Epistemic Reasoning with Byzantine-Faulty Agents

Roman Kuznets[1]*, Laurent Prosperi[2], Ulrich Schmid[1], and Krisztina Fruzsa[1]**

[1] TU Wien, Vienna, Austria
{rkuznets,s,kfruzsa}@ecs.tuwien.ac.at
[2] ENS Paris-Saclay, Cachan, France
laurent.prosperi@ens-cachan.fr

> ...By our remembrances of days foregone,
> Such were our faults, or then we thought them none.
>
> W. Shakespeare, *All's Well That Ends Well*

Abstract. We introduce a novel comprehensive framework for epistemic reasoning in multi-agent systems where agents may behave asynchronously and may be byzantine faulty. Extending Fagin et al.'s classic runs-and-systems framework to agents who may arbitrarily deviate from their protocols, it combines epistemic and temporal logic and incorporates fine-grained mechanisms for specifying distributed protocols and their behaviors. Besides our framework's ability to express any type of faulty behavior, from fully byzantine to fully benign, it allows to specify arbitrary timing and synchronization properties. As a consequence, it can be adapted to any message-passing distributed computing model we are aware of, including synchronous processes and communication, (un-)reliable uni- / multi- / broadcast communication, and even coordinated action. The utility of our framework is demonstrated by formalizing the *brain-in-a-vat* scenario, which exposes the substantial limitations of what can be known by asynchronous agents in fault-tolerant distributed systems. Given the knowledge of preconditions principle, this restricts preconditions that error-prone agents can use in their protocols. In particular, it is usually necessary to relativize preconditions with respect to the correctness of the acting agent.

1 Introduction

At least since the groundbreaking work by Halpern and Moses [13], the knowledge-based approach [7] is known as a powerful tool for analyzing distributed systems. In a nutshell, it combines epistemic logic [15] and temporal logic to reason about knowledge and belief in *multi-agent systems* (MAS). Standard epistemic logic relies on Kripke models $\mathcal{M}$ that describe possible global states s of the system, where atomic propositions, e.g., "$x_i = 0$ for a local variable of agent i" or "i has witnessed an external event e," hold true or not, along with an indistinguishability relation $s \sim_i s'$ that tells that i cannot distinguish state s from s' based on its local information. Knowledge of a statement φ about the system in global state s is represented by a modal *knowledge operator* K_i, written $(\mathcal{M}, s) \models K_i\varphi$. Agent i knows φ at global state s iff φ holds in every global state s' that i cannot distinguish from s.

In the *interpreted runs-and-systems* framework for reasoning about distributed and other multi-agent systems [7,13], the semantics of Kripke models is combined with a complex machinery representing runs of distributed MAS, thus, obtaining an additional temporal structure. For the set of all possible runs r of a system $\mathcal{I}$, all possible global states $r(t)$ in all runs $r \in \mathcal{I}$ over discrete time $t \in \mathbb{T} = \mathbb{N}$ are considered. The accessibility relation is also dictated by the distributed component:

* Supported by the Austrian Science Fund (FWF) project RiSE/SHiNE (S11405) and ADynNet (P28182).
** PhD student in the FWF doctoral program LogiCS (W1255).

two global states $r(t)$ and $r'(t')$ are indistinguishable for agent i iff i has the same local state in both, formally, $r_i(t) = r'_i(t')$. Therefore, i knows φ at time t in run $r \in \mathcal{I}$, formally,

$$(\mathcal{I}, r, t) \models K_i\varphi \qquad \text{iff} \qquad (\mathcal{I}, r', t') \models \varphi \quad \begin{array}{l}\text{in every } r' \in \mathcal{I} \text{ and}\\ \text{for every } t' \text{ with } r_i(t) = r'_i(t').\end{array}$$

Here φ can be a formula containing arbitrary atomic propositions, as well as other knowledge operators and temporal modalities such as $\Diamond$ (eventually) and $\Box$ (always), combined by standard logical operators $\neg, \land, \lor, \rightarrow$. Note that agents do not generally know the global time.

Related work. Whereas the knowledge-based approach has been used successfully for distributed computing problems in systems with uncertainty but no failures [1,2,11], few papers apply epistemic reasoning to byzantine agents that can disseminate false information. Agents suffering from crash and from send omission failures were studied in [25], primarily in the context of agreement problems, which require standard [5] or continual [14] common knowledge. More recent results are unbeatable consensus algorithms in synchronous systems with crash failures [3] and the discovery of the importance of *silent choirs* [12] for message-optimal protocols in crash-resilient systems. Still, to the best of our knowledge, the only attempt to extend epistemic reasoning to systems with some byzantine[3] faults [19] was made in Michel's PhD thesis published as [22], where faulty agents may deviate from their protocols by sending wrong messages. Even there erroneous behavior is restricted to actions that could have also been observed in some correct execution, meaning that Michel's faulty agents may not behave truly arbitrarily.

To some extent, fault-tolerance has also been considered for general multi-agent systems. For non-fault-tolerant MAS, temporal-epistemic languages like CTLK [7] and even model checkers like MCMAS [20] exist, which can be used for specification and automatic verification of temporal-epistemic properties. For MAS that may suffer from faults, replication-based fault-tolerance techniques [9], diagnosis-based approaches [16], lying agents [4], and even fault-injection based model mutation and model checking [6] have been considered. However, to the best of our knowledge, a comprehensive epistemic reasoning framework that also allows byzantine agents did not exist so far.

Contributions and paper organization. In Sects. 2 and 3, we present the cornerstones of our comprehensive modeling and analysis framework for epistemic reasoning about fault-tolerant distributed message-passing systems, the full version of which is available as a comprehensive technical report [18]. We demonstrate its utility in Sect. 5, by deriving generic results about what asynchronous agents can(not) know in the presence of byzantine faults. In order to achieve this, we first introduce in Sect. 4 a general method of *run modifications*: to show that agent i cannot know some fact φ, it is sufficient to construct a modified run, with changes imperceptible for agent i, that makes φ false. This way, we obtain our central result, the "brain-in-a-vat lemma" (with the proof relegated to Appendix A) stating that, no matter what it observed, an agent can never rule out the possibility of these observations being wholly fictitious results of its malfunction [26]. Our findings imply that the knowledge of preconditions principle [23] (any precondition for action must be known to the acting agent) severely restricts the kinds of preconditions acceptable in such systems. Thus, we introduce epistemic modalities that convert a desired property, e.g., an occurrence of an event, to a knowable precondition in Sect. 6. Finally, Sect. 7 contains some conclusions and directions of future work.

2 Runs and Systems with Byzantine Faults

We introduce our version of the runs-and-systems framework enhanced with active byzantine agents, which provides the basis for epistemic reasoning in this setting. To prevent the waste of space by multiple definition environments, we give the following series of formal definitions as ordinary text marking the defined objects by italics; consult [18] for all the details.

[3] The term "byzantine" originated from [19]. Leslie Lamport chose a defunct country to avoid offending anyone living and also as a pun [27, p. 39] because generals from Byzantium could, in fact, be expected to behave in a byzantine (i.e., devious or treacherous) fashion. Unfortunately, this pun might be responsible for the ensuing unnecessary ([28]) capitalization of the word even for faults unrelated to Byzantium proper.

We consider a non-empty finite set $\mathcal{A} = \{1, \dots, n\}$ of *agents*, representing individuals and/or computing units. Agent i can perform *actions* from $Actions_i$, e.g., send *messages*, and can witness *events* from $Events_i$, e.g., message delivery. We group all actions and events, collectively termed *haps*[4], taking place after *timestamp* t and no later than $t+1$ into a *round*, denoted $t + ½$, and treat all haps of the round as happening simultaneously. Global system timestamps taken from $\mathbb{T} = \mathbb{N}$ are not accessible to our *asynchronous agents*, who need to be woken up during a round to record the passage of time. A *local state* $r_i(t+1)$, referred to as (*process-time*) *node* $(i, t+1)$, describes the local view of the system by agent $i \in \mathcal{A}$ after round $t + ½$. Nodes $(i, 0)$ correspond to *initial local states* $r_i(0)$, taken from a set Σ_i. The set of all possible tuples of initial local states is

$$\mathscr{G}(0) := \prod_{i \in \mathcal{A}} \Sigma_i.$$

We assume $r_i(t)$ to be a list of all haps as observed by i in rounds it was *active* in, grouped by round, i.e., if agent i is *awoken* in round $t + ½$, then

$$r_i(t+1) = X : r_i(t),^{5} \qquad \text{where} \qquad X \subseteq Haps_i := Actions_i \sqcup Events_i$$

is the set of all *internal actions* and *external events* as perceived by i in round $t + ½$. Agents *passive* in the round have no record of it:

$$r_i(t+1) = r_i(t).$$

We denote

$$Actions := \bigcup_{i \in \mathcal{A}} Actions_i, \qquad Events := \bigcup_{i \in \mathcal{A}} Events_i, \qquad \text{and} \qquad Haps := Actions \sqcup Events$$

to be sets of all actions, events, and haps respectively. Each agent has a *protocol* dictating its actions (more details below). Actions prompted by the protocol are deemed *correct*, whereas actions imposed by the environment in circumvention of the protocol are *byzantine* (even when they mirror correct protocol actions). In addition to acting outside its protocol, a *byzantine agent i* may incorrectly record its actions and/or witnessed events. Events recorded correctly (incorrectly) are *correct* (*byzantine*). Thus, $r_i(t)$ may not match reality. Still agents possess *perfect recall*: though imperfect, their memories never change. The set of all local states of i is denoted $\mathscr{L}_i$.

An accurate record $r_\epsilon(t+1)$ of the system after round $t + ½$ is possessed only by the *environment* $\epsilon \notin \mathcal{A}$ that controls everything but agents' protocols: it determines which agents wake up and which become faulty; it fully controls all byzantine haps (including faulty actions by the agents); it enforces physical and causal laws; and it is the source of indeterminacy (of the type involved in throwing dice). The environment is also responsible for message passing, the details of which can be found in [18] but are largely irrelevant for the results presented in this paper. The crucial features are:

- messages are agent-to-agent;
- correctly sending (resp. receiving) a message is an action (resp. event);
- each sent message, correct or byzantine alike, is supplied with a unique *global message identifier*, or *GMI*, inaccessible for agents and used by ϵ to ensure the causality of message delivery, i.e., that an unsent message cannot be correctly received.

The *global state*

$$r(t+1) := \Big(r_\epsilon(t+1), r_1(t+1), \dots, r_n(t+1)\Big)$$

after round $t + ½$ consists of all local states $r_i(t+1)$, as well as $r_\epsilon(t+1)$. The set of all global states is denoted $\mathscr{G}$.

[4] Cf. "And whatsoever else shall *hap* to-night, Give it an understanding but no tongue." W. Shakespeare, *Hamlet*, Act I, Scene 2.

[5] ':' stands for concatenation.

To distinguish $o \in \mathit{Haps}_i$ from the same o observed by another agent $j \neq i$ and to facilitate message delivery, ϵ transforms each action $a \in \mathit{Actions}_i$ initiated by i's protocol (and, hence, correct) into an extended format

$$\mathit{global}\,(i, t, a) \in \overline{\mathit{GActions}_i},$$

e.g., incorporating a unique time-based GMI for each sent message. The main requirement is that *global* be one-to-one (see [18] for details). Haps in $r_\epsilon(t)$ are recorded in this *global*, or *environment's, view*. The sets of globally recorded correct actions/events/haps are denoted by adding G to the notations above, e.g., $\overline{\mathit{GEvents}_i}$ are pairwise disjoint sets of all i's correct events in global notation.

The duality between the local and global views is also crucial for allowing agents to have false memories. Every correct event $E = \mathit{global}\,(i, t, e)$ for $e \in \mathit{Events}_i$ has a faulty counterpart $\mathit{fake}\,(i, E)$ representing i being mistaken about witnessing e, with both E and $\mathit{fake}\,(i, E)$ recorded as e in $r_i(t+1)$. Further, a faulty agent may misinterpret its own actions, by mistakenly believing to have performed $A' = \mathit{global}\,(i, t, a')$ despite actually performing $A = \mathit{global}\,(i, t, a)$, where $a, a' \in \mathit{Actions}_i$. This is coded as a byzantine event $\mathit{fake}\,(i, A \mapsto A')$ resulting in a' recorded in $r_i(t+1)$ but the causal effects on the whole system being those of A. The case of $A = A'$ corresponds to a correctly recorded byzantine action. In addition, either of A or A' can be a special byzantine action **noop** representing the absence of actions. If $A = \mathbf{noop}$, the agent believes to have performed a' without doing anything. If $A' = \mathbf{noop}$, the agent performs A without leaving a local record. Finally,

$$\mathit{fail}\,(i) := \mathit{fake}\,(i, \mathbf{noop} \mapsto \mathbf{noop})$$

represents the byzantine inaction and leaves no local record for i. The set of all i's byzantine events, whether mimicking correct events or correct actions, is denoted $\mathit{BEvents}_i$, with

$$\mathit{BEvents} := \bigsqcup_{i \in \mathcal{A}} \mathit{BEvents}_i.$$

Apart from correct $\overline{\mathit{GEvents}_i}$ and byzantine $\mathit{BEvents}_i$, the environment issues at most one of the *system events*

$$\mathit{SysEvents}_i := \Big\{ \mathit{go}(i), \mathit{sleep}\,(i), \mathit{hibernate}\,(i) \Big\}$$

per agent i per round. The correct event $\mathit{go}(i)$ activates i's protocol (see below) for the round. Events $\mathit{sleep}\,(i)$ and $\mathit{hibernate}\,(i)$ represent i failing to implement its protocol, with $\mathit{sleep}\,(i)$ enforcing a local record of the round, whereas $r_i(t+1) = r_i(t)$ is possible for $\mathit{hibernate}\,(i)$. Thus,

$$\mathit{GEvents}_i := \overline{\mathit{GEvents}_i} \sqcup \mathit{BEvents}_i \sqcup \mathit{SysEvents}_i$$

with

$$\mathit{GEvents} := \bigsqcup_{i \in \mathcal{A}} \mathit{GEvents}_i \qquad \text{and} \qquad \mathit{GHaps} := \mathit{GEvents} \sqcup \overline{\mathit{GActions}}.$$

The first event from $\mathit{BEvents}_i$, or $\mathit{sleep}\,(i)$, or $\mathit{hibernate}\,(i)$ in a run turns i into *byzantine*. Overall,

$$r_\epsilon(t+1) := X : r_\epsilon(t)$$

for the set $X \subseteq \mathit{GHaps}$ of all haps from round $t + ½$.

As in this definition of $\mathit{GEvents}_i$, throughout the paper horizontal bars signify the correct subsets of phenomena in question, i.e.,

$$\overline{\mathit{GEvents}_i} \subseteq \mathit{GEvents}_i, \qquad \overline{\mathit{GHaps}} \subseteq \mathit{GHaps},$$

etc. Later, this would also apply to formulas, e.g., $\overline{\mathit{occurred}_i}(o)$ is a correctly recorded occurrence of $o \in \mathit{Haps}_i$ whereas $\mathit{occurred}_i(o)$ is any recorded occurrence. Note that this distinction is only made in the global format because locally agents do not distinguish correct haps from byzantine.

Each agent's *protocol*

$$P_i \colon \mathscr{L}_i \to \wp(\wp(\mathit{Actions}_i)) \setminus \{\varnothing\}$$

is designed to choose a set of actions based on i's current local state in order to achieve some collective goal. At least one set of actions is always available. In case of multiple options, the choice is up to the *adversary* part of the environment. For asynchronous agents, the global timestamp cannot be inferred from their local state.

The environment governs all events, correct, byzantine, and system, via an environment protocol

$$P_\epsilon \colon \mathbb{T} \to \wp(\wp(\mathit{GEvents})) \setminus \{\varnothing\},$$

which *can* depend on a timestamp $t \in \mathbb{T}$ but should not depend on the current state because the environment is assumed to be an impartial physical medium. Both the environment's and agents' protocols are non-deterministic, with the choice among the possible options arbitrarily made by the *adversary* part of the environment. It is also required that all events of round $t + ½$ be mutually compatible (for time t). The complete list of these *coherency* conditions can be found in [18], of which the following are relevant for this paper:

(a) at most one event from $\mathit{SysEvents}_i$ at a time is issued per agent;
(b) a correct event observed as e by agent i is never accompanied by a byzantine event that i would also register as e, i.e., an agent cannot be mistaken about observing an event that did happen.[6]

Both the global run $r \colon \mathbb{T} \to \mathscr{G}$ and its local parts $r_i \colon \mathbb{T} \to \mathscr{L}_i$ provide a sequence of snapshots of system states. Given the *joint protocol*

$$P := (P_1, \ldots, P_n)$$

and the environment's protocol P_ϵ, we focus on $\tau_{f,P_\epsilon,P}$*-transitional runs* r that result from following these protocols and are built according to a *transition relation*

$$\tau_{f,P_\epsilon,P} \subseteq \mathscr{G} \times \mathscr{G}$$

for asynchronous agents at most $f \geq 0$ of which may turn byzantine per run. Each such transitional run progresses by ensuring that

$$r\left(t\right) \;\; \tau_{f,P_\epsilon,P} \;\; r\left(t+1\right), \qquad \text{i.e.,} \qquad \Big(r\left(t\right), r\left(t+1\right)\Big) \in \tau_{f,P_\epsilon,P},$$

for each timestamp $t \in \mathbb{T}$.

Figure 1 represents one round of an asynchronous system governed by $\tau_{f,P_\epsilon,P}$, which consists of the following five consecutive phases:

1. Protocol phase A non-empty range

$$P_\epsilon\left(t\right) \subseteq \wp(\mathit{GEvents})$$

of possible coherent sets of events is determined by the environment's protocol P_ϵ; for each $i \in \mathcal{A}$, a non-empty range

$$P_i\left(r_i\left(t\right)\right) \subseteq \wp(\mathit{Actions}_i)$$

of possible sets of i's actions is determined by the agents' joint protocol P.

2. Adversary phase The adversary non-deterministically picks one (coherent) set

$$X_\epsilon \in P_\epsilon\left(t\right)$$

and one set

$$X_i \in P_i\left(r_i\left(t\right)\right)$$

for each $i \in \mathcal{A}$ from their ranges.

[6] Prohibition (b) does not extend to *actions*, which need not be correctly recorded.

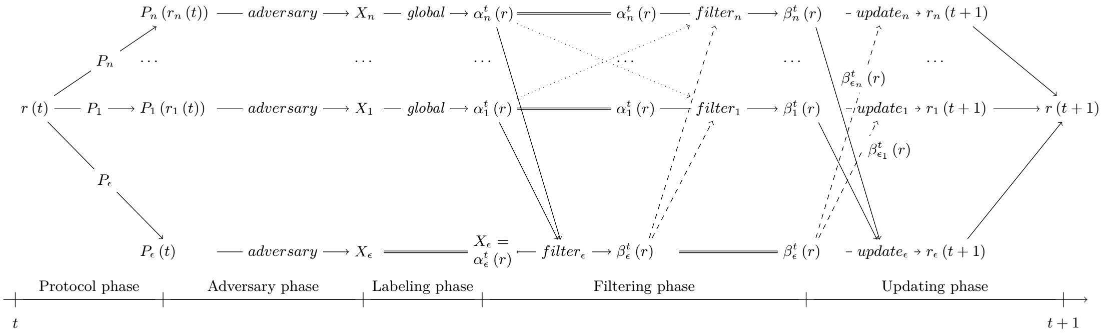
$P_n(r_n(t))$ — adversary ⟶ X_n — global ⟶ $\alpha_n^t(r)$ ═ $\alpha_n^t(r)$ — $filter_n$ ⟶ $\beta_n^t(r)$ - $update_n$ → $r_n(t+1)$
P_n
…
$r(t)$ — P_1 ⟶ $P_1(r_1(t))$ — adversary ⟶ X_1 — global ⟶ $\alpha_1^t(r)$ ═ $\alpha_1^t(r)$ — $filter_1$ ⟶ $\beta_1^t(r)$ - $update_1$ → $r_1(t+1)$ ⟶ $r(t+1)$
$\beta_{\epsilon_n}^t(r)$
$\beta_{\epsilon_1}^t(r)$
P_ϵ
$P_\epsilon(t)$ — adversary ⟶ X_ϵ ═ $X_\epsilon = \alpha_\epsilon^t(r)$ — $filter_\epsilon$ ⟶ $\beta_\epsilon^t(r)$ ═ $\beta_\epsilon^t(r)$ - $update_\epsilon$ → $r_\epsilon(t+1)$
Protocol phase
Adversary phase
Labeling phase
Filtering phase
Updating phase
t
$t+1$

Fig. 1: Details of round $t + ½$ of a $\tau_{f,P_\epsilon,P}$-transitional run r.

3. Labeling phase Locally represented actions in each X_i are translated into the global format:

$$\alpha_i^t(r) := \left\{ global(i, t, a) \mid a \in X_i \right\} \subseteq \overline{GActions}_i.$$

4. Filtering phase Functions $filter_\epsilon$ and $filter_i$ for each $i \in \mathcal{A}$ remove all attempted events from

$$\alpha_\epsilon^t(r) := X_\epsilon$$

and actions from $\alpha_i^t(r)$ that would violate causality. More precisely, the filtering phase is performed in two stages:

1. $filter_\epsilon$ filters out causally impossible events based
   (a) on the current global state $r(t)$, which could not have been accounted for by P_ϵ,
   (b) on $\alpha_\epsilon^t(r)$, and
   (c) on all $\alpha_i^t(r)$, which are not accessible to P_ϵ either.
   Specifically, two kinds of events are causally impossible and, accordingly, removed by $filter_\epsilon$:
   (1) each correct receive event that has no matching send either in the global history $r(t)$ or in the current round[7] and
   (2) all byzantine events if they would have resulted in more than f agents becoming faulty (cf. [18] for details).
   The resulting set of events to actually occur in round $t + ½$ is denoted

$$\beta_\epsilon^t(r) := filter_\epsilon\Big(r(t), \quad \alpha_\epsilon^t(r), \quad \alpha_1^t(r), \quad \dots, \quad \alpha_n^t(r)\Big).$$

2. For each agent i, $filter_i$ either
   – removes all actions whenever $go(i) \notin \beta_\epsilon^t(r)$ or
   – leaves $\alpha_i^t(r)$ unchanged otherwise.
   The resulting sets of actions to be actually performed by agents in round $t + ½$ are denoted

$$\beta_i^t(r) := filter_i\Big(\alpha_1^t(r), \quad \dots, \quad \alpha_n^t(r), \quad \beta_\epsilon^t(r)\Big).^8$$

Note that

$$\beta_i^t(r) \subseteq \alpha_i^t(r) \subseteq \overline{GActions}_i \qquad \text{and} \qquad \beta_\epsilon^t(r) \subseteq \alpha_\epsilon^t(r) \subseteq GEvents.$$

5. Updating phase The resulting mutually causally consistent events $\beta_\epsilon^t(r)$ and actions $\beta_i^t(r)$ are appended to the global history $r(t)$; for each $i \in \mathcal{A}$, all non-system events from

$$\beta_{\epsilon_i}^t(r) := \beta_\epsilon^t(r) \cap GEvents_i$$

[7] In $\alpha_\epsilon^t(r)$ for byzantine sends or in $\alpha_i^t(r)$ for correct ones that will be actually performed (see filtering stage 2).

[8] Arguments $\alpha_j^t(r)$ for $j \neq i$ are redundant here but will be used in future extensions.

and all actions $\beta_i^t(r)$ are appended in the local form to the local history $r_i(t)$, which may remain unchanged if no action or event triggers an update or be appended with the empty set if an update is triggered only by a system event $go(i)$ or $sleep\,(i)$:

$$r_\epsilon\,(t+1) := update_\epsilon\left(r_\epsilon\,(t)\,,\quad \beta_\epsilon^t\,(r),\quad \beta_1^t\,(r),\quad \dots,\quad \beta_n^t\,(r)\right);$$
$$r_i\,(t+1) := update_i\left(r_i\,(t)\,,\quad \beta_i^t\,(r),\quad \beta_\epsilon^t\,(r)\right).^9$$

Since only the protocol phase depends on the specific protocols P and P_ϵ, the operations in the remaining phases 2–5 can be grouped into a *transition template* τ_f that produces a transition relation $\tau_{f,P_\epsilon,P}$ given P and P_ϵ.

Properties of runs that cannot be implemented on a round-by-round basis, e.g., *liveness properties* requiring certain things to happen in a run *eventually*, are enforced by restricting the set of allowable runs by *admissibility conditions* Ψ defined as subsets of the set R of all transitional runs. For example, no goal can be achieved unless agents are guaranteed to act from time to time. Thus, it is standard to impose the *Fair Schedule* (*FS*) admissibility condition, which ensures that each *correct* agent is eventually given a possibility to follow its protocol:

$$FS := \left\{ r \,\middle|\, (\forall i \in \mathcal{A})\,(\forall t \in \mathbb{T})\,(\exists t' \geq t)\,\beta_\epsilon^{t'}\,(r) \cap SysEvents_i \neq \varnothing \right\}.$$

In scheduling terms, *FS* demands that the environment either provide or wrongfully deny CPU time for every processor infinitely many times. Thus, a process is always given an opportunity to act, unless its faults *sleep* (i) and/or *hibernate* (i) persist infinitely often.

Definition 1 (Context, agent-context, consistent and non-excluding runs). *A* context

$$\gamma = \left(P_\epsilon, \mathscr{G}(0), \tau_f, \Psi\right)$$

consists of the environment's protocol P_ϵ, a set of global initial states $\mathscr{G}(0)$, a transition template τ_f for $f \geq 0$, and an admissibility condition Ψ. For a joint protocol P, we call

$$\chi = (\gamma, P)$$

an agent-context*. A run $r \in R$ is called* weakly χ-consistent *if $r(0) \in \mathscr{G}(0)$ and the run is $\tau_{f,P_\epsilon,P}$-transitional. A weakly χ-consistent run r is called* (strongly) χ-consistent *if $r \in \Psi$. The set of all χ-consistent runs is denoted R^χ. Agent-context χ is called* non-excluding *if any finite prefix of a weakly χ-consistent run can be extended to a strongly χ-consistent run.*

We distinguish types of agents depending on their expected malfunctions. Let

$$FEvents_i := BEvents_i \sqcup \left\{ sleep\,(i), hibernate\,(i) \right\}$$

be all faulty events for agent i.

Definition 2 (Agent types). *Environment's protocol P_ϵ makes an agent $i \in \mathcal{A}$:*

(i) fallible *if, for any $X \in P_\epsilon\,(t)$,*
$$\{fail\,(i)\} \cup X \in P_\epsilon\,(t)\,;$$

(ii) delayable *if, for any $X \in P_\epsilon\,(t)$,*
$$X \setminus GEvents_i \in P_\epsilon\,(t)\,;$$

(iii) error-prone *if, for any $X \in P_\epsilon\,(t)$ and any $Y \subseteq FEvents_i$, the set*
$$Y \sqcup (X \setminus FEvents_i) \in P_\epsilon\,(t)$$
whenever it is coherent;

(iv) gullible *if, for any* $X \in P_\epsilon(t)$ *and any* $Y \subseteq \mathit{FEvents}_i$*, the set*

$$Y \sqcup (X \setminus \mathit{GEvents}_i) \in P_\epsilon(t)$$

whenever it is coherent;

(v) fully byzantine *if it is error-prone and gullible.*

Thus, fallible agents can always be faulty with the same behavior; delayable agents can be prevented from waking up; error-prone (gullible) agents can exhibit any faults in addition to (without) correct events, thus, implying fallibility (delayability); fully byzantine agents exhibit the widest range of faults.

3 Epistemic Modeling of Byzantine Faults

The runs-and-systems framework is traditionally used as a basis for interpreted systems, a special kind of Kripke models for multi-agent distributed environments [7]. For an agent-context χ, we consider pairs (r, t') of a run $r \in R^\chi$ and timestamp $t' \in \mathbb{T}$. A *valuation function*

$$\pi \colon \mathit{Prop} \to \wp(R^\chi \times \mathbb{T})$$

determines where an atomic proposition from *Prop* is true. The determination is arbitrary except for a small set of *designated atomic propositions* (and more complex formulas built from them) whose truth value is fully determined by r and t'. More specifically, for $i \in \mathcal{A}$, $o \in \mathit{Haps}_i$, and $t \in \mathbb{T}$ such that $t \leq t'$,

$\mathit{correct}_{(i,t)}$ is true at (r, t') iff no faulty event happened to i by timestamp t, i.e., no event from $\mathit{FEvents}_i$ appears in the $r_\epsilon(t)$ prefix of the $r_\epsilon(t')$ part of $r(t')$;

$\mathit{correct}_i$ is true at (r, t') iff no faulty event happened to i yet, i.e., no event from $\mathit{FEvents}_i$ appears in $r_\epsilon(t')$ (this is the formal definition of what it means for an agent i to be *correct*; the agent is *faulty* or *byzantine* iff it is not correct);

$\mathit{fake}_{(i,t)}(o)$ is true at (r, t') iff i has a *faulty* reason to believe that $o \in \mathit{Haps}_i$ occurred in round $t - ½$, i.e., $o \in r_i(t)$ because (at least in part) of some

$$O \in \mathit{BEvents}_i \cap \beta_\epsilon^{t-1}(r);$$

$\overline{\mathit{occurred}}_{(i,t)}(o)$ is true at (r, t') iff i has a *correct* reason to believe that $o \in \mathit{Haps}_i$ occurred in round $t - ½$, i.e., $o \in r_i(t)$ because (at least in part) of some

$$O \in \left(\overline{\mathit{GEvents}_i} \cap \beta_\epsilon^{t-1}(r)\right) \sqcup \beta_i^{t-1}(r);$$

$\overline{\mathit{occurred}}_i(o)$ is true at (r, t') iff at least one of $\overline{\mathit{occurred}}_{(i,m)}(o)$ for $1 \leq m \leq t'$ is; also

$$\overline{\mathit{occurred}}(o) := \bigvee_{i \in \mathcal{A}} \overline{\mathit{occurred}}_i(o);$$

$\mathit{occurred}_i(o)$ is true at (r, t') iff either $\overline{\mathit{occurred}}_i(o)$ is or at least one of $\mathit{fake}_{(i,m)}(o)$ for $1 \leq m \leq t'$ is.

An *interpreted system* is a pair $\mathcal{I} = (R^\chi, \pi)$. We combine the standard epistemic language with the standard temporal language

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_i\varphi \mid \Box\varphi$$

for $p \in \mathit{Prop}$ and $i \in \mathcal{A}$, with other Boolean connectives defined in the usual way and $\Diamond\varphi := \neg\Box\neg\varphi$. Truth for these *formulas* is defined in the standard way, in particular, for a run $r \in R^\chi$, timestamp $t \in \mathbb{T}$, atomic proposition $p \in \mathit{Prop}$, agent $i \in \mathcal{A}$, and formula φ we have

$(\mathcal{I}, r, t) \models p$ iff $(r, t) \in \pi(p)$;
$(\mathcal{I}, r, t) \models K_i\varphi$ iff $(\mathcal{I}, r', t') \models \varphi$ for any $r' \in R^\chi$ and $t' \in \mathbb{T}$ such that $r_i(t) = r'_i(t')$; and
$(\mathcal{I}, r, t) \models \Box\varphi$ iff $(\mathcal{I}, r, t') \models \varphi$ for all $t' \geq t$ in the same run r.
A formula φ is valid in $\mathcal{I}$, written $\mathcal{I} \models \varphi$, iff $(\mathcal{I}, r, t) \models \varphi$ for all $r \in R^\chi$ and $t \in \mathbb{T}$.

Due to the coherency of protocol P_ϵ, an agent cannot be both right and wrong about an occurrence of an event, i.e.,

$$\mathcal{I} \models \neg\Big(\overline{occurred}_{(i,t)}(e) \wedge fake_{(i,t)}(e)\Big)$$

for any $i \in \mathcal{A}$, event $e \in Events_i$, and $t \in \mathbb{T}$. Note that for actions this *can* happen.

Following the concept from [8] of global events that are local for an agent, we define conditions under which formulas can be treated as such local events. A formula φ is called *localized for i within an agent-context χ* iff

$$r_i(t) = r'_i(t') \qquad \text{implies} \qquad (\mathcal{I}, r, t) \models \varphi \Longleftrightarrow (\mathcal{I}, r', t') \models \varphi$$

for any $\mathcal{I} = (R^\chi, \pi)$, runs $r, r' \in R^\chi$, and $t, t' \in \mathbb{T}$. By these definitions, we immediately obtain:

Lemma 3. *The following statements are valid for any formula φ localized for an agent $i \in \mathcal{A}$ within an agent-context χ and any interpreted system $\mathcal{I} = (R^\chi, \pi)$:*

$$\mathcal{I} \models \varphi \leftrightarrow K_i\varphi \qquad \textit{and} \qquad \mathcal{I} \models \neg\varphi \leftrightarrow K_i\neg\varphi.$$

The knowledge of preconditions principle [23] postulates that in order to act on a precondition an agent must be able to infer it from its local state. Thus, the preceding lemma shows that formulas localized for i can *always* be used as preconditions. Our first observation is that the agent's *perceptions* of a run are one example of such epistemically acceptable (though not necessarily reliable) preconditions:

Lemma 4. *For any agent-context χ, agent $i \in \mathcal{A}$, and local hap $o \in Haps_i$, the formula $occurred_i(o)$ is localized for i within χ.*

4 Run modifications

By contrast, as we will demonstrate, correctness of these perceptions is not localized for i and, hence, cannot be the basis for actions. In fact, correctness can never be established by an agent. Such impossibility results are proved by means of controlled *run modifications.*

Definition 5 (Intervention, adjustment). *A function*

$$\rho \colon R^\chi \longrightarrow \wp(\overline{GActions_i}) \times \wp(GEvents_i)$$

is called an i-intervention for an agent-context χ and agent $i \in \mathcal{A}$. *A* joint intervention

$$B = (\rho_1, \dots, \rho_n)$$

consists of i-interventions ρ_i for each agent $i \in \mathcal{A}$. An adjustment

$$[B_t; \dots; B_0]$$

is a sequence of joint interventions $B_0 \dots, B_t$ to be performed at rounds from ½ to $t +$ ½ for some timestamp $t \in \mathbb{T}$.

An i-intervention

$$\rho(r) = (X, X_\epsilon)$$

applied to a round $t + ½$ of a given run r can be seen as a meta-action modifying the results of this round for i in such a way that

$$\beta^t_i\left(r'\right) = X \qquad \text{and} \qquad \beta^t_{\epsilon_i}\left(r'\right) = \beta^t_\epsilon\left(r'\right) \cap \mathit{GEvents}_i = X_\epsilon$$

in the artificially constructed new run r'. We denote

$$\mathfrak{a}\rho(r) := X \qquad \text{and} \qquad \mathfrak{e}\rho(r) := X_\epsilon.$$

Accordingly, a joint intervention $(\rho_1, \ldots, \rho_n)$ prescribes

$$\text{actions } \beta^t_i\left(r'\right) = \mathfrak{a}\rho_i(r) \text{ for each agent } i \qquad \text{and} \qquad \text{events } \beta^t_\epsilon\left(r'\right) = \bigsqcup_{i\in\mathcal{A}} \mathfrak{e}\rho_i(r)$$

for the round in question. Thus, an adjustment $[B_t; \ldots; B_0]$ fully determines actions and events in the initial $t + 1$ rounds of run r':

Definition 6 (Adjusting runs). *Let*

$$\mathit{adj} = [B_t; \ldots; B_0]$$

be an adjustment where

$$B_m = (\rho^m_1, \ldots, \rho^m_n)$$

for each $0 \leq m \leq t$ *and each* ρ^m_i *be an* i*-intervention for an agent-context* $\chi = \Big(\big(P_\epsilon, \mathscr{G}(0), \tau_f, \Psi\big), P\Big)$. *A run* r' *is* obtained from $r \in R^\chi$ by adjustment adj *iff for all* $t' \leq t$*, all* $T > t$*, and all* $i \in \mathcal{A}$*,*

1. $r'\left(0\right) \quad := \quad r\left(0\right),$
2. $r'_i\left(t'+1\right) \qquad := \qquad \mathit{update}_i\Big(r'_i\left(t'\right), \quad \mathfrak{a}\rho^{t'}_i(r), \quad \bigsqcup_{i\in\mathcal{A}} \mathfrak{e}\rho^{t'}_i(r)\Big),$
3. $r'_\epsilon\left(t'+1\right) \qquad := \qquad \mathit{update}_\epsilon\Big(r'_\epsilon\left(t'\right), \quad \bigsqcup_{i\in\mathcal{A}} \mathfrak{e}\rho^{t'}_i(r), \quad \mathfrak{a}\rho^{t'}_1(r), \quad \ldots, \quad \mathfrak{a}\rho^{t'}_n(r)\Big),$
4. $r'(T)\ \tau_{f,P_\epsilon,P}\ r'(T+1).$

We denote by $R\left(\tau_{f,P_\epsilon,P}, r, \mathit{adj}\right)$ *the set of all runs obtained from* r *by* adj*.*

Note that generally not all adjusted runs are $\tau_{f,P_\epsilon,P}$-transitional, i.e., obey Prop. 4 also for $t' \leq t$. Thus, special care is required to produce $\tau_{f,P_\epsilon,P}$-transitional adjustments with required properties. To demonstrate the impossibility of establishing knowledge of correctness, we use several adjustment types to formalize the infamous *brain in a vat* scenario[10], where one agent, the "brain," is to experience a fabricated, i.e., faulty, version of its local history from a given run, whereas all other agents are to remain in their initial states (and made faulty or not at will). This is achieved by using interventions

(a) Fake_i for brain i,
(b) $\mathit{CFreeze}$ for other agents j that are to be correct, and
(c) $\mathit{BFreeze}_j$ for other agents j that are to be byzantine.

Definition 7 (Interventions Fake_i**,** $\mathit{CFreeze}$**, and** $\mathit{BFreeze}_j$**).** *For an agent-context* χ*, agent* $i \in \mathcal{A}$*, and run* $r \in R^\chi$*, we define* i*-interventions*

$$\mathit{CFreeze}\left(r\right) := (\varnothing, \varnothing),$$

$$\mathit{BFreeze}_i\left(r\right) := \Big(\varnothing, \big\{\mathit{fail}\left(i\right)\big\}\Big),$$

$$\begin{aligned} \mathit{Fake}^t_i\left(r\right) := \Big(\varnothing,& \\ &\big\{\mathit{fail}\left(i\right)\big\} \quad \cup \quad \big(\beta^t_\epsilon\left(r\right) \cap \mathit{BEvents}_i\big) \quad \cup \quad \big\{\mathit{fake}\left(i, E\right) \bigm| E \in \beta^t_\epsilon\left(r\right) \cap \overline{\mathit{GEvents}_i}\big\} \quad \cup \\ &\big\{\mathit{fake}\left(i, \boldsymbol{noop} \mapsto A\right) \bigm| A \in \beta^t_i\left(r\right)\big\} \quad \sqcup \quad \big\{\mathit{sleep}\left(i\right) \bigm| r_i(t+1) \neq r_i(t)\big\}\Big). \end{aligned} \tag{1}$$

[10] For connections to the semantic externalism and a survey of philosophical literature on the subject, see [26].

The following Brain-in-a-Vat Lemma 8, whose proof can be found in the Appendix, constructs the desired modified transitional run:

Lemma 8 (Brain in a Vat). *For an agent $i \in \mathcal{A}$, for an agent-context $\chi = \Big(\big(P_\epsilon, \mathscr{G}(0), \tau_f, \Psi\big), P\Big)$ such that P_ϵ makes i gullible and every $j \neq i$ delayable and fallible, for a set*

$$Byz \subseteq \mathcal{A} \setminus \{i\} \qquad \textit{such that} \qquad |Byz| + 1 \leq f,$$

for a run $r \in R^\chi$, and for a timestamp $t > 0$, we consider an adjustment

$$adj = [B_{t-1}; \ldots ; B_0] \qquad \textit{such that} \qquad B_m = (\rho_1^m, \ldots, \rho_n^m)$$

with

$$\rho_i^m = \mathit{Fake}_i^m, \qquad \rho_j^m = \mathit{BFreeze}_j \textit{ for } j \in Byz, \qquad \textit{and} \qquad \rho_j^m = \mathit{CFreeze} \textit{ for } j \notin \{i\} \sqcup Byz$$

for all $0 \leq m \leq t-1$. Then each run $r' \in R\left(\tau_{f,P_\epsilon,P}, r, adj\right)$ satisfies the following properties:

1. *r' is weakly χ-consistent; if $r' \in \Psi$, then $r' \in R^\chi$;*
2. *$(\forall m \leq t)\, r'_i(m) = r_i(m)$;*
3. *$(\forall m \leq t)\, (\forall j \neq i)\ r'_j(m) = r'_j(0)$;*
4. *agents from $\mathcal{A} \setminus \big(\{i\} \sqcup Byz\big)$ remain correct until t;*
5. *i and all agents from Byz become faulty already in round ½;*
6. *$(\forall m < t)\, (\forall j \neq i)\ \beta_{\epsilon_j}^m(r') \subseteq \big\{\mathit{fail}(j)\big\}$. More precisely,*

$$\beta_{\epsilon_j}^m(r') = \varnothing \quad \Longleftrightarrow \quad \rho_j^m = \mathit{CFreeze} \qquad \textit{and} \qquad \beta_{\epsilon_j}^m(r') = \big\{\mathit{fail}(j)\big\} \quad \Longleftrightarrow \quad \rho_j^m = \mathit{BFreeze}_j;$$

7. *$(\forall m < t)\, \beta_{\epsilon_i}^m(r') \setminus \mathit{FEvents}_i = \varnothing$;*
8. *$(\forall m < t)(\forall j \in \mathcal{A})\, \beta_j^m(r') = \varnothing$.*

Corollary 9. *If χ is non-excluding, for $t \in \mathbb{T}$ there is a run $r' \in R^\chi$ constructed according to Lemma 8, such that for any $\mathcal{I} = (R^\chi, \pi)$, $o \in \mathit{Haps}$, $j \in \{i\} \sqcup Byz$, and $k \notin \{i\} \sqcup Byz$,*

$$(\mathcal{I}, r', t) \not\models \overline{\mathit{occurred}}(o), \qquad (\mathcal{I}, r', t) \not\models \mathit{correct}_j, \qquad \textit{and} \qquad (\mathcal{I}, r', t) \models \mathit{correct}_k. \qquad (2)$$

5 Byzantine Limitations of Certainty

The ability to construct a Brain-in-a-Vat run r' in Lemma 8 and its properties in Corollary 9 spell doom for the strategy of asynchronous agents waiting for a definitive proof of correctness before acting. More precisely, agents can never learn a particular event actually happened, nor that they are not byzantine.

Throughout this section, $\chi = \Big(\big(P_\epsilon, \mathscr{G}(0), \tau_f, \Psi\big), P\Big)$ is a non-excluding agent-context such that P_ϵ makes agent $i \in \mathcal{A}$ gullible and every other agent $k \neq i$ delayable and fallible (in particular, this covers the case of fully byzantine agents), $\mathcal{I} = (R^\chi, \pi)$ is an interpreted system, and $o \in \mathit{Haps}$.

Theorem 10. *If $f \geq 1$, then for $k \neq i$ the following statements are valid in $\mathcal{I}$:*

$$\mathcal{I} \models \neg K_i \overline{\mathit{occurred}}(o), \qquad \mathcal{I} \models \neg K_i \mathit{correct}_i, \qquad \textit{and} \qquad \mathcal{I} \models \neg K_i \neg \mathit{correct}_k. \qquad (3)$$

Proof. For any $r \in R^\chi$ and $t \in \mathbb{T}$, by Lemma 8 with $Byz = \varnothing$ and non-exclusiveness of χ, there is $r' \in R^\chi$ such that (2) for $j = i$ and $k \neq i$ holds by Cor. 9.

$$(\mathcal{I}, r, t) \quad \models \quad \neg K_i \overline{\mathit{occurred}}(o) \wedge \neg K_i \mathit{correct}_i \wedge \neg K_i \neg \mathit{correct}_k$$

follows from $r_i(t) = r'_i(t)$ by Lemma 8.2. □

Remark 11. While agent i can never learn that it is correct or that another agent k is faulty, agent i might be able to detect its own faults, for instance, by comparing actions prescribed by its protocol against actions recorded in its local history.

The case of $f = 0$ corresponds to a system without byzantine faults, where correctness of all actions, events, and agents is common knowledge. When $f = 1$, in view of Remark 11, the agent may be able to conclude that all other agents are correct. By reusing the proof of Theorem 10 with $Byz = \{k\}$, we can establish that for $f \geq 2$ this determination is not possible either:

Theorem 12. *If $f \geq 2$ and $k \neq i$, the following validity holds:*

$$\mathcal{I} \models \neg K_i \mathit{correct}_k.$$

6 Epistemes for Distributed Analysis and Design

The results of Sect. 5 clearly show that most desired properties, such as trigger events, cannot be used as preconditions in asynchronous byzantine settings. The knowledge of a precondition φ requirement stated in [23], i.e., that an agent only act on φ when the agent is sure φ is not false, would typically lead for such simple preconditions to no actions being taken at all: even when an asynchronous agent is correct, it can never discount the scenario of being a brain in a vat. This is, in fact, a "human condition," as philosophy and science have yet to provide a definitive way of discounting each of us being a brain in a vat (see [21] for discussion). It then stands to reason that the human response to act as if everything is fine could also be applied in distributed scenarios. This led to the soft or *defeasible* knowledge

$$B_i\varphi := K_i(\mathit{correct}_i \to \varphi)$$

considered, e.g., in [24]. In other words, the agent only considers situations where it has not been compromised and, while φ is the desired property, the agent acts on the precondition $\mathit{correct}_i \to \varphi$ relativized to its correctness.

We believe that this formulation can be improved in at least two directions. Firstly, a typical specification for a fault-tolerant system does not impose any restrictions on byzantine agents. For instance, in distributed consensus, all correct agents must agree on a common value, whereas faulty agents are exempted. Consequently, in a correctness proof for a particular protocol, it is common to verify $B_i\varphi$ only for correct agents. In effect, the condition being verified in such correctness analyses is

$$H_i\varphi := \mathit{correct}_i \to K_i(\mathit{correct}_i \to \varphi),$$

which we call the *hope* modality. Note that H_i is not localized for i because, by Theorem 10, the agent itself can never ascertain its own correctness. On the other hand, per Remark 11, the agent can in some cases learn its own faultiness. Assuming the agent is malfunctioning rather than malicious, this information can be used to implement self-correcting protocols or, at least, to minimize the effects of detected faults on the system as a whole. Therefore, exploiting negative introspection and factivity of K_i, we consider the modality *credence* defined by

$$Cr_i\varphi := \neg K_i \neg \mathit{correct}_i \wedge K_i(\mathit{correct}_i \to \varphi),$$

which is always localized, for protocol design.

Further, Def. 2 shows that our framework easily models agents whose faultiness is restricted in a particular way. For instance, the *send omissions failures* from [25], where an agent may fail to send some of the required messages, arbitrary *receiving failures*, where an agent may receive incorrect messages (or not receive correct ones), and the well-studied *crash failures* can easily be defined. It suffices to introduce restricted-fault propositions such as $\mathit{crashcorrect}_i$, $\mathit{sendcorrect}_i$, $\mathit{receivecorrect}_i$ and define high-level logical descriptions of the appropriate type of errors on top of it. E.g., replacing $K_i(\mathit{correct}_i \to \varphi)$ with $K_i(\mathit{receivecorrect}_i \to \varphi)$ if the truth of φ relies solely on correct communication shrinks the pool of situations ignored by the protocol to only those faults that do impede the agent's ability to ascertain φ.

The following basic relationships among the proposed modalities describing various preconditions immediately follow from the standard S5 properties of K_i:

Lemma 13. *For any formula φ, any agent i, the following formulas are valid in every interpreted system:*

$$\models K_i\varphi \to B_i\varphi \qquad \models Cr_i\varphi \to B_i\varphi \qquad \models B_i\varphi \to H_i\varphi$$
$$\models correct_i \to (H_i\varphi \to Cr_i\varphi) \qquad \models \neg correct_i \to H_i\varphi$$
$$\models K_i\varphi \to \varphi \qquad \models correct_i \to (B_i\varphi \to \varphi)$$
$$\models correct_i \to (Cr_i\varphi \to \varphi) \qquad \models correct_i \to (H_i\varphi \to \varphi)$$
$$\models B_i\varphi \to K_iB_i\varphi \qquad \models Cr_i\varphi \to K_iCr_i\varphi$$
$$\models K_icorrect_i \to (H_i\varphi \to K_i\varphi)$$

As follows from the preceding lemma, credence is stronger than belief, which is stronger than hope, with knowledge also being stronger than belief. However, for a correct agent, credence, belief, and hope all become equivalent, while knowledge generally remains stronger; for fault-free systems, this hierarchy collapses to the standard notion of knowledge. A faulty agent, however, automatically hopes for everything, making it unnecessary to check preconditions for faulty agents while verifying system correctness. At the same time, all four modalities are factive for correct agents (knowledge is factive for all agents), making them acceptable as precondition criteria modulo correctness. Finally, belief and credence satisfy the self-awareness condition that one should know one's own convictions (cf., e.g., [17]). On the other hand, hope, which represents an external view, does not generally satisfy $H_i\varphi \to K_iH_i\varphi$. A complete axiomatization of the hope modality, presented in [10], is obtained by adding to K45 the axioms

$$correct_i \to (H_i\varphi \to \varphi), \qquad \neg correct_i \to H_i\varphi, \qquad \text{and} \qquad H_icorrect_i.$$

7 Conclusions and Future Work

We presented a general framework for reasoning about knowledge in multi-agent message-passing systems with byzantine agents. Thanks to its modularity, it allows to model any timing and synchrony properties of agents and messages. We demonstrated the utility of our framework by successfully modeling the brain-in-a-vat scenario in a system of asynchronous agents, some of which are byzantine. Since this result implies that the knowledge of preconditions principle puts severe restrictions on allowable preconditions, we introduced weaker modalities, credence and hope, for the design and analysis of protocols respectively, which translate desired properties into actionable preconditions.

Part of our current work is devoted to further exploring these modalities, as well as their mutual relationships, and to the study of causality in fault-tolerant distributed systems, with the view of obtaining necessary conditions for iterated, coordinated, and simultaneous actions. Future work will be devoted to further incorporating protocols, which are currently specified outside our combined temporal–epistemic logic, into the logic itself, e.g., by using a suitable adaptation of dynamic epistemic logic.

A Appendix

This section is dedicated to proving the Brain-in-a-Vat Lemma. Before engaging with the proof, we flesh out necessary details of how our framework operates.

For a function $f\colon \Sigma \to \Theta$ and a set $X \subseteq \Sigma$ we use the following notation:

$$f(X) := \{f(x) \mid x \in X\} \subseteq \Theta.$$

For functions with multiple arguments, we allow ourselves to mix and match elements with sets of elements, e.g.,

$$\mathit{global}\,(i, \mathbb{T}, \mathit{Actions}_i) := \Bigl\{\mathit{global}\,(i,t,a) \Bigm| t \in \mathbb{T}, a \in \mathit{Actions}_i\Bigr\}.$$

As stated in Sect. 2, the function

$$\mathit{global}\colon \bigsqcup_{i \in \mathcal{A}} \Bigl(\{i\} \times \mathbb{T} \times \mathit{Actions}_i\Bigr) \longrightarrow \overline{\mathit{GActions}}$$

must be total and satisfy the following properties: for arbitrary $i, j \in \mathcal{A}$, and $t, t' \in \mathbb{T}$, and $a \in \mathit{Actions}_i$, and $b \in \mathit{Actions}_j$,

1. $\mathit{global}\,(i, \mathbb{T}, \mathit{Actions}_i) = \overline{\mathit{GActions}}_i$;
2. $\mathit{global}\,(i,t,a) \neq \mathit{global}\,(j,t',b)$ whenever $(i,t,a) \neq (j,t',b)$.

Thus, it is possible to define an inverse function on

$$\overline{\mathit{GHaps}} := \overline{\mathit{GEvents}} \sqcup \overline{\mathit{GActions}}.$$

Definition 14 (Function *local***).** *We use a function*

$$\mathit{local}\colon \overline{\mathit{GHaps}} \longrightarrow \mathit{Haps}$$

converting correct haps from the global format into the local ones for the respective agents in such a way that, for any $i \in \mathcal{A}$, $t \in \mathbb{T}$, and $a \in \mathit{Actions}_i$,

1. *$\mathit{local}\,\bigl(\overline{\mathit{GActions}}_i\bigr) = \mathit{Actions}_i$;*
2. *$\mathit{local}\,\bigl(\overline{\mathit{GEvents}}_i\bigr) = \mathit{Events}_i$; and*
3. *$\mathit{local}\Bigl(\mathit{global}\,(i,t,a)\Bigr) = a$.*

Recall that

$$\overline{\mathit{GEvents}}_i \cap \overline{\mathit{GEvents}}_j = \varnothing \qquad \text{for } i \neq j,$$

$$\mathit{BEvents}_i := \Bigl\{\mathit{fake}\,(i,E) \Bigm| E \in \overline{\mathit{GEvents}}_i\Bigr\} \sqcup \Bigl\{\mathit{fake}\,(i, A \mapsto A') \Bigm| A, A' \in \{\mathbf{noop}\} \sqcup \overline{\mathit{GActions}}_i\Bigr\},$$

$$\mathit{SysEvents}_i := \Bigl\{\mathit{go}(i), \mathit{sleep}\,(i), \mathit{hibernate}\,(i)\Bigr\}.$$

While for correct haps, *local* provides a translation to local format on a hap-by-hap basis, the same cannot be extended to all haps because system events from $\mathit{SysEvents}_i$ and byzantine actions $\mathit{fake}\,(i, A \mapsto \mathbf{noop})$ do not correspond to any local hap: they are not recorded in i's history. Thus, the *localization function* σ is defined on sets of global haps:

Definition 15 (Localization). *We define a* localization function

$$\sigma\colon \wp(\mathit{GHaps}) \longrightarrow \wp(\mathit{Haps})$$

as follows:

$$\sigma(X) \quad := \quad \mathit{local}\Bigl(\bigl(X \cap \overline{\mathit{GHaps}}\bigr) \quad \cup \quad \bigl\{E \bigm| (\exists i)\,\mathit{fake}\,(i,E) \in X\bigr\} \quad \cup \quad \bigl\{A' \neq \boldsymbol{noop} \bigm| (\exists i)(\exists A)\,\mathit{fake}\,(i, A \mapsto A') \in X\bigr\}\Bigr).$$

Thus, as intended, for any $E \in \overline{\mathit{GEvents}}_i$, the local record left by $\mathit{fake}\,(i,E)$ for agent i is the same as the record of E, whereas for any $A' \in \overline{\mathit{GActions}}_i$ and any $A \in \{\mathbf{noop}\} \sqcup \overline{\mathit{GActions}}_i$, the local record of $\mathit{fake}\,(i, A \mapsto A')$ for i is the same as that of A', whichever action A was taken in reality.

Definition 16 (System update). *We abbreviate*

$$X := X_\epsilon, X_1, \dots, X_n \qquad \text{and} \qquad X_{\epsilon_i} := X_\epsilon \cap GEvents_i$$

for a tuple of performed events $X_\epsilon \subseteq GEvents$ and actions $X_i \subseteq \overline{GActions}_i$ for each $i \in \mathcal{A}$. Given a global state

$$r(t) = \bigl(r_\epsilon(t), r_1(t), \dots, r_n(t)\bigr) \in \mathscr{G},$$

we define agent i's update function

$$update_i \colon \mathscr{L}_i \times \wp\left(\overline{GActions}_i\right) \times \wp\left(GEvents\right) \to \mathscr{L}_i$$

that outputs a new local state from $\mathscr{L}_i$ based on i's actions X_i and events X_ϵ:

$$update_i\left(r_i(t), X_i, X_\epsilon\right) := \begin{cases} r_i(t) & \text{if } \sigma(X_{\epsilon_i}) = \varnothing \text{ and } X_\epsilon \cap \bigl\{go(i), sleep(i)\bigr\} = \varnothing, \\ \Bigl[\sigma\bigl(X_{\epsilon_i} \sqcup X_i\bigr)\Bigr] : r_i(t) & \text{otherwise} \end{cases}$$

(*note that, in transitional runs, $update_i$ is always used after the action $filter_i$; thus, in the absence of $go(i)$, it is always the case that $X_i = \varnothing$*). *Similarly, the* environment's state update function

$$update_\epsilon \colon \mathscr{L}_\epsilon \times \wp\left(GEvents\right) \times \wp\left(\overline{GActions}_1\right) \times \dots \times \wp\left(\overline{GActions}_n\right) \to \mathscr{L}_\epsilon$$

outputs a new state of the environment based on events X_ϵ and all actions X_i:

$$update_\epsilon\left(r_\epsilon(t), X\right) := \left(X_\epsilon \sqcup X_1 \sqcup \dots \sqcup X_n\right) : r_\epsilon(t).$$

Summarizing,

$$update\left(r(t), X\right) := \Bigl(update_\epsilon\left(r_\epsilon(t), X\right), update_1\left(r_1(t), X_1, X_\epsilon\right), \dots, update_n\left(r_n(t), X_n, X_\epsilon\right)\Bigr).$$

The following properties directly follow from Def. 7 of the i-intervention $Fake_i^t$:

Lemma 17. *Let $t \in \mathbb{T}$ and r be an arbitrary run. Then*

1. $\mathfrak{a}Fake_i^t(r) = \varnothing$, *i.e., $Fake_i^t$ removes all actions.*
2. $go(i) \notin \mathfrak{e}Fake_i^t(r)$, *i.e., $Fake_i^t$ never lets agent i act.*
3. $\sigma\Bigl(\mathfrak{a}Fake_i^t(r) \sqcup \mathfrak{e}Fake_i^t(r)\Bigr) = \sigma\Bigl(\mathfrak{e}Fake_i^t(r)\Bigr) = \sigma\Bigl(\beta_i^t(r) \sqcup \beta_{\epsilon_i}^t(r)\Bigr)$.
4. $r_i(t+1) \neq r_i(t)$ *iff* $\mathfrak{e}Fake_i^t(r) \cap \bigl\{go(i), sleep(i)\bigr\} \neq \varnothing$.

The last two properties mean that from i's local perspective, the intervention is imperceptible (also when agent i was unaware of the passing round in the given run).

Proof (of ***Brain-in-a-Vat Lemma 8****).* Let $r' \in R\left(\tau_{f,P_\epsilon,P}, r, adj\right)$. Prop. 6 follows from the definitions of *CFreeze* and $BFreeze_j$. Prop. 7 follows from (1). Prop. 8 follows from Lemma 17.1 for i and from the definitions of *CFreeze* and $BFreeze_j$ for $j \neq i$. Prop. 5 follows from (1) for i and from the definition of $BFreeze_j$ for $j \in Byz$. Props. 2–4 depend solely on rounds from ½ to $t -$ ½ of r', whereas the transitionality of r' for Prop. 1 from round $t +$ ½ onward directly follows from Def. 6. We now show Props. 1–4 for $m \leq t$ by induction on m.

Base: $m = 0$. Props. 3–4 and transitionality for Prop. 1 are trivial. Prop. 2 follows from Def. 6.

Step from m to $m+1$. We prove Prop. 1 based on the gullibility of i and delayability (and fallibility) of all other $j \neq i$. In order to show that $r(m)\ \tau_{f,P_\epsilon,P}\ r(m+1)$, we need to demonstrate that the β-sets prescribed by adj can be obtained in a regular round. Since the adversary's choice of actions $\alpha_j^m(r)$ for all $j \in \mathcal{A}$ is immaterial due to the absence of $go(j)$ (by Prop. 7 for i and Prop. 6 for other $j \neq i$), we concentrate on ensuring the adversary can choose suitable α-sets of events. Consider $\alpha^m_\epsilon(r) \in P_\epsilon(m)$ from the original run r. The set $\alpha^m_\epsilon(r)$ is coherent because r is transitional. By the delayability of all $j \neq i$,

$$\alpha^m_{\epsilon_i}(r) := \alpha^m_\epsilon(r) \cap \mathit{GEvents}_i = \alpha^m_\epsilon(r) \setminus \bigsqcup_{j \neq i} \mathit{GEvents}_j \in P_\epsilon(m).$$

Note that for any $Z \subseteq \mathit{FEvents}_i$,

$$\bigl(\alpha^m_{\epsilon_i}(r) \setminus \mathit{GEvents}_i\bigr) \sqcup Z = \varnothing \sqcup Z = Z$$

because $\alpha^m_{\epsilon_i}(r) \subseteq \mathit{GEvents}_i$. Thus, by the gullibility of i,

$$\begin{aligned}\alpha^m_{\epsilon_i}(r') := \bigl\{\mathit{fail}(i)\bigr\} \quad \cup \quad \bigl(\beta^m_\epsilon(r) \cap \mathit{BEvents}_i\bigr) \quad \cup \quad \Bigl\{\mathit{fake}(i, E) \Bigm| E \in \beta^m_\epsilon(r) \cap \overline{\mathit{GEvents}_i}\Bigr\} \quad \cup \\ \Bigl\{\mathit{fake}(i, \mathbf{noop} \mapsto A) \Bigm| A \in \beta^m_i(r)\Bigr\} \quad \sqcup \quad \Bigl\{\mathit{sleep}(i) \Bigm| r_i(t+1) \neq r_i(t)\Bigr\} \in P_\epsilon(m)\end{aligned}$$

(note that this set is coherent because it contains no correct events and neither $go(i)$ nor $\mathit{hibernate}(i)$). Finally, by the fallibility of all agents $j \in \mathit{Byz}$,

$$\alpha^m_\epsilon(r') := \alpha^m_{\epsilon_i}(r') \sqcup \Bigl\{\mathit{fail}(j) \Bigm| j \in \mathit{Byz}\Bigr\} \in P_\epsilon(m).$$

This $\alpha^m_\epsilon(r')$ is coherent and unaffected by filtering (there are no correct receives in $\alpha^m_\epsilon(r')$ to be filtered out, and only at most f agents from $\{i\} \sqcup \mathit{Byz}$ become byzantine).

It remains to show that filtering turns these sets $\alpha^m_\epsilon(r'), \alpha^m_1(r'), \dots, \alpha^m_n(r')$ into the exact β-sets prescribed by the adjustment adj. Let us abbreviate:

$$\begin{aligned}\varUpsilon \quad &:= \quad \mathit{filter}_\epsilon\Bigl(r'(m), \quad \alpha^m_\epsilon(r'), \quad \alpha^m_1(r'), \quad \dots, \quad \alpha^m_n(r')\Bigr) \quad = \quad \alpha^m_\epsilon(r'), \\ \varXi_j \quad &:= \quad \mathit{filter}_j\Bigl(\alpha^m_1(r'), \quad \dots, \quad \alpha^m_n(r'), \quad \varUpsilon\Bigr).\end{aligned}$$

Our goal is to show that

$$\varUpsilon_j := \varUpsilon \cap \mathit{GEvents}_j = \beta^m_{\epsilon_j}(r') \qquad \text{and} \qquad \varXi_j = \beta^m_j(r')$$

for each $j \in \mathcal{A}$. After the filtering phase, for our i and $j \neq i$, we have the following:

$$\begin{aligned}\varUpsilon_i \quad &= \quad \alpha^m_\epsilon(r') \cap \mathit{GEvents}_i \quad = \quad \alpha^m_{\epsilon_i}(r') \quad = \quad \beta^m_{\epsilon_i}(r'), \\ \varUpsilon_j \quad &= \quad \alpha^m_\epsilon(r') \cap \mathit{GEvents}_j \quad = \quad \begin{cases}\varnothing & \text{if } \rho^m_j = \mathit{CFreeze}, \\ \bigl\{\mathit{fail}(j)\bigr\} & \text{if } \rho^m_j = \mathit{BFreeze}_j,\end{cases}\end{aligned}$$

the latter being exactly $\beta^m_{\epsilon_j}(r')$. Since $go(j) \notin \varUpsilon$ for any $j \in \mathcal{A}$, we also have that

$$\varXi_j \quad = \quad \varnothing \quad = \quad \beta^m_j(r')$$

for all $j \in \mathcal{A}$. This completes the induction step for Prop. 1.

For Prop. 2, the induction step follows from Lemma 17.3–4. We have:

- if $\sigma\Bigl(\beta^m_{\epsilon_i}(r)\Bigr) \neq \varnothing$, then

$$r_i(m+1) \quad = \quad \sigma\Bigl(\beta^m_i(r) \sqcup \beta^m_{\epsilon_i}(r)\Bigr) : r_i(m) \quad = \quad \sigma\Bigl(\beta^m_i(r) \sqcup \beta^m_{\epsilon_i}(r)\Bigr) : r'_i(m)$$

by IH. It remains to use Lemma 17.3 to see that the last expression is the same as

$$\sigma\Bigl(\beta^m_i(r') \sqcup \beta^m_{\epsilon_i}(r')\Bigr) : r'_i(m) \quad = \quad r'_i(m+1).$$

- if $\sigma\Big(\beta^m_{\epsilon_i}(r)\Big) = \varnothing$ but $r_i(m+1) \neq r_i(m)$, then

$$r_i(m+1) \quad = \quad \sigma\Big(\beta^m_i(r) \sqcup \beta^m_{\epsilon_i}(r)\Big) : r_i(m) \quad = \quad \sigma\Big(\beta^m_i(r)\Big) : r_i(m) \quad = \quad \sigma\Big(\beta^m_i(r)\Big) : r'_i(m)$$

  by IH. By Lemma 17.4, we also have $r'_i(m+1) \neq r'_i(m)$. Thus, this case can be concluded by using Lemma 17.3 as it was done in the previous case.
- if $\sigma\Big(\beta^m_{\epsilon_i}(r)\Big) = \varnothing$ and $r_i(m+1) = r_i(m)$, then $r'_i(m+1) = r'_i(m)$ by Lemma 17.3–4 (note that $go(i) \notin \beta^m_\epsilon(r)$, meaning that $\beta^m_i(r) = \varnothing$). Using IH, we now immediately get

$$r'_i(m+1) \quad = \quad r'_i(m) \quad = \quad r_i(m) \quad = \quad r_i(m+1)\,.$$

This completes the proof of the induction step for Prop. 2.

For Props. 3–4, the induction step is even simpler. Since, for any $j \neq i$,

$$\beta^m_{\epsilon_j}(r') \quad \subseteq \quad \big\{\mathit{fail}(j)\big\}$$

by Prop. 6, it follows that

$$r'_j(m+1) \quad = \quad r'_j(m) \quad = \quad r'_j(0)$$

by IH. Similarly, $\beta^m_{\epsilon_j}(r') = \varnothing$ by Prop. 6 for $j \notin \{i\} \sqcup Byz$. Thus, being correct at m by IH, such agents j remain correct after round $m + \frac{1}{2}$. □

Acknowledgments. We are grateful to Hans van Ditmarsch and Yoram Moses for extensive helpful comments on earlier versions of this paper. We also thank the anonymous reviewers for their comments and suggestions on related research.

References

1. I. Ben-Zvi and Y. Moses. Agent-time epistemics and coordination. In K. Lodaya, editor, *ICLA 2013*, volume 7750 of *LNCS*, pages 97–108. Springer, 2013. doi:10.1007/978-3-642-36039-8_9.
2. I. Ben-Zvi and Y. Moses. Beyond Lamport's *Happened-before*: On time bounds and the ordering of events in distributed systems. *Journal of the ACM*, 61(2:13), 2014. doi:10.1145/2542181.
3. A. Castañeda, Y. A. Gonczarowski, and Y. Moses. Unbeatable consensus. In F. Kuhn, editor, *DISC 2014*, volume 8784 of *LNCS*, pages 91–106. Springer, 2014. doi:10.1007/978-3-662-45174-8_7.
4. H. van Ditmarsch. Dynamics of lying. *Synthese*, 191(5):745–777, 2014. doi:10.1007/s11229-013-0275-3.
5. C. Dwork and Y. Moses. Knowledge and common knowledge in a Byzantine environment: Crash failures. *Information and Computation*, 88(2):156–186, 1990. doi:10.1016/0890-5401(90)90014-9.
6. J. Ezekiel and A. Lomuscio. Combining fault injection and model checking to verify fault tolerance, recoverability, and diagnosability in multi-agent systems. *Information and Computation*, 254(2):167–194, 2017. doi:10.1016/j.ic.2016.10.007.
7. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.
8. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. Common knowledge revisited. *Annals of Pure and Applied Logic*, 96:89–105, 1999. doi:10.1016/S0168-0072(98)00033-5.
9. A. Fedoruk and R. Deters. Improving fault-tolerance by replicating agents. In *AAMAS '02*, pages 737–744. ACM, 2002. doi:10.1145/544862.544917.
10. K. Fruzsa. Hope for epistemic reasoning with faulty agents! In *Proceedings of ESSLLI 2019 Student Session*, 2019. (To appear).
11. Y. A. Gonczarowski and Y. Moses. Timely common knowledge: Characterising asymmetric distributed coordination via vectorial fixed points. In B. C. Schipper, editor, *TARK XIV*, pages 79–93, 2013. Available from: https://arxiv.org/pdf/1310.6414.pdf.
12. G. Goren and Y. Moses. Silence. In *PODC '18*, pages 285–294. ACM, 2018. doi:10.1145/3212734.3212768.
13. J. Y. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3):549–587, 1990. doi:10.1145/79147.79161.

14. J. Y. Halpern, Y. Moses, and O. Waarts. A characterization of eventual Byzantine agreement. *SIAM Journal on Computing*, 31(3):838–865, 2001. doi:10.1137/S0097539798340217.
15. J. Hintikka. *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Cornell University Press, 1962.
16. M. Kalech and G. A. Kaminka. On the design of coordination diagnosis algorithms for teams of situated agents. *Artificial Intelligence*, 171(8–9):491–513, 2007. doi:10.1016/j.artint.2007.03.005.
17. S. Kraus and D. Lehmann. Knowledge, belief and time. *Theoretical Computer Science*, 58:155–174, 1988. doi:10.1016/0304-3975(88)90024-2.
18. R. Kuznets, L. Prosperi, U. Schmid, K. Fruzsa, and L. Gréaux. Knowledge in Byzantine message-passing systems I: Framework and the causal cone. Technical Report TUW-260549, TU Wien, 2019. Available from: https://publik.tuwien.ac.at/files/publik_260549.pdf.
19. L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982. doi:10.1145/357172.357176.
20. A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: A model checker for the verification of multi-agent systems. In A. Bouajjani and O. Maler, editors, *CAV 2009*, volume 5643 of *LNCS*, pages 682–688. Springer, 2009. doi:10.1007/978-3-642-02658-4_55.
21. M. McKinsey. Skepticism and content externalism. In *Stanford Encyclopedia of Philosophy*. 2018. Available from: https://plato.stanford.edu/entries/skepticism-content-externalism/.
22. R. Michel. A categorical approach to distributed systems, expressibility and knowledge. In P. Rudnicki, editor, *PODS '89*, pages 129–143. ACM, 1989. doi:10.1145/72981.72990.
23. Y. Moses. Relating knowledge and coordinated action: The knowledge of preconditions principle. In R. Ramanujam, editor, *TARK 2015*, pages 231–245, 2015. doi:10.4204/EPTCS.215.17.
24. Y. Moses and Y. Shoham. Belief as defeasible knowledge. *Artificial Intelligence*, 64(2):299–321, 1993. doi:10.1016/0004-3702(93)90107-M.
25. Y. Moses and M. R. Tuttle. Programming simultaneous actions using common knowledge. *Algorithmica*, 3:121–169, 1988. doi:10.1007/BF01762112.
26. A. Pessin and S. Goldberg, editors. *The Twin Earth Chronicles: Twenty Years of Reflection on Hilary Putnam's the "Meaning of Meaning"*. M. E. Sharpe, 1996.
27. G. Taubenfeld. *Distributed Computing Pearls*. Morgan & Claypool Publishers, 2018. doi:10.2200/S00845ED1V01Y201804DCT014.
28. R. L. Trask. *Mind the Gaffe: The Penguin Guide to Common Errors in English*. Penguin Books, 2001.