i_{k-1}，使得 $(r^0, m_0) = (r, m)$，$(r^k, m'_k) = (r' m')$，并且对于 $0 \leq j \leq k-1$，有 $i_j \in \mathrm{S}(r^j, m'_j) \cap \mathrm{S}(r^{j+1}, m_{j+1})$，且 $r_j^j(m'_j) = r_j^{j+1}(m_{j+1})$。[2] 使用 [8], 的记号，令 $\mathrm{N} \wedge \mathrm{O}$ 表示指示集，其中 $(\mathrm{N} \wedge \mathrm{O})(r, m)$ 由在点 (r, m) 无故障且即将决定 1 或已经决定 1 的所有代理组成。令 $(\mathrm{N} \wedge \mathrm{Z})$ 为对应的 0 集合。

定义 7.4（弱安全性）。如果对于 EBA 上下文 γ，知识型协议 P 的所有实现 P、$\mathrm{I} = (\mathrm{R}_{\varepsilon,\mathcal{F},P}, \pi)$ 中的所有点 (r, m) 以及所有代理i，只要 $\mathrm{I}_{\gamma_{fip},n,t,P}, (r, m) \models i \in \mathrm{N}$ ⓐ(已决定$_i = \perp$) 成立，那么就存在点 $(r^{0\prime} m)$、$(r^{0\prime\prime}, m)$、$(r^{1\prime}, m)$ 和 $(r^{1\prime\prime}, m)$，使得：

1. $r_i(m) = r_i^{0\prime}(m) = r_i^{1\prime}(m)$, 2. i在 $r^{0\prime}$和 $r^{1\prime}$ 中无故障，3. $(r^{0\prime}, m'')$ 从 $(r^{0\prime}, m)$ 是 $(\mathrm{N} \wedge \mathrm{Z})$-⊡-可达的，4. $(r^{1\prime}, m'')$ 从 $(r^{1\prime}, m)$ 是 $(\mathrm{N} \wedge \mathrm{O})$-⊡-可达的，5. $r^{0\prime\prime}$ 中所有代理的初始偏好均为 0，6. $r^{1\prime\prime}$ 中所有代理的初始偏好均为 1。

我们对弱安全性的兴趣源于 Halpern、Moses 和 Waarts [8]证明的以下结果。该结果的表述使用了两个算子：$B_i^{\mathcal{N}}$ $C^{\boxdot}$ S。 $B_i^{\mathcal{N}}$是$K_i(i \in \mathrm{N} \Rightarrow \phi)$的缩写。因此，当且仅当对于所有满足 $r_i(m) = r'_i(m')$ 且 $i \in \mathrm{N}\,(r')$ 的点 $(r' m')$，都有 $\mathrm{I}, (r' m') \models \phi$，才有 $\mathrm{I}, (r, m) \models B_i^{\mathcal{N}}\phi$。直观而言，如果 i知道若其无故障则 ϕ成立，那么 $B_i^{\mathcal{N}}$就成立。 $C^{\boxdot}$ S. 算子可以用 S-⊡-可达性来刻画。在 [8], 中，证明了当且仅当对于所有可从 (r, m) 经 S-⊡-可达的点 $(r' m')$，都有 $\mathrm{I}, (r' m') \models \phi$，才有 $\mathrm{I}, (r, m) \models C^{\boxdot}\mathrm{S}^{\phi}$。我们感兴趣的是 S 为 $\mathrm{N} \wedge \mathrm{O}$ 或 $\mathrm{N} \wedge \mathrm{Z}$ 的情形。

定理 7.5。 *[8,* 定理 *5.4]* 一个 *EBA* 协议 P 关于$\gamma_{fip、n、t}$ 是最优的，当且仅当以下两个条件成立：

$$\mathcal{I}_{\gamma_{fip,n,t},P} \models i \in \mathcal{N} \Rightarrow (\bigcirc(decided_i = 0) \Leftrightarrow B_i^{\mathcal{N}}(\exists 0 \wedge C^{\boxdot}_{\mathcal{N}\wedge O}\exists 0 \wedge \neg(\bigcirc(decided_i = 1))))$$
$$\mathcal{I}_{\gamma_{fip,n,t},P} \models i \in \mathcal{N} \Rightarrow (\bigcirc(decided_i = 1) \Leftrightarrow B_i^{\mathcal{N}}(\exists 1 \wedge C^{\boxdot}_{\mathcal{N}\wedge Z}\exists 1 \wedge \neg(\bigcirc(decided_i = 0)))).$$

利用定理 7.5，我们可以证明，对于$\mathbf{P}^1$而言，弱安全性蕴含最优性。

定理 7.6。 如果$\mathbf{P}^1$ 关于$\gamma_{fip、n、t}$ 具有弱安全性，那么$\mathbf{P}^1$ 的所有实现关于 $\gamma_{fip、n、t}$都是最优的。

为了说明基于知识的程序$\mathbf{P}^1$ 关于完整信息上下文满足弱安全性，我们给出一个构造性证明，明确构造出能够见证弱安全性定义中各项条件的点序列。

证明的主要思路源于以下观察。如果非故障代理 i 无法作出决定，那么$\mathbf{P}^1$ 中的共同知识条件就不成立。粗略地说，接下来可以证明存在一些点（$r^{0\prime}, m$）和（$r^{1\prime}, m$），如弱安全性的定义所述，使得某个故障代理 k 在整个运行过程中都表现得像非故障代理。[3] 此外，由于这个故障代理 k 并未表现出

[2]Halpern、Moses 和 Waarts [8] 引入了一族持续共同知识 算子 $C^{\boxdot}_S$，使得 $C^{\boxdot}_S\phi$ 在点（r、 m）成立，当且仅当 ϕ 在所有从（r、 m）可通过 S-⊡到达的点（r'、 m'）上都为真。通过在持续共同知识的定义中取 $m_k = m'_k$，我们得到标准（指示性）共同知识；由于我们使用的是同步系统，因此可以限制为取 $m'_j = m_{j+1}$。[3]请注意，由于运行中的故障代理集合由故障模式决定，因此，一个代理 i在某次运行中是故障代理、但在整个运行过程中始终表现为无故障，这是相容的。由于对我们的结果真正重要的只是没有代理能够检测到代理 i是故障代理，因此我们可以假设 i的故障行为仅涉及不向其自身发送消息，以此得到我们的结果。

故障行为，所有由另一个代理而非 k 发生故障的运行，都与该点无法区分。事实证明，存在一个表现得像非故障代理的故障代理，是一个强有力的条件；它使我们能够构造一系列可由 $\boxdot$到达的点，其中终点具有经过修改的消息模式，同时满足关于 k的相同条件。只需暂时让 k 在序列的中间点表现出故障行为，就能做到这一点。因此，我们将这种 k 的存在用作不变式，以便沿着 $\boxdot$-路径逐步前进。

定理 7.7。 $\mathbf{P}^1$ 关于 $\gamma_{fip,n,t}$是弱安全的。

命题 7.3、定理 7.6 和定理 7.7 一起推出以下推论。

推论 7.8。 所有关于 $\gamma_{fip,n,t}$ 的$\mathbf{P}^1$ 实现，关于 $\gamma_{fip,n,t}$都是最优的。

借助 [12]提出的全信息交换紧凑通信图表示法，可以在多项式时间内实现$\mathbf{P}^1$ 。直观地说，共同知识条件的实现利用了这样一个观察：如果代理的故障状态是非故障代理之间的共同知识，那么在前一时刻，它必定是分布式知识 （如果某个事实 ϕ 是代理指示集 S 中的分布式知识，就意味着这些代理汇集各自的知识后便会知道 ϕ；例如，如果非故障代理彼此之间知道哪些代理发生了故障，那么故障代理集合就是非故障代理之间的分布式知识）。由于非故障代理每轮都会发送描述其完整状态的消息，我们可以通过考察非故障代理在第 m 轮收到消息的那些代理在点 $(r, m-1)$ 的局部状态，来检查点 (r, m) 上是否满足 C_N（t-故障）。

命题 7.9。 存在一个关于全信息交换的 $\mathbf{P}^1$ 的多项式时间实现 P^{opt} 。

8 讨论

我们引入了有限信息交换的概念，考察了针对各种信息交换协议的 EBA 最优性，并描述了一种可高效实现的最优 FIP。显然，还有许多工作有待完成。目前我们正在探索两个近期方向。首先，我们希望研究有限信息交换对其他值得关注的协议的影响。其次，我们正在探索应用认知综合技术，在有限信息交换模型的上下文中，根据基于知识的程序自动推导协议。这似乎大大拓展了这些技术的适用范围（参见[9, 10]）。

最后，我们将更仔细地考察有限信息交换对 EBA 的成本与收益，以此结束本节讨论。我们聚焦于第 6 节讨论的两种情形，以及全信息上下文，并考察在最可能的情况下（即没有故障时），达到决定所需的成本，包括发送的比特数和所需的轮次数。虽然这些结果并不复杂，但有助于凸显其中的权衡。令$\gamma_{fip,n,t}$ 为一个存在遗漏故障的全信息 EBA 上下文，并令 P^{fip} 为$\mathbf{P}^1$ 在$\gamma_{fip,n,t}$中的一个实现。

我们首先从比特数的角度考察消息复杂度。在最简信息交换协议 P^{min}中，每条消息都可以用一个比特表示，而且代理只有在首次作出决定时才发送消息，否则保持沉默。由于每个代理在每次运行中恰好发送一条消息，并将其发送给所有其他代理， n^2 比特会被总共发送。在基本信息交换协议 P^{basic}中，我们仍然只需用常数个比特表示消息；只要代理尚未作出决定，就会向所有其他代理发送消息，这意味着最多发送 $t+1$ 轮。由此得到以下结果：

命题 8.1. 在 P^{min}的每次运行中，总共发送 n^2 比特；在 P^{basic}的每次运行中，总共最多发送 $O(n^2t)$ 比特。

相比之下，标准通信图实现的全信息交换使用 $O(n^4t^2)$ 比特 [12]。

接下来我们考虑决策时间。我们重点关注无故障情形，因为在大多数应用中，最常见的运行是没有故障的运行。

Proposition 8.2. *If r is a failure-free run, then*
(a) 如果 r 中至少有一个代理的初始偏好为 0，那么所有代理都会在第 2 轮之前作出决定，并满足 P^{min}、P^{basic} 和 P^{fip}。*(b)* 如果所有代理的初始偏好均为 1，那么所有代理都会在第 $t+2$ 轮之前满足 P^{min}，并在 2 轮之前满足 P^{basic} 和 P^{fip}。

因此，对于无故障运行，基本上下文和全信息上下文中的代理会同时作出决定。唯一一个基本上下文比最小上下文更早作出决定的无故障运行，是每个代理都以 1 作为初始偏好的运行。因此，实现全信息上下文并在比特数上产生二次开销，永远不会为无故障运行带来改进。假设每种初始偏好配置出现的可能性相同，那么对于无故障运行，使用基本上下文而非最小上下文，仅在 $1/2^n$ 的时间内有所改进。

If failure-free runs are sufficiently common, this suggests that the gain of using an FIP may not be worth the cost; even the tradeoff between P^{basic} and P^{min} is not so clear. We conjecture that even in runs with failures, P^{basic} may not be much worse than P^{fip}. This emphasizes the advantages of considering limited information exchange, and further motivates considering optimal protocols with limited information exchange more broadly.

A PROOFS

A.1 Proofs for Section 6

Proposition 6.1. *If $\gamma = (\mathcal{E}, \mathcal{F}\ \pi)$ is an EBA context, then all implementations of the knowledge-based program $\mathbf{P}^0$ with respect to γ are EBA decision protocols for γ. Indeed, all implementations of $\mathbf{P}^0$ terminate after at most $t+1$ rounds of message exchange and Validity holds even for faulty agents.*

Proof. Fix an implementation P of $\mathbf{P}^0$ in $\gamma_{fip,n,t}$.

Unique-Decision follows from the fact that P_i makes at most one decision per round and the fact that whether a decision was made is recorded in the local state variable $decided_i$.

To see that Agreement holds, we first show by induction on m that if i has not decided before round $m+1$ and $\mathcal{I}, (r, m) \models init_i = 0 \vee K_i(\bigvee_{j \in Agt} jdecided_j = 0)$, then i receives a 0-chain in round m. If $m = 0$, then it must be the case that $\mathcal{I}, (r, m) \models init_i = 0$ which is a 0-chain with $i_1 = i$. If $m > 0$, we cannot have $\mathcal{I}, (r, m) \models init_i = 0$ (as i would have decided earlier) so we must have $\mathcal{I}, (r, m) \models K_i(jdecided_j = 0)$ for some $j \in Agt$. Then, $\mathcal{I}, (r, m) \models jdecided_j = 0$ and the result follows from the induction hypothesis.

反设 r 是一个运行，其中存在无故障代理 i 和 j 以及时间 m，使得 $\mathrm{I}, (r, m) \models$已决定$_i = 0 \wedge$ 已决定 $_j = 1$。假设 j 在第$m_j + 1$轮作出决定 1，而 i 在第 $m_i + 1$轮作出决定 0，因此决策条件分别首次在时间 m_i 和 m_j 成立。如果 $m_j \leq m_i$，则必定有 $\mathrm{I}, (r, m_i) \models \mathrm{init}_i = 0$ 或者 $\mathrm{I}, (r, m_i) \models K_i$(j已决定$_k = 0$)，其中某个 $k \in \mathrm{Agt}$。根据我们的观察，可以得出结论：i 在时间m_i 收到一个 0-链，这意味着存在一个代理 i'，使得 $\mathrm{I}, (r, m_j) \models$正在作出决定$_{i'} = 0$。因此，$\mathrm{I}, (r, m_j) \models \neg K_j(\neg$(正在作出决定$_{i'} = 0$))，所以 j 不能在 m_j 作出决定 1。如果 $m_j > m_i$，由于 i 作出决定 0，截断。

在第 $m_i + 1$ 轮，我们必须有 $\mathrm{I}, (r, m_i) \models init_i = 0 \bigvee_{j \in Agt} K_i(jdecided_j = 0)$。我们可以再次利用这一观察得出结论：$i$ 在 r 中的时间 m_i 收到了一条 0-链。由于 i 是无故障的，j 必须在第 $m_i + 1$ 轮听到来自 i 的消息，因此 $\mathrm{I}, (r, m_j) \models K_j(jdecided_i = 0)$。由此可知，代理 j 在此次运行中应当决定 0，这与假设 j 决定 1 相矛盾。

对于有效性，注意到，如果 i 决定0，根据前面的观察，必定存在一个0-链，因此必定有一个初始偏好为0的代理。如果代理 i 决定1，那么 i 在第一轮中并未决定0，因此我们必须有 $init_i = 1$。请注意，即使 i 出现故障，这一论证仍然成立。

最后，我们通过证明在所有运行中，所有无故障代理都必须在 $t + 2$ 轮前作出决定来证明终止性；最多经过 $t + 1$ 轮消息交换。为此，我们首先证明：如果某个代理在运行 r 中决定 0，那么所有决定 0 的代理都必须在 $t + 2$ 轮前作出决定；最多经过 $t + 1$ 轮消息交换。反设某个代理 i 在第 $m > t + 2$ 轮决定 0。那么必定存在一条 0-链 $i_1, \ldots, i_m$，满足 $i_m = i$。0-链上的代理互不相同。由于一次运行中至多有 t 个故障代理，因此 $i_1, \ldots, i_{t+1}$ 中必有一个是无故障代理，不妨设为 i_j。但这意味着所有代理（包括 i）都会在第 $j \leq t + 1$ 轮收到来自 i_j 的消息，从而推断出 i_j 即将决定 0（根据我们对 EBA 上下文中 μ 的假设）；如果它们尚未作出决定，就都会在第 $j + 1 < m$ 轮决定 0。这就得到了所需的矛盾。因此，如果 i 到第 $t + 2$ 轮时尚未决定 0，那么 $\mathrm{I}, (r, t + 2) \models K_i(\bigwedge_{j \in 代理} \neg(作出决定_j = 0))$，所以 i 将在第 $t + 2$ 轮决定 1（如果它尚未作出决定）。□

定理 6.3。 如果 γ 是一个 *EBA* 上下文，并且 $\mathbf{P}^0$ 相对于 γ 是安全的，那么 $\mathbf{P}^0$ 的所有实现相对于 γ 都是最优的。

证明。反设 P 是相对于 EBA 上下文 γ 和 EBA 协议 P' 的 P^0 的一个实现，并且相对于 γ 严格优于 P。如果 P' 严格优于 P，根据定义，存在 $\mathrm{I}_{\gamma, P}$ 的相应运行 r，以及 $\mathrm{I}_{\gamma, P'}$ 的相应运行 r'，一个非故障代理 i 和一个轮次 k，使得 P' 在 r' 的第 k 轮作出决定，而 P 在 r 中直到第 $k + 1$ 轮之前都没有作出决定。显然，还存在相应的运行 r 和 r'，一个代理 i（可能存在故障）以及一个轮次 k，使得 P' 在 r' 的第 k 轮作出决定，而 P 要么在 r 中直到第 $k + 1$ 轮之前都没有作出决定，要么在 r 的第 k 轮作出了不同的决定。令 k 为满足上述条件的最早轮次，并令 r 和 r' 分别为 P 和 P' 的相应运行，其中在 r' 中，P' 在第 k 轮作出决定，而在 r 中，P 要么截至第 k 轮仍未作出决定，要么在第 k 轮作出了不同于 P' 的决定。（注意，我们在此使用的 k 可能小于满足以下条件的最小 k：P' 在第 k 轮作出决定，而 P 在相应运行中直到第 $k + 1$ 轮之前都没有作出决定。）由于两个系统使用相同的信息交换协议，因此，在两个系统的相应运行中，所有代理在时间 $k - 1$ 之前的状态必定完全相同：它们要么尚未作出决定，要么已经作出决定且决定相同，因此，它们将在 P 和 P' 的相应运行中发送相同的消息，并经历相同的状态转换，直至时间 $k - 1$。现在我们考虑两种情况。

假设 i 在 r' 的第 k 轮作出决定 0。由于 i 要么在 r 的第 k 轮或之前没有作出决定，要么在该轮作出决定 1，因此 i 在 $(r, k - 1)$ 之前没有收到 0-链。由于 P^0 相对于 γ 是安全的，根据安全条件的第一部分可知，存在 P 的一次运行 r_1，使得 $r_i(k - 1) = (r_1)_i(k - 1)$，且在 r_1 中，所有代理的初始偏好均为 1。令 $r'1$ 为与 r_1 对应的 P' 的运行。如上所述，i 在 $(r_1, k - 1)$ 和 $(r'1, k - 1)$ 中必定处于相同状态。由于 i 在 $(r, k - 1)$ 和 $(r'k - 1)$ 中也处于相同状态，因此 i 在 $(r'k - 1)$ 和 $(r'1, k - 1)$ 中处于相同状态，所以必定在 $r'1$ 的第 k 轮作出决定 0。现在，注意到 P^0 中作出决定 0 的决策规则要求至少有一个代理的初始偏好为 0，由此即可得到所需的矛盾。请注意，这是因为如命题 6.1 所示，有效性对故障代理也成立。

假设 i 在 r' 的第 k 轮作出决定 1，在 r 的第 k 轮没有作出决定 1，并且在 r 中没有早于第 k 轮作出决定。那么必有 $\mathrm{I}_{\gamma,P,(r,k-1)} \models \neg K_i(\bigwedge_{j\in\mathrm{Agt}} \neg(\text{正在作出决定}_j = 0))$: 如果 i 在第 k 轮作出决定 0，那么显然 i 认为某个代理正在作出决定 0 是可能的；如果 i 没有作出决定 0，而是作出决定 1（唯一的另一种可能），那么该公式也必定成立。由于 P^0 相对于 γ 是安全的，因此存在 P 的一次运行 r_1，使得 $r_i(k-1) = (r_1)_i(k-1)$，i 在 r_1 中是无故障的，并且存在一个无故障代理 j，在 r_1 的第 k 轮作出决定 0。同样，令 r_1' 为与 r_1 对应的 P' 的运行。如上所述，i 在 $(r'k-1)$ 和 $(r'1,k-1)$ 中处于相同状态，因此必定在 $r'1$ 的第 k 轮作出决定 1。由于 P' 支配 P，j 在 $r'1$ 的第 k 轮也必须作出决定，并且必须作出决定 1（因为 i 正在作出决定 1）。

若 $k = 1$，由于 j在 r_1中决定了 0，j在 r_1和 $r'1$ 中的初始值都必须为 0。令 r_1''为 P'的一个运行，其中所有代理的初始值均为 0，且均无故障。由于 $(r''1)_j(0) = (r'1)_j(0)$，j必须在运行 $r''1$ 中与 P'一起决定 1，这就得到了所需的矛盾。若 $k > 1$，根据安全条件的条件 2(c) 的后半部分，存在 P 的一个运行 r_2和一个代理 j'，满足 $(r_2)_j(k-1) = (r_1)_j(k-1)$，$j$和 j'在 r_2中均无故障，且 j'在 r_2 的第 $k-1$ 轮决定 0。令 $r'2$ 为与 r_2对应的 P'的运行。如上所述，$(r_2)_j(k-1) = (r_2)_j'(k-1)$。因此，$(r_1)_j'(k-1) = (r_2)_j'(k-1)$，所以 j必须在 $r'2$ 的第 k 轮决定 1。由于 j'在 r_2的第 $k-1$ 轮无故障且作出决定，并且 P'支配 P，代理 j'也必须在$r'2$ 的第 $k-1$ 轮或之前作出决定。由于按构造，第 k 轮是 P和 P'在对应运行中作出不同决定的最早轮次，因此 j'必须在 $r'2$ 的第 $k-1$ 轮决定 0。但这意味着 j和 j'在 r_2'中作出了不同的决定，尽管二者均无故障。这就得到了所需的矛盾。□

命题 6.4。 P^0 对于所有满足 $n - t \geq 2$ 的上下文 $\gamma_{\min,n,t}$和$\gamma_{basic,n,t}$ 都是安全的。

证明。我们同时对 $\gamma_{\min,n,t}$ and 和 $\gamma_{basic,n,t}$ 进行论证。设 P 是 P^0 在 $\gamma_{\min,n,t}$（分别为 $\gamma_{basic,n,t}$）中的一个实现，并设 r 是 $\mathrm{I}_{\gamma_{\min},n,t,P}$（分别为 $\mathrm{I}_{\gamma_{basic},n,t,P}$）中的一次运行。我们首先证明安全条件的第一部分成立。假设 i 在 (r,m) 之前尚未收到 0-链。我们要证明存在一个点 $(r'm)$，使得 $r_i(m) = r'_i(m)$，且在 r' 中所有代理的初始偏好均为 1。设 r' 是一次运行，其中所有代理的初始偏好均为 1，且对手与 r 中的相同。利用归纳法对 k 进行归纳，并根据 r 和 r' 中的故障模式相同这一事实，容易证明：对于所有代理 j 和时间 k，如果 j 在 (r,k) 之前尚未收到 0-链，那么 $r_j(k) = r'j(k)$。由此立即可得 $r_i(m) = r'i(m)$，从而完成论证的第一部分。

为证明安全条件的第二部分成立，假设 $\mathrm{I},(r,m) \models \neg(K_i(\bigwedge_{j\in\mathrm{Agt}}\neg(\text{作出决定}_j = 0))$，且 i在 r的第 $m+1$ 轮之前尚未作出决定。显然，若 $m = 0$，安全条件的第二部分便成立，因此我们假设 $m \geq 1$。这意味着 $\mathrm{init}_i = 1$，否则代理 i早已在第 1 轮作出决定。必定存在一个点 (r^+,m)，使得 $r_i(m) = r_i^+(m)$，并且某个代理 j在 r^+的第 $m+1$轮将 0 作出决定。由于 $n - t \geq 2$，必定存在代理，例如 i'和 j'，它们在 r^+中无故障；若 i在 r^+中无故障，我们可以令 $i = i'$，若 j在 r^+中无故障，则可以令 $j = j'$。若 $m > 1$，那么 i'和 j'在 r^+中的初始偏好为 1；否则，i会在 r^+中于第 2 轮或之前将 0 作出决定，因而在 r中也会如此，这与假设 i在 r的第 $m+1$轮之前尚未作出决定相矛盾。 j的初始偏好也必定为 1，否则，鉴于 $m \geq 1$，j会在第 $1 \neq m+1$轮作出决定。

注意，在 r^+中，(a) 没有代理在第 $m+1$ 轮或之前作出决定 1（因为如果代理 j'' 在 r^+ 的第 k 轮作出决定 1，那么 $\mathrm{I},(r^+,k-1) \models K_{j''}(\bigvee_{j'''\in\mathrm{Agt}} \neg(\text{决定}_{j'''} = 0))$，这与 j 在 r 的第 $m+1$ 轮作出决定 0 的事实相矛盾，因此必定存在一条以 j 结束、长度为 m 的 0-链，且该链位于 r 中），(b) 在 r^+ 中，没有非故障代理在第 m 轮之前作出决定 0（否则 i 和 j 将在 r^+ 的第 m 轮或之前作出决定 0），(c) i 和 j 不会在 r^+ 的第 m 轮或之前作出决定 0，并且 (d) i、j 以及所有非故障代理在 $\gamma_{\min}$、n、t 中均不发送消息（即发送 $\perp$），并在 γ_{basic}、n、t 中发送 (init, 1)，直至

并包括第 $m-1$ 轮；i 和 j 也在 $\gamma_{\min}$ 的第 m 轮向 r 发送 $\perp$（分别为 (初始值,1)）（n）（分别为 ($\gamma_{basic,n,t}$）。请注意，i' 或 j' 可能会在 r^+ 的第 m 轮作出决定，选择 0，此时它们将发送 0；否则，和 i、j 一样，它们会在 $\gamma_{\min,n,t}$ 中的第 m 轮向 r 发送 $\perp$（分别为 (初始值, 1)）（分别为 ($\gamma_{basic,n,t}$）。

我们想要修改 r^+，得到一个运行 r'，使得 (a) $r_i(m) = r'_i(m)$，(b) i 和 j 在 r' 中无故障，并且 (c) j 在 r' 的第 $m+1$ 轮作出决定 0。令 $\alpha = (\mathrm{N}', F')$ 为 r' 中的对手。我们假设所有代理在 r' 中的初始偏好与在 r^+ 中的相同，并据此定义 r''；r'' 中的对手 (N'' F'') 定义如下：$\mathrm{N}'' = \mathrm{N}' - \{i', j'\} \cup \{i, j\}$（因此 i 和 j 在 r'' 中无故障）。粗略地说，F'' 在 r^+ 和 r'' 中互换了 i 与 i' 的故障，以及 j 与 j' 的故障。更确切地说，对于所有代理 j''，(a) 如果按照 F'，j'' 在 r^+ 的第 k 轮没有收到来自 i（分别为 j）的消息，并且在特殊情形 $k = m$ 中，i' 和 j' 都没有发送消息 0，那么按照 F''，j'' 在 r' 的第 k 轮不会收到来自 i'（分别为 j'）的消息；(b) 如果按照 F'，j'' 在 r^+ 的某一轮收到来自 i（分别为 j）的消息，或者如果 $k = m$ 且 i' 或 j' 在 r 的第 m 轮发送了消息 0，那么按照 F''，j'' 会在 r' 的第 k 轮收到来自 i'（分别为 j'）的消息。

我们声称，对于所有 $k \le m$，所有代理 j'' 在运行 r^+ 和 r' 的时间 k 都具有相同的状态。我们通过对 k 进行归纳法来证明这一点。当 $k = 0$ 时，这显然成立，因为所有代理在 r^+ 和 r' 中具有相同的初始偏好。若 $0 < k < m$，则这是因为所有代理在时间 $k-1$ 具有相同的状态，而两次运行的唯一区别在于：在 r^+ 的第 k 轮中，j'' 没有收到来自 i（分别地，j）的消息，那么在 r' 的第 k 轮中，j'' 也不会收到来自 i'（分别地，j'）的消息。这些消息要么是 $\perp$（如果上下文为 $\gamma_{\min}$、n、t），要么是 (init, 1)（如果上下文为 γ_{basic}、n、t）。但显然，这一区别不会影响 j'' 的状态；特别是，如果上下文为 γ_{basic}、n、t，那么 j'' 在两种情况下收到的 (init, 1) 消息数量相同，而它的状态只记录这一信息。如果 $k = m$，且 i' 和 j' 均未在 r^+ 的第 m 轮发送消息 0，同样的论证也适用。若 i' 或 j' 确实在 r^+ 的第 m 轮发送消息 0，由于它们在 r^+ 中是无故障的，所有代理都会收到该消息。根据构造，它们也会在 r' 的第 m 轮发送这条消息，并且所有代理都会在 r' 收到它。随后，转移函数保证所有代理在 r^+ 和 r' 的第 m 轮具有相同的状态。特别地，由此可得 $r_i^+(m) = r'_i(m)$，且 j 在 r' 的第 $m+1$ 轮作出决定 0，如所需。

最后，如果 $m \ge 1$，我们必须构造一个运行 r''，以满足条件 2(c) 的第二部分。由于如前所示，j 在 r' 的第 $m+1$ 轮作出决定 0，因此 j 必须在 r 中的时间 m 收到一个 0-链。令 j' 为该链上的最后一个代理。因此，j' 在 r' 的第 m 轮作出决定 0。如果 j' 在 r' 中无故障，我们就完成了。如果 j' 是故障代理，则分两种情况讨论。如果 $m = 1$，那么必有 $\mathrm{init}_{j'} = 0$。考虑一个运行 r''，其中所有代理的初始值都与 r' 中相同；若 $(N' F')$ 是 r' 中的对手，则 r'' 中的对手为 ($N' - \{j'\}$ F'')，其中 F'' 在 $N' - \{j'\}$ 中的所有代理上都与 F' 一致。显然 $r'_j(1) = r''_j(1)$，论证至此完成。如果 $m > 1$，则必定存在某个不同于 i 的非故障代理 j''，因为 $n - t \ge 2$。与上面的论证一样，j' 和 j'' 的初始值必须为 1（否则 j' 会在第 1 轮作出决定，而 i 会在 r' 中的第 2 轮作出决定）。接下来，我们仿照前面的论证构造 r''：所有代理在 r' 和 r'' 中的初始值都相同，并令 r'' 中的对手为 ($N' \cup \{j''\} - \{j'\}$ F'')，其中 F'' 交换了 j' 和 j'' 的角色。细节留给读者。□

THEOREM 6.5. *If $t \le n-2$ then P^{min} implements $\mathbf{P}^0$ in the EBA context $\gamma_{\min,n,t}$.*

PROOF. Let $\mathcal{I}$ be the system $\mathcal{I}_{\gamma_{\min,n,t},\mathbf{P}^0}$. We show that for all runs r and times m, we have $P_i^{min}(r_i(m)) = (\mathbf{P}_i^0)^{\mathcal{I}}(r_i(m))$.

- 如果 $P_i^{min}(r_i(m))$ =空操作，因为已决定$_i \ne \perp$ 在 $r_i(m)$ 中，那么我们显然也有 $(\mathbf{P}_i^0)_{\mathcal{I}}(r_i(m))$ =空操作，因为 I，$(r, m) \models K_i$(已决定$_i \ne \perp$)。

- 如果 $P_i^{min}(r_i(m)) = $决定$_i(0)$，我们必须已在 $r_i(m)$ 中作出决定$_i = \perp$，且要么是 $\text{init}_i = 0$，要么是 $\text{jd}_i = 0$。如果 $r_i(m)$ 中有 $\text{init}_i = 0$，那么 $\text{I}, (r, m) \models K_i(\text{init}_i = 0)$，因此 $(\text{P}_i^0)\text{I}\,(r_i(m)) = $决定$_i(0)$。如果有 $\text{jd}_i = 0$，且 $r_i(m)$ 中有 $\text{init}_i \neq 0$，那么我们必须有 $r_i(m)$ 中的时间$_i > 0$，并且对于 $m' < m$，$\text{jd}_i = \perp$ 位于$r_i(m')$中，否则，代理 i就会更早决定 0，而我们也会在 $r_i(m)$ 中作出决定$_i \neq \perp$。此外，i必定从某个代理 j 收到了消息 0。因此，$\text{I}, (r, m) \models K_i(\bigvee_{j\in}\text{Agt jdecided}_j = 0)$。由此可得 $(\text{P}_i^0)\text{I}\,(r_i(m)) =$ 决定 $_i(0)$。
- 如果 $P_i^{min}(r_i(m)) = $作出决定$_i(1)$，那么必有 已决定$_i = \perp$、初始$_i = 1$、$jd_i \neq 0$，且 时间$_i = t + 1$属于 $r_i(m)$。正如命题 6.1 的证明中关于终止性的论证所示，我们有 $\text{I}, (r, m) \models K_i(\bigwedge_{j\in}$代理 $\neg(j$已决定 $_j = 0))$。因此，$(\text{P}_i^0)_I\,(r_i(m)) = $作出决定$_i(1)$。
- 最后，如果根据 P_i^{min} 的最后一行，$P_i^{min}(r_i(m)) = $为空操作，那么必有时间$_i < t + 1$、初始$_i = 1$、已决定$_i = \perp$和 $\text{jd}_i = \perp$属于 $r_i(m)$。（注意，$t' = m$。）考虑运行 r'，其中所有代理的初始偏好均为 1，且均无故障。显然，i在 r 和 r' 中截至时间 m 收到的消息相同，因此 $r_i(m) = r'i(m)$。于是，必有 $\text{I}, (r, m) \models \neg K_i(\text{初始}_i = 0 \vee \bigvee_{j\in}$代理 已决定$_j = 0)$。因此，$(\text{P}_i^0)\text{I}\,(r_i(m)) \neq$作出决定$_i(0)$。此外，也不难构造一个运行 r''，使 i认为存在这样一种可能性：恰有 t' 个故障代理，并且某个非故障代理 j在 r'' 的第 t' 轮中收到一条长度为 t' 的 0-链。于是，$\text{I}, (r, m) \models \neg K_i(\bigwedge_{j\in}$代理 $\neg($作出决定$_j = 0))$，因此 $(\text{P}_i^0)\text{I}\,(r_i(m)) \neq$作出决定$_i(1)$。所以，$(\text{P}_i^0)\text{I}\,(r_i(m)) = \perp$。

由此可知，P^{min} 在 $\gamma_{\min}\mathbf{P}^0{}_{n,t,}$ 中实现。 □

定理 6.6。 如果 $t \leq n - 2$，那么 P^{basic} 在 *EBA* 上下文 $\gamma_{basic}\mathbf{P}^0{}_{n,t,}$ 中实现。

证明。我们采用与前面的论证相同的方法。如果 $P_i^{basic}(r_i(m)) = \perp$或 $P_i^{basic}(r_i(m)) =$对 $_i(0)$ 作出决定，那么论证与 P^{min} 的完全相同。如果 $P_i^{basic}(r_i(m)) =$对 $_i(1)$ 作出决定，那么我们通过归纳法证明：当我们已有 已决定$_i = \perp$、$init_i = 1$，且有 $jd_i = 1$ 或 $\#1_i > n - m$ 位于 $r_i(m)$ 中时，也有 $\text{I}, (r, m) \models K_i(\bigwedge_{j\in}$代理 $\neg($作出决定$_j = 0))$。

因此，假设 $\text{decided}_i = \perp$ 和 $\text{init}_i = 1$ 位于 $r_i(m)$ 中。如果 $\#1_i > n - m$ 位于 $r_i(m)$ 中，那么很容易看出，不可能存在长度为 m、位于 r 中的 0-链（因为唯一可能参与此 0-链的代理，是那些没有发送 (init, 1) 消息的代理）。因此，$\text{I}, (r, m) \models K_i(\bigwedge_{j\in}$代理 $\neg($对$_j = 0$作出决定$))$。另一方面，如果 $\text{jd}_i = 1$ 位于 $r_i(m)$ 中，那么对于$m' < m$，我们必须有 $\text{jd}_i = \perp$ 位于 $r_i(m')$ 中，否则代理 i就会更早决定 1，而我们也会在$r_i(m)$ 中决定$_i \neq \perp$。此外，i 必定在 r 的第 m 轮收到了某个代理 j 发来的消息 1。因此，j 在 r 的第 m 轮决定 1，所以我们必须已有 $\text{decided}_j = \perp$、$\text{init}_j = 1$，并且有 $\text{jd}_j = 1$ 或 $\#1_j > n - m$ 位于 $r_j(m-1)$ 中。根据归纳假设，$\text{I}, (r, m-1) \models K_j(\bigwedge_{j'\in}$代理 $\neg($对$_{j'} = 0$作出决定$))$。如果某个代理 j' 在 r 的第 $m + 1$ 轮决定 0，那么必定存在一条以 j' 结尾、终值为 0 的 0-链，因此 j' 必须从某个在第 m 轮决定 0 的代理处收到消息，这与 $\text{I}, (r, m - 1) \models K_j(\bigwedge_{j'\in}$代理 $\neg($对$_{j'} = 0$作出决定$))$ 矛盾。因此，根据 $r(m)$ 中的信息可知，在 r 的第 $m + 1$ 轮没有代理决定 0，所以 $\text{I}, (r, m) \models K_i(\bigwedge_{j\in}$代理 $\neg($对$_j = 0$作出决定$))$。无论哪种情况，我们都有 $(\text{P}_i^0)^I(r_i(m))$ 对 $_i(1)$ 作出决定。

最后，如果在 P_i^{basic} 的最后一行中 $P_i^{basic}(r_i(m)) =$ 为空操作，那么类似于上文所用论证的推理表明，代理 i 认为所有代理的初始偏好都是 1 是可能的，因此它不知道存在一条 0-链，但也无法排除存在 0-链的可能性。我们必须有 $\text{decided}_i = \text{jd}_i = \perp$、$\text{init}_i = 1$和 $\#1_i \leq n - m$ 位于 $r_i(m)$ 中。由于 $\text{init}_i = 1$ 和 $\text{jd}_i = \perp$ 位于 $r_i(m)$ 中，在点 (r, m)，i认为运行 r' 是可能的；在该运行中，每个代理的初始偏好都是 1，且消息模式与 r 相同。于是我们有 $\text{I}, (r, m) \models \neg K_i(\bigwedge_{j\in}$ Agt (jdecided $_j = 0))$，因此 $(\text{P}_i^0)^I(r_i(m)) \neq$对 (0) 作出决定。同样，由于 $\#1_i \leq n - m$, i 认为存在一条 0-链是可能的，该链由那些没有从 i 收到 1 的代理组成

听到 1 的代理组成。因此，$\mathrm{I}, (r, m) \models \neg K_i(\bigwedge_{j \in Agt} \neg(\text{正在决定}_j = 0))$，所以 $(\mathbf{P}_i^0)^{\mathcal{I}}(r_i(m))$ $\neq$ 对 (1) 作出决定。由此可得 $(\mathbf{P}_i^0)^{\mathcal{I}}(r_i(m)) = \perp$。因此，$P^{basic}$ 在 $\gamma_{basic,\ n,\ t}$ 中实现了 $\mathbf{P}^0$。□

A.2 第 7 节的证明

A.2.1 KBP 的动机。

命题 *7.2.* 如果 P 是在上下文 $\gamma_{fip,n,t}$ 中的最优协议，并且 $I_{P,\gamma_{fip,n,t}}, (r, m) \models$ 已决定$_i = \perp \wedge K_i(C_{\mathcal{N}}(t$ 个故障 $\wedge$ 未决定$_{\mathcal{N}}(1) \wedge \exists 0))$，那么 $N(r)$ 中所有尚未决定的代理都会在第 $m+1$ 轮作出决定；若 $I_{P,\gamma}, (r, m) \models$ 已决定 $_i = \perp \wedge K_i(C_{\mathcal{N}}(t$ 个故障 $\wedge$ 未决定 $_{\mathcal{N}}(0) \wedge \exists 1))$，结论也类似。

证明。为便于说明，令 $\gamma = \gamma_{fip,n,t}$。假设 $\mathrm{I}_{P,\gamma}, (r, m) \models$已决定$_i = \perp \wedge K_i(C_{\mathcal{N}}$(t 个故障 $\wedge$未决定$_{\mathcal{N}}$(1) $\wedge \exists 0))$。令 P' 为通过如下修改 P 而得到的上下文 γ 的协议。对于每个代理 i 和局部状态 $s \in L_i$，定义 $P'_i(s)$：如果 $\mathrm{I}_{P,\gamma}, (r, m) \models K_i(C_{\mathcal{N}}$(t 个故障 $\wedge$未决定$_{\mathcal{N}}$(1) $\wedge \exists 0))$ 对于满足 $r_i(m) = s$ 且 $P'_i(s) = P_i(s)$ 的点 (r, m) 成立，则作出决定$_i(0)$；否则为 $P' \leq P$。我们声称 P' 是上下文 γ 中的 EBA 协议。

令 r 和 r' 分别为 $\mathrm{I}_{P,\gamma}$ 和 $\mathrm{I}_{P',\gamma}$ 的对应运行。我们通过对 k 进行归纳法证明，如果 k 小于或等于最早的时间 m，使得 $\mathrm{I}_{P,\gamma}, (r, m) \models C_{\mathcal{N}}$(t 个故障 $\wedge$未决定$_{\mathcal{N}}$(1) $\wedge \exists 0)$，则 $r(k) = r'(k)$。基础情形显然成立，而归纳情形则由以下事实得出：如果 $r(k) = r'(k)$ 且 $\mathrm{I}_{P,\gamma}, (r, k) \not\models C_{\mathcal{N}}$(t 个故障 $\wedge$未决定$_{\mathcal{N}}$(1) $\wedge \exists 0))$，那么对于所有代理 i，$\mathrm{I}_{P,\gamma}, (r, k) \not\models K_i C_{\mathcal{N}}$(t 个故障 $\wedge$未决定$_{\mathcal{N}}$(1) $\wedge \exists 0)$（对于所有代理 i，$\mathrm{I}_{P,\gamma}, (r, k) \not\models K_i C_{\mathcal{N}}$(t 个故障 $\wedge$未决定 N (1) $\wedge \exists 0)$ 这一事实可由引理 A.3 得出，其证明如下），因此，对于所有代理 i，$P_i(r_i(k)) = P'_i(r'i(k))$。由于这些运行中的故障模式相同，因此可得 $r(k+1) = r'(k+1)$。此外，由于 $\mathrm{I} \models (C_{\mathcal{N}}(\phi) \wedge i \in \mathrm{N}) \Rightarrow K_i(C_{\mathcal{N}}(\phi))$（见 [5]），因此一旦 $\mathrm{I}_{P,\gamma}, (r, m) \models C_{\mathcal{N}}$(t 个故障 $\wedge$未决定$_{\mathcal{N}}$(1) $\wedge \exists 0)$，N (r') 中所有尚未作出决定的代理都将使用 P' 同时决定 0。

由这些事实可立即得出 $P' \leq_\gamma P$。如果非故障代理 i在 r，中作出决定的时间 m早于共同知识条件在 r 中变为真，那么 $r(m) = r$，(m')，且代理 i会在 r 中的时间 m作出相同的决定。一旦共同知识条件变为真，尚未作出决定的非故障代理 i就会使用 P'作出决定，因此至少会和使用 P时一样早作出决定。请注意，如果 i是故障代理，那么根据 $P' \leq_\gamma P$的定义，该代理 i可以先使用 P作出决定，再使用 P'作出决定，因此我们无需考虑这种情况。

接下来，我们证明 P' 是上下文 γ 中的 EBA 协议。唯一决策源于这样一个事实：上下文 γ会在局部状态中记录决定，并且对于记录了已作出决定的状态 s，$P_i(s) = \perp$。对于有效性，考虑 P' 的运行 r'，以及上下文 γ 中 P 对应的运行 r。如果共同知识条件尚未在某个点变为真，而非故障代理 i 在该点使用 P' 对值 v 作出决定，那么 $r(m) = r'(m)$，且 i 会使用 P 作出相同的决定。由 P 的有效性可知，某个代理的初始值为 v。或者，如果非故障代理 i在 $(r'm)$ 作出决定 0，是因为它在 $\mathrm{I}_{P,\gamma}$ 的 (r, m) 知道共同知识条件已经变为真，那么实际上 $\mathrm{I}_{P,\gamma}, (r, m) \models \exists 0$，而且同一事实在 $(r'm)$ 也成立。

对于一致性，考虑运行 r，，其中代理 $i \in \mathrm{N}(r$，$)$ 在第 $m_0 + 1$ 轮作出决定 0，而代理 $j \in \mathrm{N}(r')$ 在第 $m_1 + 1$ 轮作出决定 1。如果在对应的 P 运行 r中，截至时间 $\max(m_0, m_1)$ 共同知识条件尚未成立，那么由于 r 和 r，至少在这一时间之前完全相同，因此这与 P 的一致性相矛盾。如果共同知识条件在时间 $k \leq m_1$ 成立，那么根据 P'，截至时间 k 尚未作出决定的所有非故障代理都会在第 $k+1$ 轮决定 0。这意味着在时间 k 尚未作出决定的代理 j，在 r' 的第 k 轮决定 0，却在第 $m_1 + 1$ 轮决定了不同的值，这与我们已经证明的 P' 满足唯一决策相矛盾。另一方面，共同知识条件不可能在时间 m_1 之后成立，因为这意味着 未-已决定$_{\mathcal{N}}(1)$，而 $j \in \mathrm{N}(r$，$)$ 已在第 $m_1 + 1$ 轮决定 1。因此，P，满足 Agreeement(N)。

Final而且， P' 的终止性由 $P' \leq_\gamma P$ 以及 P 满足终止性的事实得出。

Since we have $P' \leq P$ and P' is an EBA protocol for context γ, it follows that $P \leq P'$, so in $\mathcal{I}_{P,\gamma}$, all undecided nonfaulty agents make a decision as soon as they know that the common knowledge condition holds. □

Note that the result does not tell us *what* decision the nonfaulty agents must make when the common knowledge condition holds. There may be situations where no nonfaulty agent has made a decision, and both common knowledge conditions hold. Either a 0 or a 1 decision would be acceptable in this case.

We now prove all the required properties of $\mathbf{P}^1$. We start by examining what each of the conditions is $\mathbf{P}^1$ tells us.

A.2.2 The common knowledge conditions. As a first step to understanding $\mathbf{P}^1$, we characterize the common knowledge conditions (i.e., the second and third lines). The next proposition gives necessary and sufficient conditions for these conditions to hold. These conditions show that the problem of computing when the common knowledge conditions hold is tractable.

Define $dist_{\mathcal{N}}(t\text{-}faulty)$ to be an abbreviation for

$$\exists A \subseteq Agt(|A| = t \wedge \forall i \in A\, \exists j \in \mathcal{N}\, (K_j(i \notin \mathcal{N}))).$$

Roughly speaking, $dist_{\mathcal{N}}(t\text{-}faulty)$ holds if, between them, the nonfaulty agents know about t faulty agents.

*Definition A.1 (*hears-from*).* We define a *one-step hears-from* relation in a run r on pairs (j, m) consisting of agents j and times m by saying that $(j'\ m')$ *one-step hears from* (j, m) in r if agent j sends j' a non-$\perp$ message in round $m+1$ of r that j'' receives and $m+1 \leq m'$. The *hears-from* relation is the transitive closure of the *one-step hears-from* relation. We write $(j, m) \rightarrow_r (j'\ m')$ if $(j'\ m')$ *hears-from* (j, m)

Those familiar with the *Lamport causality* relation [11] will recognize that the *hears-from* relation is similar in spirit.

Proposition A.2. *For all implementations P of the knowledge-based program $\mathbf{P}^1$ with respect to $\gamma_{fip,n,t}$,*

(a) $\mathcal{I}_{\gamma_{fip,n,t},P} \models time > 0 \Rightarrow (\ominus dist_{\mathcal{N}}(t\text{-}faulty) \Leftrightarrow C_{\mathcal{N}}(t\text{-}faulty))$.

(b) $\mathcal{I}_{\gamma_{fip,n,t},P} \models time > 0 \Rightarrow (no\text{-}decided_{\mathcal{N}}(v) \Rightarrow \wedge_{j\in\mathcal{N}} \ominus (K_j \bigcirc no\text{-}decided_j(v)))$ *and* $\mathcal{I}_{\gamma_{fip,n,t},P} \models (C_{\mathcal{N}}(t\text{-}faulty) \wedge (no\text{-}decided_{\mathcal{N}}(v))) \Leftrightarrow C_{\mathcal{N}}(t\text{-}faulty \wedge no\text{-}decided_{\mathcal{N}}(v))$, *for* $v \in \{0, 1\}$.

(c) $\mathcal{I}_{\gamma_{fip,n,t},P} \models time > 0 \Rightarrow ((C_{\mathcal{N}}(t\text{-}faulty) \wedge \ominus(\vee_{j\in\mathcal{N}} K_j(\exists v))) \Leftrightarrow C_{\mathcal{N}}(t\text{-}faulty \wedge \exists v))$, *for* $v \in \{0, 1\}$.

Proof. Let P be an implementation of $\mathbf{P}^1$ and let $\mathcal{I} = \mathcal{I}_{\gamma_{fip,n,t},P}$.

对于 (a) 部分，首先反设 I, $(r, m) \models \neg \ominus \mathrm{dist}_{\mathcal{N}}$(t 个故障) $\wedge C_{\mathcal{N}}$(t 个故障)。令 $A' = \{i\colon \exists j \in \mathrm{N}(r)(\mathrm{I}, (r, m-1) \models K_j(i \notin \mathrm{N}))\}$。根据假设，$|A'| < t$。必定存在某个集合 A，满足 $|A| = t$，使得 I, $(r, m) \models C_{\mathcal{N}}(\bigwedge_{i\in A}(i \notin \mathrm{N}))$。此外，$A'$是 A 的真子集，因为 r中的每个非故障代理都会在第 m 轮得知（如果他们此前尚未知晓）A' 中的每个代理都存在故障（通过收到一条来自某个非故障代理的消息，并据此推断出这一点）。令 r' 为一个运行，其中 $\mathrm{N}(r') = A'$，所有代理在 r 和 r' 中的初始状态相同，并且对于 A'中的所有代理，在 r 和 r' 中收到的消息相同。容易验证，对于 $\mathrm{N}(r)$ 中的所有代理，必有 $(r, m-1) \sim_j (r' m-1)$。（严格来说，我们通过对 k 进行归纳法证明：若 $j \in \mathrm{N}(r)$ 且 $k < m' \leq m-1$，则 $(j' m'-k) \rightarrow_r (j, m')$ 当且仅当 $(j' m'-k) \rightarrow_{r'} (j, m')$。也就是说，$\mathrm{N}(r)$ 中的所有代理都认为某个运行是可能的，即 r'；在该运行中，只有 A'中的代理是非故障代理。现在，尽管 i可能会在 r 的第 m 轮得知其他非故障代理的信息，但对于所有 $j \in \mathrm{N}(r)$

满足 $j \neq i$、i 必须认为在 (r, m) 时，j认为 (r, m) 是可能的，因为 j 在第 m 轮收到了除 A' 中的代理之外所有代理的消息。因此，在 (r, m) 时，非故障代理之间并不存在 t-故障 的共同知识。（注意，这里我们用到了 $n - t \geq 2$ 这一事实，因此存在一个代理 $j \in \mathrm{N}(r)$，使得 $j \neq i$。）

反过来，假设 $\mathrm{I}, (r, m-1) \models \mathrm{dist}_{\mathcal{N}}$(t 个故障)。因此，存在某个集合 A with $|A| = t$，使得 $\mathrm{I}, (r, m-1) \models \mathrm{dist}_{\mathcal{N}}$(t 个故障$_A$)，其中 dist (t 个故障$_A$) 是公式 $\forall i \in A \exists j \in \mathrm{N}(K_j(i \notin \mathrm{N}))$. 此外，对于所有运行 r'，如果 $\mathrm{I}, (r' m-1) \models \mathrm{dist}_{\mathcal{N}}$(t 个故障$_A$)，那么 $\mathrm{N}(r) = \mathrm{Agt} - A$。

众所周知（见 [5]），对于所有公式 ϕ和 ψ，如果 $\mathrm{I} \models \phi \Rightarrow E_{\mathcal{N}}(\psi \wedge \phi)$，那么 $\mathrm{I} \models \phi \Leftrightarrow \ominus_{\mathcal{N}}\psi$。因此，只需证明 $\mathrm{I} \models \ominus dist_{\mathcal{N}}$($t$-故障$_A$) $\Rightarrow E_{\mathcal{N}}$($t$-故障 $\wedge \ominus dist_{\mathcal{N}}$($t$-故障$_A$)).

对于所有点 (r, m')，若 $\mathrm{I}, (r, m') \models \ominus dist_{\mathcal{N}}$($t$-个故障$_A$)，则在点 (r, m)，所有非故障代理都知道 A 中的代理是故障的，而且这些就是仅有的故障代理。由于 $|A| = t$，我们有 $\mathrm{I}, (r, m') \models E_{\mathcal{N}}$($t$-个故障)。此外，由于所有非故障代理在 r，的第 m 轮都能听到其他所有非故障代理的消息，因此它们都知道 $\ominus dist_{\mathcal{N}}$($t$-个故障$_A$)。至此，(a) 部分的证明完成。

对于 (b) 部分，第一部分显然成立：如果某个非故障代理 i 到第 m 轮仍未对值 v 作出决定，那么在时间 $m-1$，i 知道这一情况将会发生。第二部分的充分性显然成立，因为既然 $\mathrm{N} \neq \emptyset$，那么对于所有公式 ϕ [5]，$\mathrm{I} \models C_{\mathcal{N}}\phi \Rightarrow \phi$。对于必要性，假设 $\mathrm{I}, (r, m) \models C_{\mathcal{N}}$(t-faulty) $\wedge$ no-decided$_{\mathcal{N}}(v)$。对于每个代理 $i \in \mathrm{N}(r)$，正如我们所观察到的，i 在 r 中的时间 $m-1$ 知道它不会在第 m 轮对 v 作出决定。由于我们使用的是全信息协议，所有非故障代理都会在 r 中的时间 m 知道 no-decided$_i(v)$ 成立。因此，$\mathrm{I}, (r, m) \models E_{\mathcal{N}}$(no-decided$_{\mathcal{N}}(v)$)。这是 $C_{\mathcal{N}}$ 的一个标准性质：$\mathrm{I} \models C_{\mathcal{N}}\phi \Rightarrow E_{\mathcal{N}}C_{\mathcal{N}}\phi$ [5]。因此，$\mathrm{I}, (r, m) \models E_{\mathcal{N}}$(no-decided$_{\mathcal{N}}(v) \wedge C_{\mathcal{N}}$t-faulty)。我们刚刚证明了 $\mathrm{I} \models C_{\mathcal{N}}$(t-faulty) $\wedge$no-decided$_{\mathcal{N}}(v)) \Rightarrow E_{\mathcal{N}}C_{\mathcal{N}}$(t-faulty) $\wedge$no-decided N (v)。由此可得，$\mathrm{I} \models (C_{\mathcal{N}}$ (t-faulty) $\wedge$(no-decided N $(v))) \Rightarrow C_{\mathcal{N}}$(no-decided$_{\mathcal{N}}(v)$)，如所需。

对于 (c) 部分，必要性方向的证明思路与 (b) 部分类似；具体细节留给读者。对于充分性方向，假设 $\mathrm{I}, (r, m) \models C_{\mathcal{N}}$(t 个故障) $\wedge \exists v$，并通过反证法，假设 $\mathrm{I}, (r, m) \not\models \vee_{j \in \mathrm{N}}(\ominus K_j(\exists v))$。设 r' 为一个运行，其中所有代理的初始状态均为 $1 - v$，$\mathrm{N}(r') = \mathrm{N}(r)$，且 r和 r'中的故障模式相同，唯一的区别在于，在 r' 的第 m 轮中，所有代理只能听到来自非故障代理的信息。我们声称，对于所有满足 $i \in \mathrm{N}(r)$ 的情况，都有 $(r, m-1) \sim_i (r' m-1)$。显然，如果 $j \in \mathrm{N}(r)$ 且 $k < m' \leq m-1$，那么 $(j' m' - k) \rightarrow_r (j, m')$ 当且仅当 $(j' m' - k) \rightarrow_{r'} (j, m')$。由于 (j, m') 在 r 中没有从任何初始值为 v 的代理处听到信息，因此在 r' 中也必然如此。现在，通过对 m'' 进行直接归纳可知，如果 $j \in \mathrm{N}(r)$、$m'' \leq m' < m-1$，且 $(j' m'') \rightarrow_r (j, m')$，那么 $(r, m'') \sim_{j'} (r' m'')$。与 (a) 部分类似，尽管 i 可能在 r 的第 m 轮获知 $\exists v$，但对于所有满足 $j \neq i$、i 的 $j \in \mathrm{N}(r)$，i 必须认为，在 (r, m) 时，j 认为 $(r' m)$ 是可能的，因此不会获知 $\exists v$。因此，在 (r, m) 时，$\exists v$ 并非非故障代理之间的共同知识。

□

Lemma A.3. *For all implementations P of the knowledge-based program $\mathbf{P}^1$ with respect to $\gamma_{fip,n,t}$, if $\mathcal{I}_{\gamma_{fip,n,t},P}, (r, m) \models C_{\mathcal{N}}(t\text{-}faulty \wedge no\text{-}decided_{\mathcal{N}}(1-v) \wedge \exists v)$, then for all $i \in Agt$, $\mathcal{I}_{\gamma_{fip,n,t},P}, (r, m) \models K_i(C_{\mathcal{N}}(t\text{-}faulty \wedge no\text{-}decided_{\mathcal{N}}(1-v) \wedge \exists v))$. Moreover, for all agents i, $\mathcal{I}_{\gamma_{fip,n,t},P}, (r, m) \models C_{\mathcal{N}}(t\text{-}faulty) \Rightarrow K_i(C_{\mathcal{N}}(t\text{-}faulty)$.*

Proof. Let $\mathcal{I} = \mathcal{I}_{\gamma_{fip,n,t},P}$ and suppose that $\mathcal{I}, (r, m) \models C_{\mathcal{N}}(t\text{-}faulty \wedge no\text{-}decided_{\mathcal{N}}(1-v) \wedge \exists v)$ for some $v \in \{0, 1\}$ and $(r, m) \sim_i (r' m)$. We clearly must have $m > 0$, since we cannot have common knowledge of the faulty agents at time 0. By Proposition A.2, it follows that $\mathcal{I}, (r, m-1) \models dist_{\mathcal{N}}(t\text{-}faulty) \wedge (\wedge_{j \in \mathcal{N}}(K_j \bigcirc no\text{-}decided_j(v)) \wedge (\vee_{j \in \mathcal{N}} K_j(\exists v))$. Since the nonfaulty agents send messages to all agents in round m, all agents (not just the nonfaulty agents) will know at time m

that this was true at time $m-1$. It now follows from Proposition A.2 that, for all agents i, we have $\mathcal{I}_{\gamma_{fip,n,t},P}, (r,m) \models K_i(C_{\mathcal{N}}(t\text{-}faulty \wedge no\text{-}decided_{\mathcal{N}}(1-v) \wedge \exists v))$. □

接下来我们证明，一旦非故障代理知道了 t 故障代理的情况，每个代理都会在下一轮结束前作出决定。

引理 A.4。对于基于知识的程序$\mathbf{P}^1$关于$\gamma_{fip,n,t}, \mathcal{I}_{\gamma_{fip,n,t},P} \models C_{\mathcal{N}}(t$ 个故障$) \Rightarrow \bigwedge_{i\in}$代理 $\bigcirc\neg($已决定$_i = \perp)$的所有实现 P均满足。

证明。令 I $=\mathrm{I}_{\gamma_{fip,n,t},P}$，并假设 $i \in$代理、 $j \in \mathbf{N}$ 和 $\mathrm{I}, (r,m) \models C_{\mathcal{N}}$（$t$-个故障）。如果 j在第 $m' \leq m$轮作出决定 0，那么 i必须在第 $m'+1$轮前作出决定，因为它会在第m'轮从 j那里得知对方正在作出决定，因此 $\mathrm{I}, (r,m') \models K_i$（已决定$_j = 0$）。(请注意， i可能更早作出决定，或因满足某个共同知识条件而作出决定，但它肯定会在第 $m'+1$轮前作出决定。)如果 j 在某一轮 $m' \leq m$作出决定 1，那么以下情况之一成立

- I, $(r, m'-1) \models K_j(C_{\mathcal{N}}(t$-故障 $\wedge$不-作出决定$_{\mathcal{N}}(0) \wedge \exists 1)$，因此 I, $(r, m'-1) \models K_i(C_{\mathcal{N}}(t$-故障 $\wedge$不-作出决定 $_{\mathcal{N}}(0) \wedge \exists 1)$ 根据引理 A.3，且 i还必须在轮次 m'作出决定；
- $\mathrm{I}, (r, m'-1) \models K_j(\bigwedge_{k\in}$代理 $\neg($作出决定$_k = 0))$, 因此，由于 P是一个全信息协议，$\mathrm{I}, (r, m') \models \ominus K_j(\bigwedge_{k\in}$ 代理 $\neg($ 作出决定 $_k = 0)$ ）。由于 P 是一个 FIP， i 在第 m' 轮从 j 获悉，没有代理会在第 m'轮作出决定 0。因此，根据 P的第 4 行， i知道没有代理会在第 m' 轮作出决定 $0+1$。如果某个代理因第$m'+1$轮的共同知识条件而作出决定 0，那么根据上述论证， i 也会在第 $m'+1$轮作出决定 0。否则，适用 P的第 5 行，且 i会在第 $m'+1$轮作出决定 1。

We have just shown that that i decides by round $m+1$ if some nonfaulty agent j decides by round m. If no nonfaulty agent decides by round m, then $\mathcal{I}, (r,m) \models no\text{-}decided_{\mathcal{N}}(0) \wedge no\text{-}decided_{\mathcal{N}}(1)$. We must have $m > 0$ since $C_{\mathcal{N}}(t\text{-}faulty)$ cannot hold at time 0, and clearly $\mathcal{I}, (r, m-1) \models (\vee_{k\in\mathcal{N}}\exists 0) \vee (\vee_{k\in\mathcal{N}}\exists 1)$. It now follows easily from Proposition A.2 that $\mathcal{I}, (r,m) \models C_{\mathcal{N}}(t\text{-}faulty \wedge no\text{-}decided_{\mathcal{N}}(1) \wedge \exists 0)$ or $\mathcal{I}, (r,m) \models C_{\mathcal{N}}(t\text{-}faulty \wedge no\text{-}decided_{\mathcal{N}}(0) \wedge \exists 1)$; that is, one of the common knowledge conditions must hold. Hence, by Lemma A.3, agent i must decide by round $m+1$. □

A.2.3 刻画作出决定 *0* 的条件。

引理 *A.5.* 对于基于知识的程序 P^1 关于 $\gamma_{fip,n,t}$ 的所有实现 P，若 $I_{\gamma_{fip,n,t},P}, (r,m) \models \neg K_i(C_{\mathcal{N}}(t$ 个故障 $\wedge$未决定$_{\mathcal{N}}(1) \wedge \exists 0)) \wedge \neg K_i(C_{\mathcal{N}}(t$ 个故障 $\wedge$未决定 $N(0) \wedge \exists 1))$，则代理 i 在第 m 轮收到一条 *0*-链，当且仅当 $I_{\gamma_{fip,n,t},P}, (r,m) \models$初始$_i = 0 \vee \bigvee_{j\in}$代理集 $K_i(j$已决定$_j = 0)$，且 i在第 $m+1$ 轮之前尚未作出决定。

证明。我们对 m 进行归纳。设 I $=\mathrm{I}_{\gamma_{fip,n,t},P}$，并假设 $\mathrm{I}, (r,m) \models \neg K_i(C_{\mathcal{N}}($t 个故障 $\wedge$未决定 N (1) $\wedge \exists 0)) \wedge \neg K_i(C_{\mathcal{N}}($t 个故障 $\wedge$ 未决定 N (0) $\wedge \exists 1))$。由引理 A.3，可得 $\mathrm{I}, (r,m) \models \neg C_{\mathcal{N}}($t 个故障 $\wedge$ 未决定 $_{\mathcal{N}}(1) \wedge \exists 0)$ $\wedge \neg C_{\mathcal{N}}($t 个故障 $\wedge$未决定 N (0) $\wedge \exists 1)$。

If $m=0$, then the only-if direction follows from the fact that agent i receives a 0-chain in round 0 iff $\mathcal{I}, (r,0) \models init_i = 0$. Obviously, i has not decided at time 0. For the converse, we first observe that $\mathcal{I}, (r,0) \models \neg K_i(jdecided_j = 0)$ for all $j \in Agt$ as at time 0, $\mathcal{I}, (r,0) \models \neg(jdecided_j = 0)$ for all $j \in Agt$. Thus, $\mathcal{I}_{\gamma_{fip,n,t},P}, (r,0) \models init_i = 0 \vee \bigvee_{j\in Agt} K_i(jdecided_j = 0)$ iff $\mathcal{I}_{\gamma_{fip,n,t},P}, (r,0) \models init_i = 0$. So agent i decides 0 in round 1 and receives a 0-chain in round 0.

如果 $m > 0$，那么仅当方向可直接由 0-链的定义得出，因为最后一个代理 i 是在听到一位刚刚作出决定的代理的消息后，首次作出决定的。反过来，假设 $\mathrm{I}, (r,m) \models$init$_i = 0 \vee \bigvee_{j\in}$Agt K_i(jdecided $_j = 0)$ 在时间 m首次成立。由于该公式首次成立的时间是 $m > 0$，且 i在第 $m+1$轮之前尚未作出决定，因此我们必须有 $\mathrm{I}, (r,m) \models$init$_i \neq 0$。于是，对于某个 $j \in$Agt，有 $\mathrm{I}, (r,m) \models K_i$(jdecided $_j = 0)$，因此代理 j在第 m轮作出决定 0。我们必须有 $\mathrm{I}, (r, m-1) \models \neg C_{\mathcal{N}}$(t 个故障)，否则根据引理 A.4，

i 将在第 m轮之前作出决定。因此，由于共同知识条件， j 不可能决定 0。由$\mathbf{P}^1$可知，$\mathrm{I},(r,m-1) \models$初始 $_j = 0 \vee \bigvee_{j' \in$代理 $} K_j($已决定 $_{j'} = 0)$。根据归纳假设， j在第 $m-1$轮收到一条 0- 链。由于 $\mathrm{I},(r,m) \models K_i($ 已决定 $_j = 0)$，代理 i 在第 m 轮收到 j 发送的消息。由于不满足共同知识条件，且 i 尚未作出决定，代理 i 也在第 $m+1$轮决定 0。因此。 i 在第 $m+1$轮收到一条 0-链。□

A.2.4 刻画决定 *1* 的条件。为了刻画代理何时无法决定 1（不包括利用某个共同知识条件作出的决定），我们需要引入一些额外的定义。

定义 A.6。令 $\mathrm{len}_i(r,m)$ 表示 i在运行 r的时间 m所知道的最长 0-链的长度（其中，若 i不知道任何 0-链，则 $\mathrm{len}_i(r,m) = 0$）；令 $\mathrm{last}_{ij}(r,m)$ 表示满足 $(j,m') \rightarrow_r (i,m)$ 的最后一个时间m'（其中，若 (i,m) 根本没有收到来自 (j,m') 的消息，则 $\mathrm{last}_{ij}(r,m) = -1$）；并令 $\mathrm{latest}_{0_i}(r,m)$ 表示满足以下条件的最后一个时间 m'：对于某个代理 j，有 $(j,m') \rightarrow_r (i,m)$ 且 $\mathrm{I},(r,m') \models$正在决定$_j = 0$（若不存在这样的时间，则 $\mathrm{latest}_{0_i}(r,m) = -1$）。

直观来说，如果到时间 m共同知识条件仍不成立，那么代理 i无法作出决定 1，当且仅当有足够多的代理可能扩展代理 i所知的最长 0-链，使其长度至少达到 m。以下引理将这一直觉形式化。

命题 *A.7.* 对于关于$\gamma_{fip,n,t}$的基于知识的程序 P^1 的所有实现 P，若 $I_{\gamma_{fip,n,t},P},(r,m) \models \neg K_i(C_N($t 个故障 $\wedge$ 未决定$_N(0) \wedge \exists 1)), I_{\gamma_{fip,n,t},P},(r,m) \models \neg K_i(C_N($ t 个故障 $\wedge$未决定$_N(1) \wedge \exists 0))$, 且 $I\gamma_{fip,n,t},P,(r,m) \models$已决定 $_i = \perp$，则有以下结论：

- $I_{\gamma fip_{n,t},P},(r,m) \models \neg K_i \neg(\exists j \in Agt($作出决定$_j = 0))$ 当且仅当对于所有满足 $len_i(r,m) < m'' \leq m$ 的 m''，至少存在 $m'' - len_i(r,m)$ 个代理 j，使得 $last_{ij}(r,m) < m''$ 且 $I,(r,last_{ij}(r,m)+1) \models$ 已决定 $_j = \perp$。

证明。令 $\mathrm{I} = \mathrm{I}_{\gamma_{fip,n,t},P}$，并假设 $\mathrm{I},(r,m) \models \neg K_i(C_N($t 个故障 $\wedge$未决定$_N(1) \wedge \exists 0)) \wedge \neg K_i($ $C_N($t 个故障 $\wedge$未决定$_N(0) \wedge \exists 1))$。根据引理 A.3，我们有 $\mathrm{I},(r,m) \models \neg C_N($t 个故障 $\wedge$未决定 N (1) $\wedge \exists 0) \wedge \neg C_N($t 个故障 $\wedge$未决定$_N(0) \wedge \exists 1)$。注意，对于任何更早的时间，共同知识条件也不成立，否则 i 根据引理 A.4 早就已作出决定。

对于仅当方向，假设对于某个 $j \in Agt$，$\mathrm{I},(r,m) \models \neg K_i \neg($决定$_j = 0)$。必定存在一个运行 r'，使得 $(r,m) \sim_i (r',m)$，且 $\mathrm{I},(r',m) \models$决定$_j = 0$。

由于 $(r,m) \sim_i (r'm)$，并且 i 在 (r,m) 中不满足共同知识条件，因此在 $(r'm)$ 中也不满足这些条件。根据引理 A.5， j 必须在 r' 的第 m 轮收到一个 0-链。设 $i_0,\ldots,i_m$ 为这个 0-链。对于所有满足 $\mathrm{len}_i(r,m) < m^* \leq m$ 的 m^*，考虑 0-链中的代理 i_{m^*}。我们声称，对于所有 $m' \geq m^*$，都有 $(i_{m^*},m') \not\rightarrow_r (i,m)$。假设 $(i_{m^*},m') \rightarrow_r (i,m)$。那么，由于 $(r,m) \sim_i (r'm)$，我们也有 $(i_{m^*},m') \rightarrow_{r'} (i,m)$。由于 i_{m^*} 在 r' 的时间 m^* 收到一个 0-链，因此在 r 中也必然如此，并且 i 会得知这一情况，这与 $\mathrm{len}_i(r,m) < m^*$ 的假设矛盾。我们必须有 $\mathrm{last}^{ii_{m^*}}(r,m) < m^*$（否则 $\mathrm{len}_i(r,m)$ 至少为 m^*）。由此可知，$\mathrm{I},(r,\mathrm{last}^{ii_{m^*}}(r,m)+1) \models$ 已作出决定$_{i^*_m} = \perp$，否则 i_{m^*} 就不会出现在 0-链上。因此，对于所有满足 $\mathrm{len}_i(r,m) < m'' \leq m$ 的 m''，至少存在 $m'' - \mathrm{len}_i(r,m)$ 个代理 j，使得 $\mathrm{last}_{ij}(r,m) < m''$，即满足 $\mathrm{len}_i(r,m) < m^* \leq m^*$ 的代理 i_{m^*}。

Conversely, suppose that for all m'' where $len_i(r,m) < m'' \leq m$, there exist at least $m'' - len_i(r,m)$ agents j such that $last_{ij} < m''$ and $\mathcal{I},(r,last_{ij}(r,m)+1) \models decided_j = \perp$. Let $i_0, i_1, \ldots, i_{len_i(r,m)}$ be the longest 0-chain that i knows about at time m in run r.

根据假设，必定存在代理 $i_{len_i(r,m)+1}, \ldots\ldots, i_m$ 使得对于所有 m'' 满足 $\mathrm{len}_i(r,m) < m'' \leq m$，我们有 $\mathrm{last}_{ii_{m''}} < m''$ 和 $\mathrm{I},(r,\mathrm{last}^{ii_{m''}}(r,m)+1) \models$已决定$_{i_{m''}} = \perp$。考虑运行 r''

使得所有代理在 r 和 r'' 中的初始状态相同，$\mathrm{N}(r) = \mathrm{N}(r'')$，并且 r 和 r' 中的故障模式相同，唯一的区别在于代理 $i_{\mathrm{len}_i}(r, m)+1$、……、$i_m$ 接收和发送的消息。每个满足 $\mathrm{len}_i(r, m) < m'' \leq m$ 的代理 $i_{m''}$，在 r 和 r'' 中，直到包括第 last_i 轮 $i_{m''}$ 在内，都会从相同的代理接收消息；同样的代理也会从 $i_{m''}$ 接收消息，直到第 last_i 轮 $i_{m''}$。在第 last_i 轮 $i_{m''}$ 之后，代理 $i_{m''}$ 只会在第 m'' 轮收到来自代理 $i_{+m''-1}$ 的一条消息；并且如果 $m'' < m$，则在第 last_i 轮 $i_{m''}$ 之后，只会从 $i_{m''}$ 收到一条消息：代理 $i_{m''+1}$ 在第 $m''+1$ 轮收到来自 $i_{m''}$ 的消息。（如果 $m'' = m$，则在第 last_i 轮 $i_{m''}$ 之后，不会收到来自 $i_{m''}$ 的消息。）容易验证，$(r, m) \sim_i (r'' m)$，并且 i_0、……、i_m 在 r'' 中构成一个 0-链。□

推论 *A.8.* 对于基于知识的程序 P^1 关于 γ_{fip}、$_n$、$_t$ 的所有实现 P，如果 $I_{\gamma_{fip}}$、$_n$、$_t$、$_P$、$(r, m) \models \neg C_N$（t 个故障），$I_{\gamma_{fip}}$、$_n$、$_t$、$_P$、$(r, m) \models$ 已决定$_i = \perp$，$\mathit{len}_i(r, m) \leq m - 2$，并且代理 i 在 r 的第 m 轮中听到了除一个代理之外所有代理的消息，那么 $I_{\gamma_{fip}}$、$_n$、$_t$、$_P$、$(r, m) \models$ 作出决定$_i = 1$。

证明。假设 $\mathrm{I}_{\gamma_{fip},n,t,P}, (r, m) \models \neg C_N(\text{t 个故障}), \mathrm{len}_i(r, m) \leq m-2$，且代理 i 在 r 的第 m 轮中听到了除一个代理之外的所有代理的消息。对于代理 j 在第 m 轮中听到消息的所有代理，$\mathrm{last}_{ij}(r, m) \geq m$。因此，只有一个代理可能满足 $\mathrm{last}_{ij}(r, m) < m$。但是，由于 $m - \mathrm{len}_i(r, m) \geq 2$，命题 A.7 意味着 $\mathrm{I}_{\gamma_{fip},n,t,P}, (r, m) \models K_i \neg(\exists j \in \mathrm{Agt}(\text{作出决定}_j = 0))$。因此，代理 i 在第 $m+1$ 轮作出决定 1。□

A.2.5 $\mathbf{P}^1$ 满足 *EBA* 条件。

命题 7.3。关于 $\gamma_{fip,\ n,\ t}$ 的 $\mathbf{P}^1$ 的所有实现都是针对 $\gamma_{fip,\ n,\ t}$ 的 *EBA* 决策协议。

证明。固定 $\mathbf{P}^1$ 在 $\gamma_{fip,n,t}$ 中的一个实现 P。

唯一决策基本上可由与 $\mathbf{P}^0$ 情形相同的论证得出。唯一的区别在于，由于 P 是 FIP，因此可以从局部状态推断决定，而无需显式存储它们。

为证明一致性成立，反设存在一个运行 r，其中存在非故障代理 i 和 j，以及时间 m，使得 $\mathrm{I}, (r, m) \models$ 已决定$_i = 0 \wedge$ 已决定$_j = 1$。假设 j 在第 $m_j + 1$ 轮作出决定 1，而 i 在第 $m_i + 1$ 轮作出决定 0，因此决策条件分别首次在时间 m_i 和 m_j 成立。首先注意，如果 i 或 j 任一方使用某个共同知识条件作出决定，就会导致矛盾。如果 $\mathrm{I}, (r, m_i) \models K_i(C_N(\text{t 个故障} \wedge \text{no-decided}_N(1) \wedge \exists 0))$，那么 j 不可能在第 m_i 轮或之前作出决定 1，因为 $j \in \mathrm{N}$ 且 $\mathrm{I}, (r, m_i) \models \text{no-decided}_N(1)$。那么代理 j 会在第 $m_i + 1$ 轮作出决定 0，因为 $\mathrm{I}, (r, m_i) \models K_j(C_N(\text{t 个故障} \wedge \text{no-decided}(1) \wedge \exists 0))$，这与假设 j 在此运行中作出决定 1 矛盾。如果 $\mathrm{I}, (r, m_i) \models \neg K_i(C_N(\text{t 个故障} \wedge \text{no-decided}_N(1) \wedge \exists 0))$ 且 $\mathrm{I}, (r, m_j) \models K_j(C_N(\text{t 个故障} \wedge \text{no-decided}_N(0) \wedge \exists 1))$，那么 i 不可能在 m_j 之前作出决定 0，因为 $i \in \mathrm{N}$ 且 $\mathrm{I}, (r, m_j) \models \text{no-decided}_N(0)$。由于 $\mathrm{I}, (r, m_j) \models K_i(C_N(\text{t 个故障} \wedge \text{no-decided N}(0) \wedge \exists 1))$，并且另一个用于作出决定 0 的共同知识条件在时间 m_j 对 i 不成立，根据上述论证，i 会在第 $m_j + 1$ 轮作出决定 1，这与假设 i 在第 $m_i + 1$ 轮作出决定 0 矛盾。

It remains to consider the cases where neither i nor j decides using one of the common knowledge conditions.

If $m_j \leq m_i$, must have either $\mathcal{I}, (r, m_i) \models \mathit{init}_i = 0$ or $\mathcal{I}, (r, m_i) \models K_i(\mathit{jdecided}_k = 0)$ for some $k \in \mathit{Agt}$. Since, agent i does not decide using a common knowledge condition, we can apply Lemma A.5 to conclude that i receives a 0-chain at time m_i, which implies that there exists an agent i' such that $\mathcal{I}, (r, m_j) \models \mathit{deciding}_{i'} = 0$. Hence, $\mathcal{I}, (r, m_j) \models \neg K_j(\neg(\mathit{deciding}_{i'} = 0))$, so j cannot decide 1 at m_j.

If $m_j > m_i$, since i decides 0 in round $m_i + 1$ without using the common knowledge condition, we must have $\mathcal{I}, (r, m_i) \models \mathit{init}_i = 0 \bigvee_{j \in \mathit{Agt}} K_i(\mathit{jdecided}_j = 0)$. We can again apply Lemma A.5 to conclude that i receives a 0-chain at time m_i in r. As i is nonfaulty, j must hear from i in round $m_i + 1$,

so $\mathcal{I}, (r, m_j) \models K_j(jdecided_i = 0)$. It follows that agent j should decide 0 in this run, contradicting the assumption that j decides 1.

对于有效性，注意到如果代理 i 使用共同知识条件作出决定 v，则可知某个代理的初始偏好为 v。如果 i 未使用共同知识条件而作出决定 0，根据引理 A.5，必定存在一条 0-链，因此必有某个代理的初始偏好为 0。最后，如果代理 i 未使用共同知识条件而作出决定 1，那么 i 在第一轮没有作出决定 0，因此必有 初始$_i = 1$。

对于终止性，我们必须证明所有非故障代理都在第 $t + 2$轮之前作出决定。假设某个非故障代理 i 在运行 r中到第 $t + 1$ 轮仍未作出决定，并且 r' 是一个运行，满足 $(r' t + 1) \sim_i (r, t + 1)$。由于我们使用的是全信息协议，很容易推出，对于所有 $m \leq t+1$，$(r, m) \sim_i (r'm)$。由于 i在 r的第 $t+1$轮之前未作出决定，因此对于 $v \in \{0, 1\}$ 和 $m \leq t$，我们没有 I, $(r, m) \models K_i C_N$(t 个故障 $\wedge$未决定$_N(1 - v) \wedge \exists v)$。根据引理 A.3，很容易推出，对于 $v \in \{0, 1\}$ 和 $m \leq t$，I $(r' m) \models \neg C_N$(t 个故障 $\wedge$ 未决定$_N(1 - v)$ $\wedge \exists v)$。这意味着没有代理 j在 r' 中于第 $t + 1$轮或之前使用共同知识条件作出决定。

By Lemma A.5, if any agent j decides 0 in round $t + 1$ in r', then that agent receives a 0-chain $i_0, \ldots, i_t$ at time t in r'. Since agents on a chain are distinct, and there are at most t faulty agents, this chain contains at least one nonfaulty agent i_k that decides 0. But then i must receive a message (and a 0-chain) from i_k in round $k + 1$, which means that i decides 0 by round $t + 2$, as claimed.

We have shown if an agent decides by round $t + 1$ using the commnon knowledge conditions or decides 0 by line 4 of $\mathbf{P}^1$ in a run r' that i considers possible, then i decides by round $t + 2$ in r. If this is not the case, then $\mathcal{I}, (r, t + 1) \models K_i(\bigwedge_{j \in Agt} \neg(deciding_j = 0))$, so i decides 1 in round $t + 2$. □

A.2.6 $\mathbf{P}^1$ 是最优的。为了证明 $\mathbf{P}^1$ 是最优的，我们首先证明，只需证明弱安全性蕴含最优性，然后证明 $\mathbf{P}^1$ 是弱安全的。

定理 7.6。如果 $\mathbf{P}^1$ 相对于$\gamma_{fip,\ n,\ t}$ 是弱安全的，那么 $\mathbf{P}^1$ 的所有实现相对于 $\gamma_{fip,\ n,\ t}$都是最优的。

证明。假设 $\mathbf{P}^1$ 相对于 $\gamma_{fip,\ n,\ t}$是弱安全的。为了证明 $\mathbf{P}^1$ 的所有实现相对于 $\gamma_{fip,\ n,\ t}$都是最优的，我们使用定理 7.5。只需证明“仅当”方向，因为 [8] 中的命题 4.3 表明，对于 EBA 协议，“当”方向成立，而命题 7.3 表明 $\mathbf{P}^1$ 是一个 EBA 协议。

假设 $\mathbf{P}^1$ 相对于 $\gamma_{fip,n,t}$是弱安全的。令 P为 $\mathbf{P}^1$ 的一个实现，并令 I $= \mathrm{I}_{\gamma_{fip,n,t},P}$。我们给出 N $\wedge$ O 情形下的论证。首先假设对于某个点 (r, m)，I, $(r, m) \models i \in$N。用 (N $\wedge$ O)-□-可达性来表述，我们要证明：如果对于所有满足 $i \in$N (r') 的点 $(r' m)$，都有 I, $(r' m) \models \exists 0 \wedge$◯((已决定$_i = 1$))，并且对于所有从 $(r' m)$ (N $\wedge$O)-□-可达的点 $(r''m')$，都有 I, $(r'' m') \models \exists 0$，那么 I, (r, m) ⊨◯(已决定$_i = 0$)。反设，对于所有满足 $i \in$N (r') 的 $(r'm)$，上述条件都成立，但 I, $(r, m) \models$◯(已决定$_i = 0$) 不成立。于是，I, $(r, m) \models i \in$N △(已决定$_i = \perp$)，根据弱安全性，存在点 $(r^{1\prime}, m)$ 和 $(r^{1\prime\prime}, m)$，使得 $r_i(m) = r^{1\prime}{}_i(m)$，$i \in$N $(r^{1\prime})$，$(r^{1\prime\prime} m')$ 从 $(r^{1\prime}, m)$ (N $\wedge$O)-□-可达，并且 I, $(r^{1\prime\prime}, m')$ $\models \neg\exists 0$。这就产生了矛盾。由于这对所有 (r, m) 都成立，因此第一个最优性条件的“仅当”方向成立。N $\wedge$Z 情形的论证完全类似。

Therefore all implementations P of $\mathbf{P}^1$ with respect to $\gamma_{fip,n,t}$ are optimal. □

Let P be an implementation on $\mathbf{P}^1$ and let $\mathcal{I} = \mathcal{I}_{\gamma_{fip,n,t},P}$.. We want to show that $\mathbf{P}^1$ is weakly safe. So suppose that $\mathcal{I}, (r, m) \models i \in \mathcal{N} \wedge \bigcirc(decided_i = \perp)$. We need to show that there exist points $(r^{0\prime}, m)$, $(r^{0\prime\prime}\ m)$, $(r^{1\prime}\ m)$, and $(r^{1\prime\prime}\ m)$ satisfying the conditions of weak safety. Before we do this, we introduce an invariant.

Definition A.9 (Invariant condition for $v \in \{0,1\}$). $\mathcal{I}, (r,m) \models inv_v(i,j,k)$ if and only if there exist distinct agents i, j, k such that:

- $\mathcal{I}, (r,m) \models (decided_i = \bot) \wedge (deciding_j = v) \wedge (decided_k = \bot)$,
- $\mathcal{I}, (r,m) \models i \in \mathcal{N} \wedge j \in \mathcal{N} \wedge k \notin \mathcal{N}$, and
- k does not exhibit any faulty behaviour throughout r.

The motivation for the constraint on k in the definition is the following:

Lemma A.10. *If $k \notin \mathcal{N}(r)$ and k does not exhibit faulty behavior in r, then $\mathcal{I}, (r,m) \models \neg C_{\mathcal{N}}(t\text{-faulty})$ for all m.*

证明。若若 $k \notin \mathrm{N}(r)$，但未表现出故障行为，设 r'为一个运行，使得 $\mathrm{N}(r') = \mathrm{N}(r) - \{k\}$，$r$中的故障模式与 r'中的相同，并且所有代理在 r和 r'中的初始偏好相同。显然，对于所有时间 m，$(r,m) \sim_i (r',m)$，且 $\mathrm{I}, (r',m) \not\models t$-故障，因此 $\mathrm{I}, (r,m) \models \neg C_{\mathcal{N}}(t$-故障$)$。□

由此立即可得，若 $\mathrm{I}, (r,m) \models$不变式$_1(i,j,k)$，则没有代理会依据 r 中的共同知识条件作出决定。

现在我们可以构造弱安全性所需的点 $(r_i^{0''}, m)$ 和 $(r^{1''}, m)$。此情形论证的主要部分由引理 A.15 和 A.16 完成。引理 A.15 表明，一旦我们处于满足 1 的不变式条件的点，就可以通过 (N ∧O)-⊡ 到达一个每个代理的初始偏好均为 1 的点。引理 A.16 表明，如果一个非故障代理 i 无法作出决定，i 就必须认为存在一种可能的运行，其中满足 1 的不变式条件。所需结果由这两个引理得出。

The following technical lemma will play a key role in our proof of weak safety. For a point (r,m), we write $E(r,m)$ for the set of edges $(i,k-1) \rightarrow_r (j,k)$ with $1 \le k \le m$.

Lemma A.11. *If $i \in \mathcal{N}(r)$, $\mathcal{I}, (r,m) \models \neg C_{\mathcal{N}}(t\text{-faulty})$, $latest0_i(r,m) = \ell < m$, and $\mathcal{I}, (r,m) \models decided_i = \bot$, then there exists a run r' such that*

- $(r,m) \sim_i (r'\ m)$,
- $\mathcal{N}(r) = \mathcal{N}(r')$,
- $E(r'\ m) \subseteq E(r,m)$,
- *all 0-chains in r' are known to agent i at time m; that is, if $i_0, \ldots i_k$ is a 0-chain in r', then $(i_k,k) \rightarrow_{r'} (i,m)$,*
- *all agents that do not exhibit faulty behavior in r also do no not exhibit faulty behavior in r'.*

Proof. Note that since $\mathcal{I}, (r,m) \models decided_i = \bot$ and $latest0_i(r,m) < m$, we cannot have $\mathcal{I}, (r,m) \models deciding_i = 0$, so in fact agent i cannot hear from any agent that decides 0 in round m. This means that we must have $\ell = latest0_i(r,m) < m-1$. We construct r' by modifying r appropriately. For all $(j,m') \not\rightarrow_r (i,m)$, if $m' = 0$ and $\mathcal{I}, (r,0) \models init_j = 0$, then we modify $init_j$ to 1 in r', and if $m' > 0$, we restrict the messages arriving at (j,m') to be only those from the nonfaulty agents in $\mathcal{N}(r)$, and the agents that exhibit no faulty behavior in r. We have $\mathcal{N}(r') = \mathcal{N}(r)$.

Clearly, $E(r'\ m) \subseteq E(r,m)$. Since we do not modify the failure pattern for messages corresponding to pairs on the path $(j,m') \rightarrow_r (i,m)$, we have $(r,m) \sim_i (r'\ m)$. Moreover, the construction changes only the failure behaviour of agents who exhibit faulty behavior in r, so all agents that do not exhibit faulty behavior in r also do not exhibit faulty behavior in r'.

注意，由于 $\mathrm{I}, (r,\ m) \models i \in \mathrm{N} \wedge \neg C_{\mathcal{N}}(t$-故障$)$ 且 $(r,\ m) \sim_i (r',m)$，我们有 $\mathrm{I}, (r',m) \models \neg C_{\mathcal{N}}(t$-故障$)$，因此在 r' 中，没有代理会在第 $m+2$ 轮之前依据共同知识条件作出决定。特别是，在此轮之前作出的任何 0 决定，都必须依据基于知识的程序的第四行作出。

该构造保证，如果 $i_0, \ldots, i_\ell$ 是 r' 中的一条 0-链，那么 $(i_\ell, \ell) \rightarrow_{r'} (i, m)$。为证明这一点，首先注意到 $i_0, \ldots, i_\ell$ 中没有一个可能是无故障的；否则，i 会在 r 和 r' 中收到一条 0-链，并在时间 m 之前作出 0 决定，这与 $\mathrm{I}, (r, m) \models$ 正在作出决定的假设矛盾。如果 $\ell > 0$，那么 i_ℓ 在 r' 中收到来自 $i_{\ell-1}$ 的消息这一事实，加上 $i_{\ell-1}$ 是故障的，意味着 $(i_\ell, \ell) \rightarrow_r (i, m)$。由此很容易推出 $(i_\ell, \ell) \rightarrow_{r'} (i, m)$。因此，$i$ 知道 r' 中存在这条 0-链。如果 $\ell = 0$，那么要么 $(i_0, 0) \rightarrow_r (i, m)$，从而 $(i_0, 0) \rightarrow_{r'} (i, m)$；要么 i_0 的初始值在 r' 中被改成了 1，因此这实际上并不是一条 0-链。

接下来要证明，r' 不存在长度大于 ℓ 的 0-链。反设 $i_0, \ldots, i_{\ell+1}$ 是 r' 中长度为 $\ell + 1$ 的 0-链。若 $\ell \geq 0$，那么由于 $\mathsf{latest}0_i(r, m) = \ell$，我们必有 $(i_{\ell+1}, \ell + 1) \not\rightarrow_{r'} (i, m)$；否则就会有 $(i_{\ell+1}, \ell + 1) \rightarrow_r (i, m)$，而 $i_0, \ldots, i_{\ell+1}$ 将是 r 中长度为 $\ell + 1 > \mathsf{latest}0_i(r, m)$ 的 0-链，这就产生了矛盾。若 $\ell = -1$（即 i 不知道 r 中任何 0-链的存在），假设 r' 中存在 0-链。那么，r' 中必定有某个初始偏好为 0 的代理 i_0。我们不可能有 $(i_0, 0) \rightarrow_r (i, m)$，否则 i 就会知道 r 中存在 0-链。但这样一来，我们的构造就保证了 r' 中 i_0 的初始偏好为 1，而非 0。由此可知，r' 中不存在长度至少为 $\ell + 1$ 的 0-链，证毕。□

引理 *A.12.* 若 $m \geq 2$、I、$(r, m) \models \mathit{inv}_1(i, j, k)$，且 $\mathit{latest}0_j(r, m) \leq m - 3$，则存在一个运行 r^*，使得 I、$(r^*, m - 1) \models \mathit{inv}_1(i', j' k')$，其中某些 $i', j' k'$ 属于 Agt，$E(r^*, m - 2) \subseteq E(r, m - 2)$，且 $N(r) \cup \{k\} = N(r^*) \cup \{k'\}$。

证明。根据引理 A.11，存在一个点 $(r^\dagger, m) \sim_j (r, m)$，使得 $\mathrm{N}(r) = \mathrm{N}(r\dagger)$，$r\dagger$ 中最长的 0-链长度至多为 $m-3$，且 $E(r\dagger m) \subseteq E(r, m)$。此外，由于 k 在 r 中未表现出任何故障行为，因此在 $r\dagger$ 中也同样如此。另外，由于 $\mathrm{I}, (r, m) \models$已决定$_i = \perp \wedge$已决定$_j \neq \perp \wedge$已决定$_k = \perp$，我们也有 $\mathrm{I}, (r^\dagger, m) \models$已决定$_i = \perp \wedge$已决定$_j = \perp \wedge$已决定$_k = \perp$。

We construct a run r' by modifying $r^\dagger$ so that agent i is faulty, k is nonfaulty, and agent k hears from every other agent except agent i in round m. Since the only modifications are in round m, we have $E(r'\ m-1) = E(r^\dagger\ m-1) \subseteq E(r, m-1)$. Agent j is nonfaulty in both r and r' and has the same local state in both (r, m) and $(r'\ m)$. Hence, $j \in (\mathcal{N} \wedge \mathcal{O})(r, m) \cup (\mathcal{N} \wedge \mathcal{O})(r'\ m)$; therefore, $(r'\ m)$ is $(\mathcal{N} \wedge \mathcal{O})$-⊡-reachable from (r, m) through agent j. Moreover, the length of the longest 0-chain in r' is $m-3$, because we do not modify $r^\dagger$ in round $m-2$ or earlier in constructing r'. Since $m \geq 2$ and $\mathit{latest}0_k(r'\ m) \leq m-3$, in r', agent k decides 1 in round $m+1$ by Corollary A.8.

接下来考虑运行 r''，它与 r' 完全相同，只是在第 $m-1$ 轮中，代理 i 收到了每个代理的消息。这里有 $E(r''\, m-2) = E(r'\, m-2) \subseteq E(r, m-2)$。由于代理 k 在第 m 轮收到的消息完全相同，因此我们有 $k \in (\mathrm{N} \wedge \mathrm{O})(r'm) \cup (\mathrm{N} \wedge \mathrm{O})(r''m)$。因此，$(r''m)$ 可通过代理 k 从 $(r'm)$ 经由 $(\mathrm{N} \wedge \mathrm{O})$-⊡-可达。由于 r' 中最长的 0-链长度为 $m-3$，并且在构造 r'' 时，我们没有改变第 $m-2$ 轮或更早轮次中的 r'，因此 r'' 中最长的 0-链长度也为 $m-3$。根据推论 A.8，代理 i 在 r'' 的第 m 轮中，收到第 $m-1$ 轮每个代理的消息后，作出决定 1。由于代理 j 收到了代理 i 的消息，而且 j 在运行 r'' 中没有更早作出决定，因此代理 j 在 r'' 的第 $m+1$ 轮作出决定 1。

Now consider the run r^* that is identical to r'' except that agent i is in $\mathcal{N}$ (so that agent k does get a message from agent i in round m), and k is faulty in $\mathcal{N}$ (but does not exhibit any faulty behavior.) Clearly, $E(r^*, m-2) = E(r''\ m-2) \subseteq E(r, m-2)$. Since j has the same local state in both $(r''\ m)$ and (r^*, m), and is nonfaulty in both r'' and r^*, we have $j \in (\mathcal{N} \wedge \mathcal{O})(r''\ m) \cup (\mathcal{N} \wedge \mathcal{O})(r^*, m)$. Therefore, (r^*, m) is $(\mathcal{N} \wedge \mathcal{O})$-⊡-reachable from $(r''\ m)$ through agent j.

Finally, observe that in run r^*, i is nonfaulty and decides 1 in round m, k is nonfaulty and does not decide before round m, and j is faulty but exhibits no faulty behavior and does not decide before round m. Hence, $\mathcal{I}, (r^*, m-1) \models \mathit{inv}_1(k, i, j)$. By the transitivity of the $(\mathcal{N} \wedge \mathcal{O})$-⊡-reachability relation, $(r^*, m-1)$ is $(\mathcal{N} \wedge \mathcal{O})$-⊡-reachable from (r, m), and the claim holds with $(i', j'\ k') = (k, i, j)$. Each step of the construction involved a swap of faultiness between a faulty

and nonfaulty agent among i, j, and k, leaving the faultiness of other agents invariant, so we have $\mathcal{N}(r^*) \cup \{k'\} = \mathcal{N}(r) \cup \{k\}$. □

下文中，我们用 $r[0..m]$ 表示运行 r 截至时间 m（包括该时刻）的前缀，但移除环境状态中有关哪些代理无故障的信息；并用 $发送方_i(r, m)$ 表示满足 $(i, m-1) \rightarrow_r (j, m)$ 的代理 j 集合。

引理 A.13。若 $I, (r, m) \models inv_i(i, j, k)$，且 r 中最长的 0-链长度至多为 $m-2$，则对所有 $p \notin N(r) \cup \{k\}$，都存在一个 $(N \wedge O)$-⊡- 可达点 $(r'm')$，满足 $m' \in \{m, m+1\}$，且 $r[0..m-1] = r'[0..m-1]$，$I, (r'm') \models inv_i(i', j'k')$，$p \notin \{i', j'k'\}$，以及 $(p, m-1) \not\rightarrow_{r'} (j'm')$。

证明。我们构造一条 (N ∧O)-⊡-路径来证明该结果。我们指出，inv_1 通常不会沿着这条路径保持成立：我们使inv_1失效，但在最后一步重新使其成立。该构造分为若干分支，如图 1 所示。图中只展示了为得到每次后续运行所做的更改，并在因果图中追踪故障边。代理的时间线沿水平方向绘制。故障边以虚线表示，标记为 1 的节点表示该节点处代理正在决定 1。

如果 $(p, m-1) \not\rightarrow_r (j, m)$，我们可以取 $(r'm') = (r, m)$ 和 $(i', j'k') = (i, j, k)$，这样就完成了。否则，考虑通过修改 r 得到的运行 r^1，使得在第 m 轮中，p 发给除 j 和其自身以外所有代理的消息都丢失，即 $\text{Sends}^p(r^1, m) = \{p, j\}$。我们有 $\text{N}(r^1) = \text{N}(r)$。由于 $(r, m) \sim_j (r^1, m)$ 且 $j \in (\text{N} \wedge \text{O})(r, m) \cap (\text{N} \wedge \text{O})(r^1, m)$，因此 (r^1, m) 可从 (r, m) 通过 (N ∧O)-⊡ 到达。

在 r^1 中，i、j 和 k 仍然没有表现出故障行为，且在时间 m 时尚未作出决定。由于 k 存在故障，但在 r^1 中没有表现出故障行为，根据引理 A.10，没有代理会使用 r^1 中的共同知识条件作出决定。长度大于 $m-2$ 的 0-链仍不存在，因此在 r^1 中，经过 $m-1$ 轮后，没有代理会决定 0。如果 $\text{I}, (r^1, m) \models$ 作出决定$_i = 1$，那么取 $(r'm') = (r^1, m)$ 和 $(i', j'k') = (j, i, k)$ 即可得证，因为 $i \in \text{N}(r^1)$，而 $(p, m-1) \not\rightarrow_{r^1} (i, m)$，且 $r[0..m-1] = r^1[0..m-1]$ 根据构造成立。另一方面，如果 $\text{I}, (r^1, m) \models$ 作出决定$_k = 1$，那么同理，取 $(r'm') = (r^{1'}, m)$ 和 $(i', j'k') = (j, k, i)$ 即可得证，其中 $r^{1'}$ 是通过令 $\text{N}(r^{1'}) = \text{N}(r) \cup \{k\} \setminus \{i\}$ 从 r^1 得到的运行。

如果 $\text{I}, (r^1, m) \models$正在作出决定$_i = \perp \wedge$正在作出决定$_k = \perp$，则有 $\text{I}, (r^1, m+1) \models$正在作出决定$_i = 1$，因为 $\text{I}, (r^1, m) \models$正在作出决定$_j = 1$且 $j \in \text{N}(r^1)$。令 r^2为通过修改 r^1得到的运行，使得在第 $m+1$ 轮中，代理 i从除 p之外的所有代理接收消息。我们有 $\text{N}(r^2) = \text{N}(r)$。由于 $(r, m) \sim_j (r^2, m)$ 且 $j \in (\text{N} \wedge \text{O})(r, m) \cap (\text{N} \wedge \text{O})(r^1, m)$，因此 (r^2, m) 可从 (r, m) 经 (N ∧O)-⊡ 到达。

仍然是这样的情况：k发生故障，但在 r^2 中没有表现出故障行为，因此没有代理依据 r^2 中的共同知识条件作出决定。注意，$r^2[0..m-1] = r^1[0..m-1] = r[0, m-1]$和 $\text{Sends}^p(r^2, m) = \text{Sends}^p(r^1, m) = \{p, j\}$。我们还有 $r_i^2(m) = r_i^1(m)$ 且 $r_k^2(m) = r_k^1(m)$，所以 $\text{I}, (r^1, m) \models$正在作出决定$_i = \perp \wedge$正在作出决定$_k = \perp$。因此，利用推论 A.8，我们有 $\text{I}, (r^2, m+1) \models$正在作出决定$_i = 1$，因为 r^2 中最长的 0-链长度至多为 $m-2$。

Let r^3 be the run obtained by modifying r^2 so that $\mathcal{N}(r^3) = \mathcal{N}(r) \cup \{k\} \setminus \{j\}$, and agent k receives a message from all agents but j in round $m+1$. Since $(r^2, m) + 1 \sim_i (r^3, m+1)$ and $i \in (\mathcal{N} \wedge \mathcal{O})(r^2, m+1) \cap (\mathcal{N} \wedge \mathcal{O})(r^3, m+1)$, we have that (r^3, m) is $(\mathcal{N} \wedge \mathcal{O})$-⊡ reachable from (r, m). We remark that inv_1 does not hold in r^3 with a permutation of i, j, k, since j now exhibits faulty behavior. However, we still have $\mathcal{I}, (r^3, m+1) \models \neg C_{\mathcal{N}}(t\text{-}faulty)$, because $i \in \mathcal{N}(r^3)$ and i does not observe j's faulty behavior. It follows that no agent decides in r^3 using the common knowledge conditions before round $m+3$.

Note that $Sends_p(r^3, m) = Sends_p(r^2, m) \setminus \{j\} = \{p, j\}$ and $Sends_j(r^3, m+1) = Agt \setminus \{k\}$. We have $r^3[0..m-1] = r^2[0..m-1] = r[0, m-1]$. This implies that none of i, j, k has decided at time m. In addition, $r_i^3(m) = r_i^2(m)$ and $r_k^3(m) = r_k^2(m)$, so neither i nor k decides in round $m+1$. By

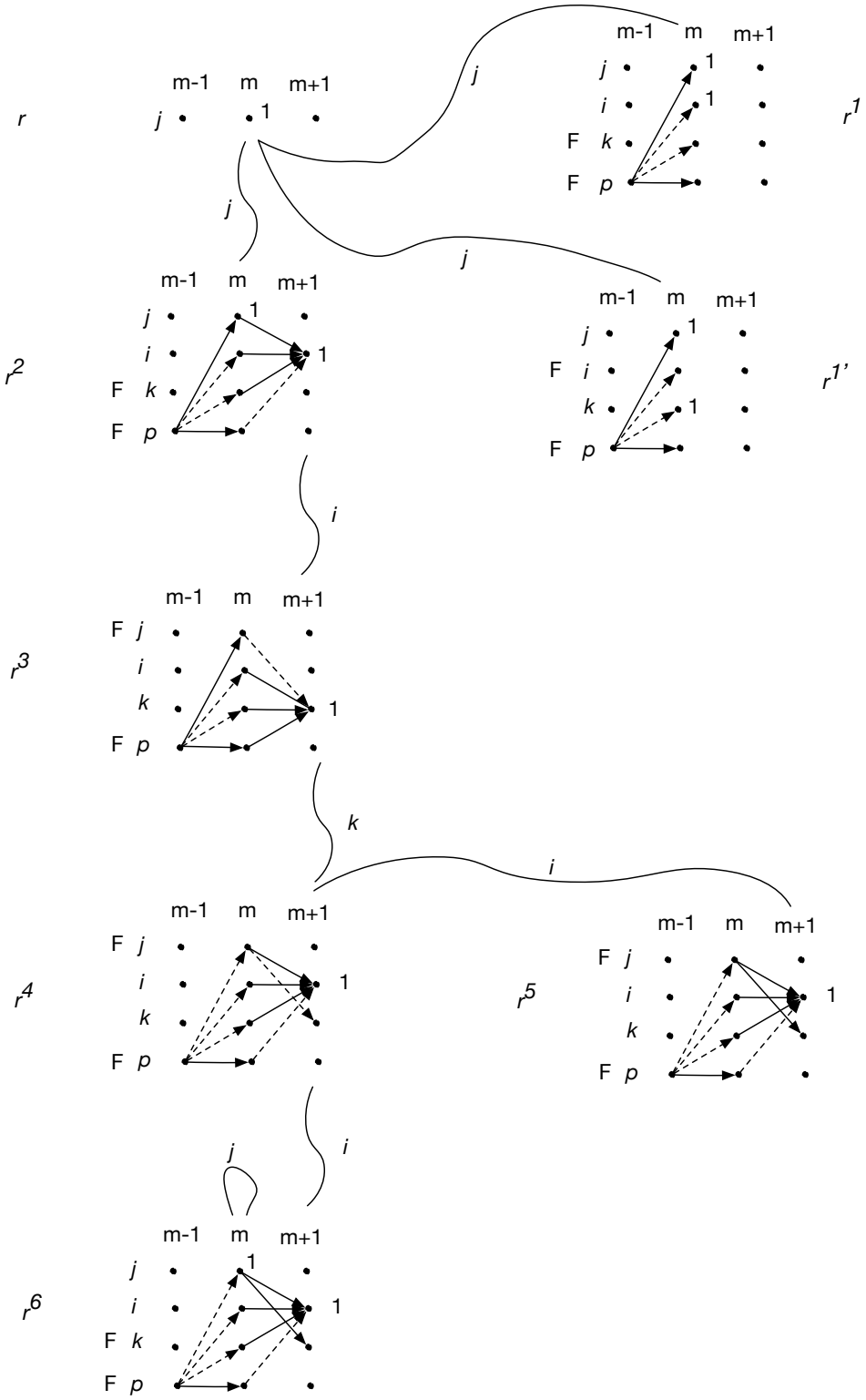


Fig. 1. Construction for Lemma A.13

由推论 A.8 可知，$\mathrm{I}, (r^3,\ m+1) \models$作出决定$_k = 1$，因为 r^3 中最长的 0-链长度至多为 $m-2$。

令 r^4 为通过修改 r^3 得到的运行，使得代理 p 在第 m 轮不向代理 j 发送任何消息，并且在第 $m+1$ 轮，除 p 外的所有代理都向代理 i 发送一条消息。我们有 $\mathrm{N}(r^4) = \mathrm{N}(r^3) = \mathrm{N}(r) \cup \{k\} \setminus \{j\}$ 和 $r^4[0..m-1] = r^3[0..m-1] = r[0, m-1]$。此外，$\mathit{Sends}_P(r^4, m) = \mathit{Sends}_P(r^3, m) \setminus \{j\} = \{p, j\} \setminus \{j\} = \{p\}$，并且 $\mathit{Sends}_j(r^4, m+1) = \mathit{Sends}_j(r^3, m+1) = \mathit{Agt} \setminus \{k\}$。由于在 r^3 中，代理 j 在第 $m+1$ 轮没有向代理 k 发送消息，因此第 m 轮的变化在时间 $m+1$ 对代理 k 不可见，第 $m+1$ 轮的变化在时间 $m+1$ 对 k 也不可见。因此，我们有 $(r^3, m+1) \sim_k (r^4, m+1)$。此外，$k \in (\mathrm{N} \wedge \mathrm{O})(r^3, m+1) \cap (\mathrm{N} \wedge \mathrm{O})(r^4, m+1)$，所以 $(r^4, m+1)$ 可从 (r, m) 经 $(\mathrm{N} \wedge \mathrm{O})$-⊡ 到达。我们有 $r_i^4(m) = r_i^3(m)$，因此代理 i 在 r^4 中不会在第 $m+1$ 轮作出决定。由推论 A.8 可知，$\mathrm{I}, (r^4, m+1) \models$作出决定$_i = 1$。

在 r^4 中，代理 j 仍然表现出故障行为。为恢复 inv_1,，令 r^5 为通过将第 $m+1$ 轮中从 j 到 k 的失败消息改为成功传输，从 r^4 得到的运行。由于这是此前引入的 j 唯一一次故障，这确保 j 在 r^5 中不会表现出故障行为。我们有 $\mathrm{N}(r^5) = \mathrm{N}(r^4) = \mathrm{N}(r) \cup \{k\} \setminus \{j\}$，且 $r^4[0..m-1] = r^3[0..m-1] = r[0, m-1]$。后者意味着在第 $m-1$ 轮之后，没有代理会作出决定 0。构造 r^5 时所做的更改在时间 $m+1$ 对代理 i 不可见，因此我们有 $(r^4, m+1) \sim_i (r^5, m+1)$。我们有 $i \in (\mathrm{N} \wedge \mathrm{O})(r^4, m) \cap (\mathrm{N} \wedge \mathrm{O})(r^5, m)$，因此 (r^5, m) 可从 (r, m) 经由 $(\mathrm{N} \wedge \mathrm{O})$-⊡ 到达。

此外，$r_i^5(m) = r_i^4(m)$，因此代理 i在运行 r^5的第 $m+1$轮不会作出决定，因为它在 r^4的第 $m+1$轮也没有作出决定。类似地，$r_k^5(m) = r_k^4(m)$，因此代理 k在运行 r^5的第 $m+1$轮不会作出决定。对于代理 j，我们有两种可能。

- 如果 $\mathrm{I}, (r^5, m) \models$作出决定$_j = 1$，那么令 r^6为与 r^5相同的运行，唯一的区别是 $\mathrm{N}(r^6) = \mathrm{N}(r)$。（也就是说，我们将 k从在 r^5中无故障切换为在 r^6中故障，并将 j从故障切换为无故障。）仍有 $(r^4, m+1) \sim_i (r^6, m+1)$ 且 $i \in (\mathrm{N} \wedge \mathrm{O})(r^4, m+1) \cap (\mathrm{N} \wedge \mathrm{O})(r^6, m+1)$，因此 $(r^6, m+1)$ 可从 (r, m) 通过 $(\mathrm{N} \wedge \mathrm{O})$-⊡到达。我们还有 $\mathrm{I}, (r^6, m) \models$作出决定$_j = 1$，因为 $r_j^6(m) = r_j^5(m)$。由于 $(r^6, m) \sim_j (r^6, m)$ 且 $i \in (\mathrm{N} \wedge \mathrm{O})(r^6, m) \cap (\mathrm{N} \wedge \mathrm{O})(r^6, m)$，因此 (r^6, m) 可从 (r, m) 通过 $(\mathrm{N} \wedge \mathrm{O})$-⊡到达。注意，$(p, m-1) \not\to_{r^6} (j, m)$。因此，取 $(r'\ m') = (r^6, m)$ 以及 $(i'\ j'\ k') = (i, j, k)$，即可得证。
- If $\mathcal{I}, (r^5, m) \models \neg \mathit{deciding}_j = 1$, then we have $\mathcal{I}, (r^5, m+1) \models \mathit{decided}_i = \mathit{decided}_j = \mathit{decided}_k = \bot$, and $\mathcal{I}, (r^5, m+1) \models \mathit{deciding}_i = 1$. Moreover, $\mathit{Sends}_p(r^5, m) = \{p\}$ and $i \notin \mathit{Sends}_p(r^5, m+1)$. Thus $(p, m-1) \not\to_{r^5} (i, m+1)$. Hence we are done, taking $(r'\ m') = (r^5, m+1)$ and $(i'\ j'\ k') = (k, i, j)$.

□

The following lemma allows us to reduce the size of the set $S_{m-1}(r)$, so that there are fewer paths by which a 0-chain reaching time $m-2$ is visible at time m.

Lemma A.14. *If $\mathcal{I}, (r, m) \models \mathit{inv}_1(i, j, k)$ and the longest 0-chain in r has length at least $m-2$, then there exists a run r' and a permutation $i'\ j'\ k'$ of i, j, k such that $\mathcal{I}, (r'\ m) \models \mathit{inv}_1(i'\ j'\ k')$, $\mathcal{N}(r) \cup \{k\} = \mathcal{N}(r') \cup \{k'\}$, the point $(r'\ m)$ is $(\mathcal{N} \wedge \mathcal{O})$-⊡-reachable from (r, m), and there are no 0-chains in r' of length $m-2$.*

Proof. Consider an agent $q \in \mathit{Agt}(\mathcal{N}(r) \cup \{k\})$. We write $F_q(r, m-1)$ for the set of edges of the form $(p, m-2) \to (q, m-1)$ in $E(r, m-1)$, where $p \in \mathit{Agt} \setminus \mathcal{N}(r) \cup \{k, q\}$.

We first show that there exists a point (r^q, m) that is $(\mathcal{N} \wedge \mathcal{O})$-⊡-reachable from (r, m) such that $\mathcal{I}, (r^q, m) \models \mathit{inv}_1(i^q, j^q, k^q)$, where $\{i, j, k\} = \{i^q, j^q, k^q\}$, $\mathcal{N}(r) \cup \{k\} = \mathcal{N}(r^q) \cup \{k^q\}$, and $E(r^q, m-1) \subseteq E(r, m-1) \setminus F_q(r, m-1)$. To do this, we first apply Lemma A.13 to (r, m) and

$(N \wedge O)$-⊡-reach a point (r^1, m_1) such that $\mathcal{I}, (r^1, m_1) \models inv_1(i_1, j_1, k_1)$, $N(r) \cup \{k\} = \mathcal{N}(r^1) \cup \{k_1\}$, $E(r^1, m-1) = E(r, m-1)$, $(q, m-1) \not\rightarrow_{r^1} (j_1, m_1)$, and m_1 is either m or $m+1$.

令 r^2 为通过将 $F_q(r, m-1)$ 中所有边 $(p, m-2) \rightarrow_{r^1} (q, m-1)$ 转换为故障，从 r^1 得到的运行，因此 $E(r^2, m-1) = E(r^1, m-1) \setminus F_q(r, m-1) = E(r, m-1) \setminus F_q(r, m-1)$。我们有 $(r^2, m_1) \sim_{j_1} (r^1, m_1)$，且 $\mathrm{N}(r^1) = \mathrm{N}(r^2)$，所以 (r^2, m_1) 可从 (r^1, m_1) 经 $(N \wedge O)$-⊡ 到达。仍然有 i_1、j_1、k_1 均未在 (r^2, m_1) 作出决定，因为这些代理在时间$m_1 - 1$的状态在 r^1 中对 j_1 在时间 m_1 可见，且 $(r^2, m_1) \sim_{j_1} (r^1, m_1)$。因此 $\mathrm{I}, (r^2, m_1) \models \mathrm{inv}_1(i_1, j_1, k_1)$，且 $\mathrm{N}(r) \cup \{k\} = \mathrm{N}(r^2 \cup \{k_1\}$。

若 $m_1 = m$ 我们便完成了，取 $r^q = r^2$；否则，$m_1 = m+1$，并应用引理 A.12，得到一个点 (r^q, m)，它可从 (r, m) 经 $\mathrm{I}, (r^q, m_1) \models \mathrm{inv}_1(i_q, j_q, k_q)$ 进行 $(\mathrm{N} \wedge \mathrm{O})$-⊡可达，且 $\mathrm{N}(r^q) \cup \{k^q\} = \mathrm{N}(r^2) \cup \{k_1\} = \mathrm{N}(r) \cup \{k\}$。我们有 $E(r^q, m-1) = E(r^q, m_1 - 2) \subseteq E(r^2, m_1 - 2) = E(r^2, m-1) = E(r, m-1) \setminus F_q(r, m-1)$。

我们依次为 $\mathrm{Agt} \setminus (\mathrm{N}(r) \cup \{k\})$ 中的所有代理重复上述步骤，从而通过 $(N \wedge \mathrm{O})$-⊡-到达一个点 (r^3, m)，满足 $\mathrm{I}, (r^3, m) \models \mathrm{inv}_1(i_3, j_3, k_3), \mathrm{N}(r^3) \cup \{k_3\} = \mathrm{N}(r) \cup \{k\}$，且$E(r^3, m-1) \subseteq E(r, m-1) \setminus (\bigcup_{q \in \mathrm{Agt} \setminus (\mathrm{N}(r) \cup \{k\})} F_q(r, m-1))$. 也就是说，在 r^3 的第 $m-1$ 轮中，$\mathrm{Agt} \setminus (\mathrm{N}(r) \cup \{k\})$ 中不同的（故障）代理之间不会传递消息。

仍可能存在这样的故障代理 q：它们在 r^3 的第 $m-1$ 轮作出决定，且对于这些代理，$(q, m-1) \rightarrow_{r^3} (j', m)$。在这种情况下，我们再次将引理 A.13 应用于 (r^3, m)，并通过 $(\mathrm{N} \wedge \mathrm{O})$-⊡-可达性到达点 (r^4, m_4)，使得 $\mathrm{I}, (r^4, m_4) \models \mathrm{inv}_1(i_4, j_4, k_4)$，$\{i_4, j_4, k_4\} = \{i, j, k\}$，且 $\mathrm{N}(r^4) \cup \{k_4\} = \mathrm{N}(r^3) \cup \{k_3\} = \mathrm{N}(r) \cup \{k\}$，$r^4[0 \ldots m-1] = r^3[0 \ldots m-1]$，$(q, m-1) \not\rightarrow_{r^4} (j_1, m_4)$，并且 m_4 要么为 m，要么为 $m+1$。此时，我们应用引理 A.11 的构造，从 r^4 生成一个运行r^5，使得 (r^5, m_4) 可通过代理 j_4 从 (r^4, m_4) 经 $(\mathrm{N} \wedge \mathrm{O})$-⊡-可达，$\mathrm{I}, (r^5, m_4) \models \mathrm{inv}_1(i_4, j_4, k_4)$，$\mathrm{N}(r^5 \cup \{k_4\} = \mathrm{N}(r^4) \cup \{k_4\} = \mathrm{N}(r) \cup \{k\}$，所有 0- 链均对 (j_4, m_4) 可见，并且 $E(r^5, m-1) \subseteq E(r^4, m-1) \subseteq E(r, m-1) \setminus (\bigcup_{q \in \mathrm{Agt} \setminus (\mathrm{N}(r) \cup \{k\})} F_q(r, m-1))$。

We claim that q does not decide in round $m-1$ of r^5. Obviously, we do not have an edge from $(q, m-2)$ into $(\mathcal{N}(r) \cup \{k\}) \times \{m-1\}$ in r^4, otherwise r would have a 0-chain of length at least $m-1$. We also do not have an edge in r^4 from $(q, m-2)$ into $(\mathcal{N}(r) \cup \{k\}) \times \{m-1\}$, since all such edges have been eliminated. Thus, the only edge from $(q, m-2)$ is to $(q, m-1)$. Since $(q, m-1) \not\rightarrow_{r^4} (j_4, m_4)$, it follows that $(q, m-2) \not\rightarrow_{r^4} (j_4, m_4)$, and the construction of r^5 ensures that $\mathcal{I}, (r^5, m-2) \models \neg deciding_q = 0$.

如果 $m_4 = m+1$，我们应用引理 A.12，得到一个点 (r^6, m)，它从 (r, m) 出发是 $(\mathrm{N} \wedge \mathrm{O})$-⊡可达的，满足 $E(r^6, m-1) \subseteq E(r^5, m-1)$，$\mathrm{I}, (r^6, m) \models \mathrm{inv}_1(i_6, j_6, k_6)$，$\{i_6, j_6, k_6\} = \{i, j, k\}$，$\mathrm{N}(r^6) \cup \{k_6\} = \mathrm{N}(r) \cup \{k\}$，并且 q在第 $m-1$ 轮不作出决定。而如果 $m_4 = m$，我们已经通过 $r^6 = r^5$ 得到了这一结论。

我们依次对所有 $q \in \mathrm{Agt} \setminus (\mathrm{N}(r) \cup \{k\})$ 应用此构造，直到得到一个 $(\mathrm{N} \wedge \mathrm{O})$-⊡-可达点 (r^7, m)，使得 $\mathrm{I}, (r^7, m) \models \mathrm{inv}_1(i_7, j_7, k_7), \{i_7, j_7, k_7\} = \{i, j, k\}, \mathrm{N}(r^7) \cup \{k_7\} = \mathrm{N}(r^6) \cup \{k_6\} = \mathrm{N}(r) \cup \{k\}$，$E(r^7, m-1) \subseteq E(r^4, m-1)$，且对于所有$q \in \mathrm{Agt} \setminus (\mathrm{N}(r) \cup \{k\})$，$\mathrm{I}, (r^7, m-2) \models \neg$作出决定$_q = 0$。由于 $\mathrm{I}, (r^7, m) \models \mathrm{inv}_1(i_7, j_7, k_7)$ 蕴含 $\mathrm{N}(r^7) \cup \{k_7\} = \mathrm{N}(r) \cup \{k\}$ 中没有代理会在第 $m-1$ 轮作出决定，我们得出结论：r^7 没有长度为 $m-2$ 的 0-链。因此，我们得到 $r' = r^7$ 的结果。□

Lemma A.15. *If $\mathcal{I}, (r, m) \models inv_1(i, j, k)$, then there exists a run $\hat{r}$ such that $(\hat{r}, 0)$ is $(\mathcal{N} \wedge \mathcal{O})$-⊡-reachable from (r, m) and $\mathcal{I}, (\hat{r}, 0) \models \neg \exists 0$.*

Proof. We proceed by induction on m. Suppose that $\mathcal{I}, (r, m) \models inv_1(i, j, k)$, so that j decides 1 in round $m+1$ of r. No agent can decide 1 in round 1, so we have $m \geq 1$.

If $m = 1$, agent j can't know $\exists 0$ at time 1 right before j decides 1 in round 2, as otherwise j would have gotten a 0-chain and, by Lemma A.5, would decide 0. Hence, j considers possible the

run $\hat{r}$ where no agent has an initial preference of 0 and the message pattern is identical to r. Clearly, $(\hat{r}, 1)$ is $(\mathcal{N} \wedge \mathcal{O})$-$\Box$-reachable from $(r, 1)$ through agent j.

若 $m > 0$，设 $\text{len}_j(r, m) = m'$。注意，j无法得知长度为 $m-1$或更长的 0-链，因为在这种情况下，j会在第 $m+1$轮作出 0 的决定。若 $m' < m-2$，则我们可以立即将引理 A.12 应用于 (N $\wedge$O)-$\Box$-可达性，从而到达一个点 $(r^*, m-1)$，在该点上 1 的不变式条件成立。接着，我们可以应用归纳假设，得出存在一个运行 $\hat{r}$，使得 $\text{I}, (\hat{r}, 0) \models \neg\exists 0$，且 $(\hat{r}, 0)$ 可从 $(r^*, m-1)$ 经 (N $\wedge$O)-$\Box$-可达。根据 (N $\wedge$O)-$\Box$-可达关系的传递性，$(\hat{r}, 0)$ 也可从 (r, m) 经 (N $\wedge$O)-$\Box$-可达，因此结论成立。若 $m' = m-2$，则根据引理 A.14，存在一个点 $(r'm)$，它可从 (r, m) 经 (N $\wedge$O)-$\Box$ 可达，并且 $\text{I}, (r'm) \models \text{inv}_1(i_1, j_1, k_1)$，且 r' 中所有 0-链的长度都小于 $m-2$。将前一种情况应用于 $(r'm)$，即可得出结论。□

引理 A.16. 如果 $\mathcal{I}, (r, m) \models (i \in \mathcal{N} \wedge \bigcirc(\text{已决定}_i = \perp))$，那么 i认为点 (r, m)在 (r, m)时可能，其中 $\mathcal{I}, (r, m) \models inv_1(i, j, k)$。

证明。假设在某个点 (r, m)，$\text{I}, (r, m) \models i \in \text{N} \wedge \bigcirc(\text{已决定}_i = \perp)$。我们首先注意到，必有 $\text{I}, (r, m) \models \neg C_{\mathcal{N}}(t\text{-故障})$，否则根据引理 A.4 就会得到矛盾。为了说明 i认为点 (r, m)在 (r, m) 时可能，且 $\text{I}, (r, m) \models inv_1(i, j, k)$，我们分两种情况讨论：

- 假设 $\text{I}, (r, m) \models \neg K_i(\text{t 个故障})$。根据定义，存在一个代理 k，使得 $\text{I}, (r, m) \models \neg K_i \neg(k \in \text{N}) \wedge \neg K_i \neg(k \notin \text{N})$。令 r' 为一个与 r 相同的运行，但 $k \notin \text{N}$，k 在 r' 中不表现出任何无故障行为，某个代理 $j \in \text{N}$ 在第 m 轮听到所有其他代理的消息，并且运行 r' 中唯一的 0-链是 i 知道的那些。这意味着 r' 中的 0-链长度至多为 $m-2$，因此最终所有代理都应作出决定 1。我们声称 $(r, m) \sim_i (r'm)$。反设 i 在 (r, m) 和 $(r'm)$ 中的局部状态不同。那么，要么 i 在 r' 中听到某个代理的消息，而该代理在 r 的第 m 轮没有收到来自 k 的消息，且该代理在 r 的第 m 轮听到了来自 k 的消息；要么 i 听说了 r' 的第 m 轮中的一条 0-链，而它在 r 中的 m 时间并不知道这条链。在第一种情况下，i 本会知道 k 在 (r, m) 中有故障，这就产生了矛盾；第二种情况则与假设矛盾，即 i 不知道 r' 中的这类 0-链。

  根据推论 A.8，代理 j 在轮次 $m+1$ 收到第 m 轮中所有其他代理的消息后，作出决定 1。因此，$\text{I}, (r, m) \models inv_1(i, j, k)$ 且 $(r, m) \sim_i (r, m)$。
- 如果 $\text{I}, (r, m) \models K_i(\text{t 个故障})$，我们便利用如下观察：$\text{I}, (r, m) \models \neg C\text{N}(\text{t 个故障})$。根据命题 A.2，必有 $\text{I}, (r, m-1) \models \neg\text{distN}(\text{t 个故障})$；等价地，对于某个故障代理 k，$\text{I}, (r, m) \models \forall j \in \text{N}(\neg K_j \neg(k \in \text{N}))$。由于 $i \in \text{N}$，我们必有 $\text{I}, (r, m-1) \models \neg K_i \neg(k \in \text{N})$。由于 $\text{I}, (r, m) \models K_i(\text{t 个故障})$，代理 i 必定在轮次 m 得知 k 发生故障；这可能是因为它首次没有收到来自 k 的消息，也可能是因为它收到了某个故障代理 k' 的消息，而该代理已知代理 k 发生故障。代理 k' 不可能是非故障代理，因为在时间 $m-1$，没有非故障代理知道 k 发生故障。如果 i 是通过在第 m 轮没有收到来自 k 的消息而得知 k 发生故障，那么在 (r, m)，代理 i 会认为点 $(r'm)$ 是可能的；在 r' 的第 m 轮，某个非故障代理 j 收到了所有其他代理的消息。根据推论 A.8，在 r' 中，j 于轮次 m 作出决定 1。因此，$\text{I}, (r'm) \models \text{inv1}(i, j, k)$，正如所需。

  If i learned that k is faulty by hearing from some faulty agent k' that k is faulty, then i considers possible the run r' which is identical to r except that j and k receive a message from all other agents except k' in round m. By Corollary A.8, both agents j and k are about to decide 1 at $(r'\ m)$, since all 0-chains in r' have length at most $m-2$ and they both receive a message from all agents but k'. We must have $(r, m) \sim_i (r'\ m)$, as the only change between

the runs is in round m, and i receives the same messages in both runs. Moreover, i is nonfaulty in both runs.
现在考虑一次运行 r''，在其中 i 在第 m 轮及之后都收不到来自 k' 的消息，但会收到其他代理的所有消息。因此，在 r'' 中，i、j 和 k 在第 m 轮和第 $m+1$ 轮都能收到除 k' 之外所有代理的消息。由于 $j \in (\mathrm{N} \wedge \mathrm{O})(r'', m') \cup (\mathrm{N} \wedge \mathrm{O})(r, m')$，$(r'', m')$ 可从 (r, m) 经 $(\mathrm{N} \wedge \mathrm{O})$-⊡-到达。

由于在 r'' 中，j 在第 m 轮之后收不到来自 k' 的消息，因此它无法发现 k 已发生故障。因此，存在一次运行 r'''，除代理 k 不表现出无故障行为外，它与 r' 完全相同。因此，$\mathrm{I}, (r''', m') \models inv_1(i, j, k)$。此外，$j \in (\mathrm{N} \wedge \mathrm{O})(r''', m') \cup (\mathrm{N} \wedge \mathrm{O})(r, m')$，因此 (r''', m') 可从 (r, m') 经 $(\mathrm{N} \wedge \mathrm{O})$-⊡-到达。

□

最后，我们构造弱安全性所需的点 $(r_i^{0'}, m)$ 和 $(r^{1'}, m)$。这一情形的论证比前一种情形更简单。本情形的大部分工作由引理 A.17 和 A.18 完成，它们分别对应于引理 A.15 和 A.16。

引理 A.17。如果 $\mathcal{I}, (r, m) \models inv_0(i, j, k)$，则存在一个运行 $\hat{r}$，使得 $(\hat{r}, 0)$ 可从 (r, m) 经过 $(N \wedge \mathrm{Z})$-⊡-可达，并且 $\mathcal{I}, (\hat{r}, 0) \models \neg\exists 1$。

证明。我们对 m 使用归纳法。

如果 $m = 0$，那么由于代理 j 在第 1 轮次作出决定 0，因此必有 $\mathrm{I}, (r, 0) \models \mathrm{init}_j = 0$。因此，代理 j 认为一个每个代理的初始偏好都是 0 的运行 $\hat{r}$ 是可能的。显然，$(\hat{r}, 0)$ 可从 $(\mathrm{N} \wedge \mathrm{Z})$-⊡-可达于 $(r, 0)$。如果 $m > 1$，由于代理 k 是故障的，但在 r 中没有表现出故障行为，$\mathrm{I}, (r, m) \models \neg C_{\mathcal{N}}$（t 个故障）。根据引理 A.5，代理 j 必须至少收到一条长度为 m 的 0-链，才能作出决定 0。令 k' 为向 j 发送 0-链的某个代理。注意，k' 不可能是 i 或 j，因为 i 和 j 在第 $m+1$ 轮次之前不会作出决定。令 r' 为一个与 r 相同的运行，但其中 i 是故障的而非 k，k 从 k' 处获知信息，并且 k 在 r' 的第 m 轮次没有从 i 处获知信息。由于 $j \in (\mathrm{N} \wedge \mathrm{Z})(r, m) \cap (\mathrm{N} \wedge \mathrm{Z})(r' m)$，且 j 在这两个点的局部状态相同，因此 $(r' m)$ 可从 (r, m) 经过 $(\mathrm{N} \wedge \mathrm{Z})$-⊡-可达。

In r', agent k is nonfaulty and receives a 0-chain from k' in round m. Let k^* be an agent that sent a 0-chain to k' in round $m-1$. Consider a run r'' that is identical to r' except that agent i receives a 0-chain in round $m-1$ from k^* and agent j receives a message from agent i in round m. (If $m = 1$, then we take r'' to be a run where i has initial preference 0 and j receives a message from i in round 1.) Agent i decides 0 in round m of r''. Since k has the same local state in both $(r' m)$ and $(r'' m)$, and $k \in (\mathcal{N} \wedge \mathcal{Z})(r' m) \cap (\mathcal{N} \wedge \mathcal{Z})(r'' m)$, $(r'' m)$ is $(\mathcal{N} \wedge \mathcal{Z})$-⊡-reachable from $(r' m)$. Because j received a message from i in round m of r'' and j is nonfaulty, j decides 0 in round $m+1$ of r''.

设 r''' 是一次运行，除代理 i 无故障外，它与 r'' 完全相同。于是，j 在 $(r'' m)$ 和 $(r''' m)$ 中具有相同的局部状态，并且 $j \in (\mathrm{N} \wedge \mathrm{Z})(r'' m) \cap (\mathrm{N} \wedge \mathrm{Z})(r''' m)$。因此，$(r''' m)$ 可从 (r, m) 经 $(\mathrm{N} \wedge \mathrm{Z})$-⊡-可达；在 r''' 中，代理 i 在第 m 轮作出决定 0，$i \in \mathrm{N}$，$j \in \mathrm{N} k \notin \mathrm{N}$，并且 k 在整个运行 r''' 中均无故障。此外，$\mathrm{I}, (r''' m-1) \models \mathrm{inv}_0(j, i, k)$。根据归纳假设，可知存在一次运行 $\hat{r}$，使得 $(\hat{r}, 0)$ 可从 $(r''' m-1)$ 经 $(\mathrm{N} \wedge \mathrm{Z})$-⊡-可达，且 $\mathrm{I}, (\hat{r}, 0) \models \neg\exists 1$。根据 $(\mathrm{N} \wedge \mathrm{Z})$-⊡-可达关系的传递性，$(\hat{r}, 0)$ 也可从 (r, m) 经 $(\mathrm{N} \wedge \mathrm{Z})$-⊡-可达，因此命题成立。□

Lemma A.18. *If $\mathcal{I}, (r, m) \models (i \in \mathcal{N} \wedge \bigcirc(decided_i = \perp))$, then i considers a point $(r' m)$ possible at (r, m) where $\mathcal{I}, (r' m) \models inv_0(i, j, k)$.*

Proof. Suppose that $\mathcal{I}, (r, m) \models i \in \mathcal{N} \wedge \bigcirc(decided_i = \perp)$. We must have $\mathcal{I}, (r, m) \models \neg C_{\mathcal{N}}(t\text{-}faulty))$, since otherwise, by Lemma A.4, we get a contradiction. We next show that, whether

or not i knows which agents are faulty, i considers a point $(r' m)$ possible at (r, m) from which a point $(r''' m)$ is $(\mathcal{N} \wedge \mathcal{O})$-⊡-reachable from (r, m) such that $\mathcal{I}, (r''' m) \models inv_0(i, j, k)$.

- If $\mathcal{I}, (r, m) \models \neg K_i(t\text{-}faulty)$ then, by definition, there exists a $k \in Agt$ such that $\mathcal{I}, (r, m) \models \neg K_i \neg (k \in \mathcal{N}) \wedge \neg K_i \neg (k \notin \mathcal{N})$. Let r' be a run identical to r except that $k \notin \mathcal{N}$ and k does not exhibit faulty behavior in r'. We claim that $(r, m) \sim_i (r' m)$. Suppose by way of contradiction that i has different local states in (r, m) and $(r' m)$. Since the only difference between these runs are the blocked messages sent by k that are no longer blocked, i must have heard from an agent (either directly or indirectly) that did not receive a message from k in r but did in r'. But in that case, i would have known that k is faulty in r, which is a contradiction.
由于 i 在 $(r'm)$ 中尚未作出决定，并且在第 m 轮不会作出决定（因为 i 在 (r, m) 和 $(r' m)$ 中具有相同的局部状态），因此 $\mathrm{I}, (r'm) \models \neg K_i(\bigwedge_{j \in \mathrm{Agt}} \neg(\text{正在作出决定}_j = 0))$。也就是说，$i$ 认为可能存在一个代理 j' 在第 $m+1$ 轮决定 0。令 r'' 为与 r' 完全相同的运行，唯一的区别是 j' 在 $m+1''$ 的第 r 轮决定 0。根据引理 A.5，如果 $m \geq 1$，j' 在 r'' 的第 m 轮从某个代理 k' 收到一条 0-链，并且如果 $m = 0$，$\mathrm{init}_{j'} = 0$。令 r''' 为与 r'' 完全相同的运行，唯一的区别是代理 j（其也无故障）在 $m \geq 1$ 时于第 m 轮收到来自 k' 的消息，并在 $m = 0$ 时具有初始偏好 0。因此，代理 j 在 r''' 的第 $m+1$ 轮决定 0。由知识关系的传递性可得 $(r, m) \sim_i (r''' m)$。

- 如果 $\mathrm{I}, (r, m) \models K_i$（$t$-故障），我们利用如下观察：$\mathrm{I}, (r, m) \models \neg C_{\mathcal{N}}$（$t$-故障）。与引理 A.16 的证明一样，我们必须有 $\mathrm{I}, (r, m-1) \models \neg K_i \neg (k \in \mathrm{N}))$，其中 k 是某个非故障代理。

  代理 i 必定是在第 m 轮次得知 k 是故障的，这可能是因为它首次未收到来自 k 的消息，也可能是因为它收到了某个已知晓 k 故障的故障代理 k' 发来的消息。（代理 k' 不可能是非故障代理，因为在时间 $m-1$，没有任何非故障代理知道 k 是故障的。）由于代理 i 尚未作出决定，且在 r 的第 $m+1$ 轮次没有作出决定，因此代理 i 知道其他非故障代理没有在更早的轮次作出决定。此外，在 (r, m) 中，i 认为存在一个可能的点 $(r_{\prime}, m)$，其中 $\mathrm{I}, (r_{\prime} m) \models (\text{作出决定}_{j'} = 0)$ 对于某个代理 j' 成立，因为 i 在 $r_{\prime}$ 的第 $m+1$ 轮次没有作出决定 1。

  如果代理 i 在第 m 轮中由于未收到来自 k 的消息而得知 k 是故障的，那么在点 $(r_{\prime\prime} m)$ 处，代理 i 会认为 $(r_{\prime\prime} m)$ 是可能的，其中 $r_{\prime\prime}$ 满足以下条件：某个无故障代理 j 在第 m 轮中收到了来自 k 的消息，并且 k 在第 $m-1$ 轮中作出决定 0（这是由于它收到了来自某个代理的消息，而该代理向在 0-链中向 j' 发送消息的代理发送了消息）。在 r'' 中，j 在第 m 轮收到了一个 0-链，因此在第 $m+1$ 轮中作出决定 0。由此可得 $\mathrm{I}, (r_{\prime\prime} m) \models inv_0(i, j, k)$。

  如果代理 i 因从某个故障代理 k' 处听说代理 k 发生故障而得知 k 发生故障，那么代理 i 会认为点 $(r_{\prime\prime} m)$ 在 $(r_{\prime} m)$ 处是可能的，其中 $r_{\prime\prime}$ 与 $r_{\prime}$ 完全相同，唯一的区别在于，在 $r_{\prime\prime}$ 中，代理 j 在第 m 轮及之后的轮次中都没有收到来自代理 k' 的消息，并且在第 m 轮收到了来自代理 $j_{\prime}$ 的一个 0-链。如果 $k \neq k'$，那么我们再次有 $\mathrm{I}, (r_{\prime\prime} m) \models inv_0(i, j, k)$。
  如果 $k = k'$，那么 $\mathrm{I}, (r'' m) \not\models \mathrm{inv}_0(i, j, k)$，因为在运行 r'' 期间，k 并非始终表现为无故障。在这种情况下，考虑一个运行 r^*，它与 r'' 完全相同，只是在 r^* 的第 m 轮中，代理 k 向除 j 之外的所有代理发送一条消息。由于 $j \in (\mathrm{N} \wedge \mathrm{Z})(r'' m) \cap (\mathrm{N} \wedge \mathrm{Z})(r^*, m)$，且 j 在这两个点具有相同的局部状态，因此 (r^*, m) 可从 $(r'' m)$ 经 $(\mathrm{N} \wedge \mathrm{Z})$-⊡ 可达。令 r''' 为与 r^* 完全相同的运行，只是代理 k 在 r''' 中没有表现出故障行为，且 i 从 j' 接收到一个 0-链。根据构造，$\mathrm{I}, (r''' m) \models \mathrm{inv}_0(i, j, k)$。由于代理 i 在 r^* 中并不知道 k 发生了故障，因此 $i \in (\mathrm{N} \wedge \mathrm{Z})(r^*, m) \cap (\mathrm{N} \wedge \mathrm{Z})(r''' m)$，且 $(r''' m)$ 可从 (r^*, m) 经 $(\mathrm{N} \wedge \mathrm{Z})$-⊡ 可达。同样，这足以得到所需结果。

□

这就完成了对$\mathbf{P}^1$弱安全性的证明，并得到以下结果：定理 7.7。$\mathbf{P}^1$相对于$\gamma_{fip、n、t}$是弱安全的。

A.2.7 实现。现在，我们使用与 Moses 和 Tuttle [12] 所用类似的通信图，明确地定义全信息上下文。直观地说，代理i的通信图是一个带标记的图，它简洁地描述了代理 i发送和接收的所有消息。形式上，通信图 $G_{i,m}$ 定义如下。顶点集合 $V(G_{i,m})$由所有形如 (j, m') 的点组成，其中 $m' \leq m$ 取遍所有值，代理 j 取遍所有代理；边集合 $E(G_{i,m})$由从 $(j, m'-1)$ 到 $(j' m')$ 的所有边组成，其中 j, j' 属于$\in \mathsf{Agt}$，且 $m' \leq m$；每条边都有一个消息标记 $l_{G_{i,m}} \in \{0, 1, ?\}$（见下文），每个 $j \in \mathsf{Agt}$ 都有一个初始偏好标记 $p_{G_{i,m}} \in \{0, 1, ?\}$（可将其视为形如 $(j, 0)$ 的顶点上的标记）。从 $(j, m'-1)$ 到 $(j'm')$ 的边，如果 i知道 j在第 m' 轮向 j' 发送了消息，则标记为 1；如果 i知道 j在第 m' 轮没有向 j' 发送消息，则标记为 0；如果 i不知道 j在第 m' 轮是否向 j' 发送了消息，则标记为 ?。请注意，在全信息协议中，如果 i知道代理的初始偏好，以及哪些代理为 $m'' < m'$发送了第 m'' 轮的消息，那么 i就很容易推断出所发送消息的内容。$(j, 0)$ 上取值为 $v \in \{0, 1\}$ 的偏好标记表示 i知道 j 的初始偏好是 v；而标记 ? 表示 i不知道 j 的初始偏好。我们用 $G_{i,m}(r)$ 表示全信息交换中一次运行 r 的时间 m 时代理 i 的通信图，并用 $\mathbf{G}_{i,m}$ 表示代理 i 在时间 m 的所有通信图组成的集合。

设 $\mathbf{E}^{fip}(n)$ 为适用于 n 个代理的信息交换协议，其中对每个代理 i，以下条件均成立：

- 局部状态的形式为 $(时间_i, 已决定_i, 初始_i, G_{i,\ 时间_i})$，其中$G_{i,\ 时间_i}$是一个通信图。
- 每个代理 i的初始局部状态具有形式 $(0, \bot, 初始化_i, G_{i,0})$，其中 $G_{i,0} \in \mathbf{G}_{i,0}$。（注意，在 $G_{i,0}$ 中，对于所有代理 $j \neq i$，我们必须有 $p_{G_{i,0}}(j) = ?$，并且 $p_{G_{i,0}}(i) = 初始化_i$。）
- $M_i = \mathcal{G}_{i,时间_i}$。
- 对于所有代理 j和动作 a，$\mu_{ij}(时间_i, 已决定_i, 初始化_i, G_{i,时间_i}\ a) = G_{i,时间_i}$.
- $\delta_i(时间_i,\ 已决定_i,\ 初始化_i, G_{i,m}\ a, (m_1, \ldots, m_n)) = (时间_i + 1,\ 已决定'_i,\ 初始化_i, G_{i,时间_{i+1}},$ 其中$G_{i,时间_{i+1}} \in \mathbf{G}_i$，时间$_{i+1}$是通过为轮次时间$_i$+1 添加顶点和边，并合并 i和G_i在时间$_i$收到的所有图中的标签而得到的。更确切地说，如果 RG_i 由 i 截至并包括轮次时间$_i$ + 1所收到的所有图组成，那么

$$l_{G_{i,time_i+1}}((j,m),(j',m+1)) = \begin{cases} v & \text{if } \exists G \in RG_i(l_G((j,m),(j',m+1)) = v) \wedge v \in \{0,1\}, \\ 1 & \text{if } m = time_i \wedge j' = i \wedge m_j \neq \bot, \\ 0 & \text{if } m = time_i \wedge j' = i \wedge m_j = \bot, \\ ? & \text{otherwise.} \end{cases}$$

$$p_{G_{i,time_i+1}}(j) = \begin{cases} v & \text{if } \exists G \in S, v \in \{0,1\}((p_G(j) = v)), \\ ? & \text{otherwise.} \end{cases}$$

Finally, $decided'_i$ is determined by the action a, just as in the standard EBA context.

Let $\gamma_{fip,n,t} = (\mathcal{E}_{fip}(n)\ SO(t)\ \pi_{fip,n})$, where $\pi_{fip,n}$ interprets the standard propositions in the standard way. To check the knowledge conditions in $\mathbf{P}^1$, it is useful to define the following sets, which can be computed in polynomial time from the communication graph:

- For $m' < m$, $f(j, m'\ G_{i,m})$ is the set of faulty agents that i knows that j knows about at time m', given $G_{i,m}$. The set $f(j, m'\ G_{i,m})$ is the union of (a) $f(j'\ m'-1, G_{i,m})$ for all j' that sent a message to j in round m' in $G_{i,m}$ if $m' > 0$, (b) $\{j'\}$ for each agent j' that did

not send a message to j in round m' in $G_{i,m}$, and (c) $f(j, m'-1, G_{i,m})$ if $m' > 0$. (Note that $f(j, 0, G_{i,m}) = \emptyset$.)

- 对于 $m' < m$, $D(S, m', G_{i,m})$ 是故障代理的集合；在给定 $G_{i,m}$ 的情况下，i知道这些代理在时间 m' 知道 S 中的代理所知的内容。$D(S, m', G_{i,m}) = \cup_{k \in S} f(k, m', G_{i,m})$。

除了 f和 D之外，代理 i 还可以在时间 $m' < m$ 计算每个代理 j 的动作，条件是 $(j, m') \rightarrow_r (i, m)$，其中 r 是当前运行，因为我们使用的是全信息协议。令$d(j, m', G_{i,m}) \in \{0, 1, \perp, ?\}$ 表示 i 所知的代理 j 在第 $m'+1$ 轮的动作。如果$d(j, m', G_{i,m}) = v \in \{0, 1\}$，则 i 知道 j 在第 $m'+1$ 轮作出决定 v；如果 $d(j, m', G_{i,m}) = \perp$，则 i 知道 j 在第 $m'+1$ 轮没有作出决定；最后，如果 $d(j, m', G_{i,m}) = ?$，则 $(j, m') \not\rightarrow_r (i, m)$。

每个直接或间接向 i 发送过消息的代理所知的值集合，也可以根据通信图在多项式时间内计算。令 $V(j, m', G_{i,m})$ 表示在时间 m'，如果根据 $G_{i,m}$，i知道 j知道什么，则 j 知道的值集合；否则为 $\emptyset$。

接下来，我们定义布尔值族 $common_v(i, m, G_{i,m})$ 和 $cond_1(i, m, G_{i,m})$，它们也可以在多项式时间内计算。

定义 A.19（$common_v$和 $cond_v$）。直观而言，若在给定 $G_{i,m}$ 的情况下，时间 m 满足 $K_i(\text{dist}_N(\text{t 个故障}) \wedge (\bigwedge_{j \in N} (K_j \text{no-decided}_j(v))) \wedge (\bigvee_{j \in N} K_j(\exists v)))$（这意味着时间 $m+1$ 满足 $C_N(\text{t 个故障} \wedge \text{no-decided}_N(1) \wedge \exists v)$），则 $common_v$ 成立。我们按如下方式计算 $common_v$。如果以下三个条件中的任意一个成立，则 $common_0(i, m, G_{i,m}) =$为假：

- $|D(f(i, m, G_{i,m}) m-1, G_{i,m})| \neq t$，其中 $f(j, m', G_{i,m}) =$代理 $-f(j, m', G_{i,m})$（这些是 i 在时间 m'，根据 $G_{i,m}$ 认为可能无故障的代理）；
- 存在一个代理 $j \notin f(i, m, G_{i,m})$，使得 $d(j, m', G_{i,m}) = 1 - v$对于某个 $m' < m$。
- 对于所有代理 $j \notin D(f(i, m, G_{i,m}) m-1, G_{i,m})$，$v \notin V(j, m-1, G_{i,m})$

否则，通用$_v(i, m, G_{i,m}) = true$.

将 $common_v(i, m, G) =$设为 false 的第一个条件，对应于代理 i认为，在时间 m，可能无故障的代理们并不具备关于故障代理的分布式知识 t。如果是这样，那么实际上无故障的代理们在时间 $m-1$当然也不会具备关于故障代理身份的分布式知识，因此，无故障代理之间不会对故障代理的身份形成共同知识 m——参见引理 5。如果某个代理 j认为在时间 m可能无故障的某个代理已经作出决定 $1-v$，则第二个条件成立。我们可以假设 $|D(f(i, m, G_{i,m}) m-1, G_{i,m})| = t$（否则，根据第一个条件，$common_v(i, m, G) =$为 false）。因此，$i$在时间 m知道哪些代理是无故障的，所以 i知道某个无故障代理已经作出决定 $1-v$，因此，无故障代理之间不可能形成“没有任何无故障代理作出决定 $1-v$”这一共同知识。

The Boolean $cond_0 = true$ holds if the formula $init_i = 0 \vee \bigvee_{j \in Agt} K_i(jdecided_j = 0)$ holds at time m, given $G_{i,m}$. Formally,

- $cond_0(i, 0, G_{i,0}) = (init_i = 0)$.
- For $m > 0$, $cond_0(i, m, G_{i,m}) = true$ if there exists an agent j such that $d(j, m-1, G_{i,m}) = 0$ and $l_{G_{i,m}}((j, m-1), (i, m)) = 1$; otherwise, $cond_0(i, m, G_{i,m}) = false$.

The Boolean $cond_1(i, m, G_{i,m}) = true$ holds if $K_i(\bigwedge_{j \in Agt} \neg(deciding_j = 0))$ holds at time m, given $G_{i,m}$. Formally,

- $cond_1(i, 0, G_{i,0}) = false$.
- For $m > 0$, let m' be the latest time such that $d(j, m' G_{i,m}) = 0$ for some agent j (as usual, $m' = -1$ if $d(j, m' G_{i,m}) \neq 0$ for all agents j), and let m_j be the latest time that $(j, m_j) \rightarrow_r (i, m)$, where r is a run for which $G_{i,m}$ describes i's view at time m in r. (There are many

such runs; it does not matter which one we choose, since they all agree on the whether $(j, m_j) \rightarrow_r (i, m)$.) Intuitively, $m' = len_j(r, m)$. If, for all m'' with $m' < m'' \leq m$, there exist at least $m'' - m'$ agents j such that $d(j, m'', G_{i,m}) = ?$, then $cond_1(i, m, G_{i,0}) = true$; otherwise $cond_1(i, m, G_{i,0}) = false$.

Using these definitions, we can define an implementation of $\mathbf{P}^1$ in the full-information context. Let P^{opt} be the EBA decision protocol implemented by the following program:

Program: P_i^{opt}

if $decided_i \neq \perp$ **then** noop
else if $common_0(i, time_i - 1, G_{i,time_i})$ **then** $\mathtt{decide}_i(0)$
else if $common_1(i, time_i - 1, G_{i,time_i})$ **then** $\mathtt{decide}_i(1)$
否则如果$cond_0(i, \text{时间}_i, G_{i,time_i})$**则**作出决定$_i(0)$
否则如果 条件$_1(i, \text{时间}_i, G_{i,time_i})$**则** 作出决定$_i(1)$
否则 空操作

以下引理表明，在共同$_v$的定义中所检查的初始条件对应于检查 C_N（t-故障）。

引理 *A.20*。 $|f(i, m, G_{i,m}(r))| = |D(f(i, m, G_{i,m}(r))\ m-1, G_{i,m}(r))| = t$对于某个代理 i，当且仅当 $I_{\gamma_{fip},n,t},P^1,(r, m) \models C_N(t$ 个故障$)$。

证明。令 $\mathrm{I} = \mathrm{I}_{\gamma_{fip},n,t},\mathrm{P}^1$和 $G = G_{i,m}(r)$。假设 $|f(i, m, G)| = |D(f(i, m, G)\ m-1, G)| = t$对于某个代理 i成立。我们首先注意到 $\mathrm{Agt} - f(i, m, G) = \mathrm{N}$，因为存在 t故障代理。这意味着集合 $D(f(i, m, G)\ m-1, G)$ 是无故障代理在 $m-1$时间所知的所有故障代理的集合。由于 $|D(f(i, m, G)\ m-1, G)| = t$，因此对于所有故障代理 j，必然存在一个无故障代理知道代理 j发生故障。因此，$\mathrm{I}, (r, m) \models \ominus \mathrm{dist}_N(\mathrm{t}$ 个故障$)$。根据命题 A.2，我们有 $\mathrm{I}, (r, m) \models C_N(\mathrm{t}$ 个故障$)$。

反之，假设 $\mathrm{I}, (r, m) \models C_N(t$-故障$)$。同样，根据命题 A.2，我们得到 $\mathrm{I}, (r, m) \models \ominus dist_N(t$-故障$)$。根据定义，无故障代理在时间 $m-1$已知的所有故障代理的并集就是所有故障代理的集合。因此，$|D(\mathrm{N}\ m-1, G)| = t$。所以，$\mathrm{N} = f(i, m, G)$。□

定理 A.21. 如果 $n - t \geq 2$，则 P^{opt} 在完整信息 *EBA* 上下文 $\gamma_{fip,n,t}$中实现 $\mathbf{P}^1$。

证明。令 $\mathrm{I} = \mathrm{I}_{\gamma_{fip,n,t},\mathrm{P}^1}$以及 $G = G_{i,m}(r)$。我们证明，对于所有点 (r, m)，$P_i^{opt}(r_i(m)) = (\mathbf{P}_i^1)^{\mathcal{I}}(r_i(m))$。

- 如果 $P_{opt\,i}(r_i(m)) =$由于已决定$_i \neq \perp$而执行 $r_i(m)$ 时为空操作，那么显然我们也有 $(\mathbf{P}_i^1)_{\mathcal{I}}(r_i(m)) =$为空操作，因为 $\mathrm{I}, (r, m) \models K_i($已决定$_i \neq \perp)$。
- 如果 $P\mathrm{opt}_i(r_i(m)) =$作出决定$_i(0)$是根据第二行作出的，那么我们必然已在 $r_i(m)$ 中作出决定$_i = \perp$，且$m > 0$。根据 common0 的定义，我们还必然有 (a) $|D(f(i, m-1, G)\ m, G)| = t$，(b) 对所有 $j \notin f(i, m, G)$ 和 $m' < m$，$d(j, m' G) \neq 1$，以及 (c) 对某个$j \notin f(i, m, G)$，$0 \in V(j, m-1, G)$。由 (a) 可知，利用引理 A.20，$\mathrm{I}, (r, m) \models C\mathrm{N}$ (t 个故障)。由 (b) 可知，在任何轮次 $m' < m+1$，都没有非故障代理作出决定 1。因此，$\mathrm{I}, (r, m) \models$no-decidedN (1)。最后，(c) 意味着 i知道某个非故障代理 j在时间 $m-1$的初始偏好为 0；也就是说，$\mathrm{I}, (r, m) \models \ominus K_j(\exists 0)$。结合这些观察结果，并利用命题 A.2，我们可以得出结论：$\mathrm{I}, (r, m) \models C\mathrm{N}$ (t 个故障 $\wedge$no-decided N (1) $\wedge \exists 0$)。根据引理 A.3，$\mathrm{I}, (r, m) \models K_i(C\mathrm{N}$ (t 个故障 $\wedge$no-decided N (1) $\wedge \exists 0))$，因此 $(\mathrm{P}_i^1)\mathrm{I}\,(r_i(m)) =$作出决定$_i(0)$。

- 如果 $P_{opt\,i}(r_i(m)) =$作出决定$_i(1)$ 是根据第三行作出的，那么我们必然已在 $r_i(m)$ 中作出决定已决定$_i = \perp$，$common_0(i, m, G) = false$, $common_1(i, m, G) = true$, and $m > 0$. 如前所述，$common_1(i, m, G) = true$ 意味着 $I, (r, m) \models C_{\mathcal{N}}$(t-faulty) $\wedge$ no-decided$_{\mathcal{N}}(0) \wedge \ominus K_j(\exists 1)$，因此 $I, (r, m) \models K_i(C_{\mathcal{N}}$(t-faulty $\wedge$ no-decided$_{\mathcal{N}}(0) \wedge \exists 1))$。因此，只需证明 $I, (r, m) \models \neg C_{\mathcal{N}}$(t-faulty$\wedge$no-decided$_{\mathcal{N}}(1)\wedge\exists 0)$。由于 $common_0(i, m, G) =$ false，因此必有以下情况之一：(a) $|D(f(i, m-1, G)\ m-1, G)| \neq t$；(b) $|D(f(i, m, G)\ m-1, G)| = t$，并且存在某个 $j \notin f(i, m, G)$ 和 $m' < m, d(j, m', G) = 1$；或者 (c) $|D(f(i, m-1, G)\ m-1, G)| = t$，且对于所有 $j \notin f(i, m, G)$，都有 $0 \notin V(j, m-1, G)$。若 (a) 成立，则由引理 A.20 可得 I, $(r, m) \models \neg C_{\mathcal{N}}$(t-faulty)。若 (b) 成立，则如我们所观察到的，I, $(r, m) \models \neg C_{\mathcal{N}}$ (t-faulty)，此外，对某个非故障 j 和 $m' < m$，有 $d(j, m', G) = 1$。因此，I, $(r, m) \models \neg$ no-decided$_{\mathcal{N}}(1)$。若 (c) 成立，则我们再次有 I, $(r, m) \models \neg$no-decided N (1)；此外，对于所有非故障 j，都有 $0 \notin V(j, m-1, G)$。因此，i 认为根据给定的 G，在时间 $m-01$ 没有任何非故障代理知道一个 0 是有可能的，所以由命题 A.2，I, $(r, m) \models \neg C_{\mathcal{N}}(\exists 0)$。因此，在所有情况下，I, $(r, m) \models \neg C_{\mathcal{N}}$(t-faulty $\wedge$no-decided$_{\mathcal{N}}(1) \wedge \exists 0)$，所以 $(P_i^1)^{\mathcal{I}}(r_i(m)) =$ 对 (1) 作出决定。

- 如果 $P_i^{pt}(r_i(m)) =$作出决定$_i(0)$ 根据第四行，我们必须已决定$_i = \perp$在 $r_i(m)$，common$_0(i, m, G) =$ common$_1(i, m, G) =$为假，且 cond$_0(i, m, G) =$为真。同样，common$_0(i,m, G) =$为假意味着 I, $(r, m) \models \neg C_{\mathcal{N}}$(t 个故障 $\wedge$no-decided$_{\mathcal{N}}(1) \wedge \exists 0)$。类似地，common$_1(i, m, G) =$ 为假意味着 I, $(r, m) \models \neg C_{\mathcal{N}}$(t 个故障 $\wedge$no-decided$_{\mathcal{N}}(0) \wedge \exists 1)$。因此，我们只需证明 I, $(r, m) \models$init$_i = 0 \vee \bigvee_{j\in}$Agt K_i(j 已决定 $_j = 0)$。我们对 m 使用归纳法。对于基础情形，如果 cond$_0(i, 0, G) =$为真，根据定义，必有 init$_i = 0$。对于归纳步骤，假设对于某个 $m > 0$，cond$_0(i, m, G) =$为真。根据定义，这意味着 $d(j, m-1, G) = 0$且 $l_G((j, m-1), (i, m)) = 1$，其中某个代理 j。因此，$(P_i^1)^{\mathcal{I}}(r_i(m)) =$作出决定$_i(0)$。于是代理 j 作出决定 0，且代理 i 在轮次 m 从代理 j 收到消息。由此可得 I, $(r, m) \models K_i$(j已决定 $_j = 0)$，因此 $(P_i^1)^{\mathcal{I}}(r_i(m)) =$作出决定$_i(0)$。

- 如果到第五行时 Popt $_i(r_i(m)) =$作出决定$_i(1)$，那么我们必定已在 $r_i(m)$ 中作出决定$_i = \perp$，common0$(i,m, G) =$common1$(i, m, G) =$cond0$(i, m, G) =$为假，而 cond1$(i, m, G) =$为真。我们还知道$m > 0$，因为 cond1$(i, m, G) =$为真。和之前一样，common0$(i, m, G) =$common1$(i, m, G) =$为假，这意味着共同知识条件不成立。因此，我们需要证明 I, $(r, m) \models \neg$(init$_i = 0 \vee \bigvee_{j\in}$ Agt K_i(jdecided $_j = 0)$)，以及 I, $(r, m) \models K_i(\bigwedge_{j\in}$Agt $\neg$(deciding$_j = 0))$。根据定义，cond0$(i, m, G) =$为假意味着对所有 $j \in$Agt，$d(j, m-1, G) \neq 0$或 $l_G((j, m-1), (i, m)) \neq 1$，因此要么 j在第 m轮没有决定 0，要么 i在第 m轮没有收到来自代理 j的消息。无论哪种情况，I, $(r, m) \models \neg K_i$ (jdecided $_j = 0)$，因此 I, $(r, m) \models \neg$(init$_i = 0 \vee \bigvee_{j\in}$Agt K_i(jdecided $_j = 0)$)。最后，如果 cond1$(i, m, G) =$为真，根据命题 A.7.，我们可以得出 I, $(r, m) \models K_i(\bigwedge_{j\in}$Agt $\neg$(deciding$_j = 0))$，因此 (P_i^1)I $(r_i(m)) =$作出决定$_i(1)$。

- 如果根据最后一行，Popt $_i(r_i(m)) =$为空操作，那么我们必定已在 $r_i(m)$ 中作出决定$_i = \perp$，并且 common0$(i,m, G) =$common1$(i, m, G) =$cond0$(i, m, G) =$cond1$(i, m, G) =$为假。若 $m = 0$，则除了 I, $(r, m) \models$init$_i = 0 \vee \bigvee_{j\in}$Agt K_i(jdecided $_j = 0)$ 之外，P^1中的条件都不可能成立。不过，由于 cond0$(i, 0, G) =$为假，我们必定有 init$_i \neq 0$，并且 (P_i^1)I $(r_i(0)) =$为空操作。若$m > 0$，那么上述论证表明共同知识条件不成立，且 I, $(r, m) \models \neg$(init$_i = 0 \vee \bigvee_{j\in}$Agt K_i(jdecided $_j = 0)$)。因此，我们只需证明 I, $(r, m) \models \neg K_i(\bigwedge_{j\in}$Agt $\neg$(deciding$_j = 0))$。由于共同知识条件不成立，我们可以应用命题 A.7 得出结论：该命题成立。因此，P^1中的条件均不成立，且我们有 (P_i^1)I $(r_i(m)) =$为空操作。

□

由此我们可以得出结论：就全信息交换而言， P^{opt} 也是最优的。由于 P^{opt} 中的每个条件都可以在关于通信图大小的多项式时间内检查，而通信图本身使用 $O(n^2t)$ 位，因此我们得到以下结果：

Proposition 7.9. *There exists a polynomial-time implementation P^{opt} of $\mathbf{P}^1$ with respect to a full-information exchange.*

A.3 Proof for Section 8

Proposition 8.2. *If r is a failure-free run, then*

(a) If there is at least one agent with an initial preference of 0 in r, then all agents decide by round 2.
(b) If all agents have an initial preference of 1, then all agents decide by round $t + 2$ with P^{min} and by round 2 with P^{basic} and P^{fip}.

Proof. For the first part, suppose that some nonfaulty agent has an initial preference of 0. Clearly that agent decides 0 in the first round and tells all the other agents, who decide in the second round (for all three protocols).

For the second part, suppose that all the agents are nonfaulty and have an initial preference of 1. Then with P^{min}, since no agent will decide 0 or hear about a decision of 0, the agents will wait for $t + 1$ rounds of information exchange decide 1 in round $t + 2$. With P^{basic} and P^{fip}, no agent will decide right away and since all agents i will get a message from every other agent j in the first round from which they can conclude that j's initial preference was 1 ($(init, 1)$ in the case of P^{basic} and an explicit message saying that j's initial preference was 1 in the case of P^{fip}), agents can all decide on 1 in round 2. □

ACKNOWLEDGMENTS

Alpturer and Halpern were supported in part by AFOSR grant FA23862114029. Halpern was additionally supported in part by ARO grants W911NF-19-1-0217 and W911NF-22-1-0061. The Commonwealth of Australia (represented by the Defence Science and Technology Group) supported this research through a Defence Science Partnerships agreement. We thank Yoram Moses for useful comments on the paper.

REFERENCES

[1] A. Castañeda、Y. A. Gonczorowski 和 Y. Moses。2014。《无懈可击的共识》。载于第 *28* 届分布式计算国际会议论文集（*DISC '14*）。91–106。[2] A. Castañeda、Y. Moses、M. Raynal 和 M. Roy。2017。《同步共识中的提前决定与停止：基于谓词的导览》。载于第 *5* 届网络系统国际会议论文集（*NETYS 2017*）。206–221。[3] B. Coan。1986。《容错分布式协议的一种通信高效规范形式》。载于第 *5* 届 *ACM* 分布式计算原理研讨会论文集。63–72。[4] C. Dwork 和 Y. Moses。1990。《拜占庭环境中的知识与共同知识：崩溃故障》。《style id='11'>信息与计算 88，2（1990），156–186。[5] R. Fagin、J. Y. Halpern、Y. Moses 和 M. Y. Vardi。1995。《关于知识的推理》。《style id='14'>Reasoning About Knowledge。麻省理工学院出版社，马萨诸塞州剑桥。略作修订的平装版于 2003 年出版。。[6] J. Y. Halpern 和 R. Fagin。1985。《分布式系统中知识、动作与通信的形式模型：初步报告》。载于第 *4* 届 *ACM* 分布式计算原理研讨会论文集。224–236。[7] J. Y. Halpern 和 Y. Moses。1990。《分布式环境中的知识与共同知识》。《style id='20'>J. ACM 37，3（1990），549–587。[8] J. Y. Halpern、Y. Moses 和 O. Waarts。2001。《最终拜占庭一致性的一种刻画》。《style id='23'>SIAM 计算杂志 31，3（2001），838–865。https://doi.org/10.1137/S0097539798340217 本文的初步版本发表于第 *9* 届 *ACM* 分布式计算原理研讨会论文集，1990 年，第 333–346 页。。[9] X. Huang 和 R. van der Meyden。2013。《具有同步语义的基于知识的程序实现的符号综合》。载于理性与知识的理论方面：第十四届会议论文集（*TARK 2013*）。

[10] X. Huang 和 R. van der Meyden。2014。《具有观察语义的认知规范的符号综合》。载于系统构造与分析工具及算法第 *20* 届国际会议论文集（*TACAS 2014*）。455–469。[11] L. Lamport。1978。《分布式系统中的时间、时钟与事件排序》。《style id='5'>《ACM 通讯》 21，7（1978），558–565。[12] Y. Moses 和 M. R. Tuttle。1988。《利用共同知识编写同时动作程序》。《style id='8'>Algorithmica 3（1988），
121 169 10 1007 01762112 –。https://doi.org/ 。/BF G. Neiger 和 M. R. Tuttle。。《共同知
[13] 1993
识与一致的同时协调》。《style id='11'>《分布式计算》6，3（1993），334–352。