Declarative distributed protocols using three-valued modal logic and semitopologies

Murdoch J. Gabbay

Paris Automata and Concurrency Theory Seminar,
19 February 2026

Decentralised algorithms as axiom-systems

Decentralised algorithms are algorithms that run across a system of participants, without central control.

I may use 'decentralised' and 'distributed' synonymously, but they're not: distributed algorithms may still be centrally controlled.

Decentralised/distributed algorithms are hard: hard to design; hard to understand; hard to verify.

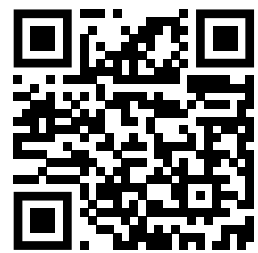
Academic examples of distributed algorithms include Paxos and Bracha Broadcast. I introduce a method of studying these algorithms *declaratively* as *axiomatic theories*, thus capturing their logical structure in a new, high-level way.

I have found this personally helpful for understanding. It has also been applied to detect fatal errors in proposed industrial protocols (i.e. this works).

Bracha Broadcast

By the end of this talk, I hope you'll have a flavour of what *declarative Bracha Broadcast* looks like.

A journal paper is submitted:
Declarative distributed broadcast using three-valued modal logic and semitopologies
https://arxiv.org/abs/2512.21137.



An axiomatisation of Paxos is also submitted:
A declarative approach to specifying distributed algorithms using three-valued modal logic
https://arxiv.org/abs/2502.00892.



A toy protocol

Fix a pair $(\mathsf{P}, \mathsf{Open})$ of a set of participants P and a set of quorums $\mathsf{Open} \subseteq \mathit{powerset}(\mathsf{P})$. Secretly, this is a *semitopology* — we'll come back to that.

Toy is as follows:

1. *Propose.* Any participant p may broadcast a `propose` message to all participants.
2. *Accept.* If a participant receives a `propose` message, then it responds with `accept`.
3. *Decide.* If the participant p, having broadcast its `propose` message, receives an `accept` message, then it declares `decide`.

(I made this up; it's loosely based on a 1-value version of Paxos.)

Background 1. Modal logic

We will model **Toy** using a modal logic, meaning that the truth-value of a predicate depends on a *possible world*, which consists of a time and/or a place. For example:

"It's raining" is a predicate.

"It's raining *now*" and "It's raining *here*" are modal predicates, because validity depends on the time and place.

"It's raining *somewhere*" is modal: this is typically written as $\Diamond\texttt{raining}$.

"It's raining *everywhere*" is modal: this is typically written as $\Box\texttt{raining}$.

For **Toy**, the possible worlds are participants $p \in \mathrm{P}$ (no time; just places).

More formally: the truth-value $[\![\phi]\!]_w$ is a function of ϕ and w. Write $w \vDash \phi$ for '$[\![\phi]\!]_w$ is a valid truth-value'.

Background 2. Three-valued (modal) logic

We will use *three* truth-values:

1. $\mathbf{t}$ ('true'). If $[\phi]_w = \mathbf{t}$ then call ϕ *true* at w.
2. $\mathbf{f}$ ('false'). If $[\phi]_w = \mathbf{f}$ then call ϕ *false* at w.
3. $\mathbf{b}$ ('byzantine'). If $[\phi]_w = \mathbf{b}$ then call ϕ *byzantine* at w.

Call $x \in \{\mathbf{t}\}$ true, $x \in \{\mathbf{t}, \mathbf{f}\}$ correct, and $x \in \{\mathbf{t}, \mathbf{b}\}$ valid.

For x a truth-value, define modalities $\mathtt{T}$, $\mathbb{F}$, and $\mathbb{B}$ by:

- $\mathtt{T}\, x = \mathbf{t}$ when $x \in \{\mathbf{t}\}$ and $\mathtt{T}\, x = \mathbf{f}$ otherwise;
- $\mathbb{F}\, x = \mathbf{t}$ when $x \in \{\mathbf{t}, \mathbf{f}\}$ and $\mathbb{F}\, x = \mathbf{f}$ otherwise; and
- $\mathbb{B}\, x = \mathbf{t}$ when $x \in \{\mathbf{t}, \mathbf{b}\}$ and $\mathbb{B}\, x = \mathbf{f}$ otherwise.

Write $w \vDash \phi$ to mean $[\phi]_w \in \{\mathbf{t}, \mathbf{b}\}$. Write $\vDash \phi$ to mean $\forall w. w \vDash \phi$.

Some connectives (for reference)

Weak implication

$p \to q$	**t**	**b**	**f**
t	**t**	**b**	**f**
b	**t**	**b**	**b**
f	**t**	**t**	**t**

Strong implication

$p \twoheadrightarrow q$	**t**	**b**	**f**
t	**t**	**f**	**f**
b	**t**	**b**	**b**
f	**t**	**t**	**t**

$\neg p$	
t	**f**
b	**b**
f	**t**

$\mathbb{T}\, p$	
t	**t**
b	**f**
f	**f**

$\mathbb{B}\, p$	
t	**t**
b	**t**
f	**f**

$\mathbb{F}\, p$	
t	**t**
b	**f**
f	**t**

Intuitively:

- $\models p \twoheadrightarrow q$ means $p = \mathbf{t}$ implies $q = \mathbf{t}$.
- $\models p \to q$ means $p = \mathbf{t}$ implies $q \in \{\mathbf{t}, \mathbf{b}\}$.

Recall **Toy**

We have participants P and quorums Open $\subseteq$ *powerset*(P).

1. *Propose.* Any participant p may broadcast a `propose` message to all participants.
2. *Accept.* If a participant receives a `propose` message, then it responds with `accept`.
3. *Decide.* If the participant p, having broadcast its `propose` message, receives an `accept` message, then it declares `decide`.

Let's make this declarative. We propose *three* theories (= set of axioms), corresponding to failure assumptions as follows:

1. all participants honest (non-byzantine) and live (uncrashed),
2. all participants honest & a quorum of live participants,
3. a contraquorum *(defined in later slide)* of live honest participants.

Theory 1. Honest & live

Assume atomic propositions $\texttt{propose}$, $\texttt{accept}$, and $\texttt{decide}$.

Backward rules	
(**SimpAccept**?)	$\texttt{accept} \twoheadrightarrow \Diamond\texttt{propose}$
(**SimpDecide**?)	$\texttt{decide} \twoheadrightarrow (\texttt{propose} \wedge \Diamond\texttt{accept})$
Forward rules	
(**SimpAccept**!)	$(\Diamond\texttt{propose}) \twoheadrightarrow \texttt{accept}$
(**SimpDecide**!)	$(\texttt{propose} \wedge \Diamond\texttt{accept}) \twoheadrightarrow \texttt{decide}$
Other rules	
(**Correct**)	$\mathbb{F}\,[\texttt{propose}, \texttt{accept}, \texttt{decide}]$

$p \vDash \phi \twoheadrightarrow \phi'$ means $p \vDash \mathtt{T}\,\phi \Rightarrow p \vDash \mathtt{T}\,\phi'$
(if ϕ is true then ϕ' is true; material implication).

$p \vDash \Diamond\phi$ means $\exists p'.p' \vDash \phi$. ($\phi$ is valid somewhere).

A model (= assignment of truth-values to atomic propositions) satisfies a theory $\mathcal{T}$ when $\vDash$ (**Ax**) in that model, for every (**Ax**) $\in \mathcal{T}$.

Theory 2. Quorum of live participants

Backward rules	
(**SimpAccept**?)	$\texttt{accept} \twoheadrightarrow \Diamond\texttt{propose}$
(**SimpDecide**?)	$\texttt{decide} \twoheadrightarrow (\texttt{propose} \wedge \Diamond\texttt{accept})$
Forward rules	
(**SimpAccept**!)	$(\Diamond\texttt{propose}) \rightarrow \texttt{accept}$
(**SimpDecide**!)	$(\texttt{propose} \wedge \Diamond\texttt{accept}) \rightarrow \texttt{decide}$
Other rules	
(**Correct**)	$\boxdot\mathbb{F}\,[\texttt{propose}, \texttt{accept}, \texttt{decide}]$

$p \vDash \phi \twoheadrightarrow \phi'$ means $p \vDash \mathtt{T}\,\phi \Rightarrow p \vDash \mathtt{T}\,\phi'$

$p \vDash \phi \rightarrow \phi'$ means $p \vDash \mathtt{T}\,\phi \Rightarrow p \vDash \mathbb{B}\,\phi'$

$p \vDash \Diamond\phi$ means $\exists p'.p' \vDash \phi$.

$p \vDash \boxdot\phi$ means $\exists O \in \mathrm{Open}.\forall p' \in O.p' \vDash \phi$.
(ϕ valid on some quorum).

Theory 3. Contraquorum of honest participants

Backward rules	
(**SimpAccept**?)	$\texttt{accept} \to \Diamond \texttt{propose}$
(**SimpDecide**?)	$\texttt{decide} \to (\texttt{propose} \wedge \Diamond \texttt{accept})$
Forward rules	
(**SimpAccept**!)	$(\Diamond \texttt{propose}) \to \texttt{accept}$
(**SimpDecide**!)	$(\texttt{propose} \wedge \Diamond \texttt{accept}) \to \texttt{decide}$
Other rules	
(**Correct**)	$\boxdot \mathbb{F}\,[\texttt{propose}, \texttt{accept}, \texttt{decide}]$

$p \vDash \phi \to \phi'$ means $p \vDash \mathtt{T}\,\phi \Rightarrow p \vDash \mathbb{B}\,\phi'$

$p \vDash \Diamond \phi$ means $\exists p'.p' \vDash \phi$.

$p \vDash \boxdot \phi$ means $\exists O \in \mathrm{Open}.\forall p' \in O.p' \vDash \phi$.

Background: quorums, coquorums, & contraquorums

Recall the semitopology $(\mathrm{P}, \mathrm{Open})$ ($\approx$ quorum system) .

- Call $O \in \mathrm{Open}$ an open or a quorum.
- Call $C \subseteq \mathrm{P}$ closed or a coquorum when $C = \mathrm{P} \setminus O$ for some $O \in \mathrm{Open}$.
- Call $D \subseteq \mathrm{P}$ dense or a contraquorum when $D \cap O \neq \varnothing$ for every nonempty $O \in \mathrm{Open}$ (also called a *blocking set*). This is exactly the notion of *dense set* from topology.

In the case that $\#\mathrm{P} = 3f + 1$ and $O \in \mathrm{Open}$ when O is empty or $\#O \geq 2f + 1$,

- $C \subseteq \mathrm{P}$ is closed precisely when $C = \mathrm{P}$ or $\#C \leq f$, and
- $D \subseteq \mathrm{P}$ is dense precisely when $\#D \geq f + 1$.

Background: quorums, coquorums, & contraquorums

Have you seen the papers on distributed systems that talk about

- taking a set of participants having size $3f + 1$; and then
- quorums are sets of size at least $2f + 1$;
- failure sets have size at most f; and
- blocking sets have size at least $f + 1$, and so on?

These are *topological notions*, corresponding to open sets, closed sets, and dense sets respectively.

They're doing (semi)topology.

Bracha Broadcast (high-level view)

1. A designated sender participant *broadcasts* v to all processes. (We assume all messages arrive.)
2. If a participant receives a broadcast v message from the sender — we assume the sender's signature can't be forged — it sends an *echo* v to all processes.
   Each participant will echo at most *once*, so if it receives two broadcast messages with different values from a (byzantine) sender, it will only echo one of them.
3. If a participant receives a quorum of echo messages for a value v, or a contraquorum of ready messages for a value v, then it sends messages to all processes declaring itself *ready* with v.
4. If a process receives a quorum of ready messages for a value v, then the participant *delivers* v.

Declarative Bracha Broadcast

Backward rules

(**BrDeliver**?) $\texttt{dlvr}(v) \rightarrow \boxdot\texttt{ready}(v)$

(**BrReady**?) $\texttt{ready}(v) \rightarrow \boxdot\texttt{echo}(v)$

(**BrEcho**?) $\texttt{echo}(v) \rightarrow \texttt{brdcst}(v)$

Forward rules

(**BrDeliver**!) $\boxdot\texttt{ready}(v) \rightarrow \texttt{dlvr}(v)$

(**BrReady**!) $\boxdot\texttt{echo}(v) \rightarrow \texttt{ready}(v)$

(**BrEcho**!) $\Diamond\texttt{brdcst}(v) \rightarrow \texttt{echo}(v)$

(**BrReady**!!) $\diamondsuit\texttt{ready}(v) \rightarrow \texttt{ready}(v)$

Other rules

(**BrEcho01**) $\exists_{\mathbf{01}} v.\texttt{echo}(v)$

(**BrBroadcast1**) $\exists_{\mathbf{1}} v.\Diamond\texttt{brdcst}(v)$

(**BrCorrect**) $\boxdot\mathbb{F}\,[\texttt{ready}, \texttt{echo}, \texttt{brdcst}]$

(**BrCorrect**′) $\mathbb{F}\,[\texttt{P}] \vee \texttt{B}\,[\texttt{P}] \quad (\texttt{P} \in \{\texttt{ready}, \texttt{echo}\})$

(**BrCorrect**″) $\Box\,\mathbb{F}\,[\texttt{brdcst}] \vee \Box\,\texttt{B}\,[\texttt{brdcst}]$

Some notation

$\exists_{\mathbf{01}} v.\phi(v)$ is shorthand for $\forall v, v'.\phi(v) \to \phi(v') \to v = v'$.

- This means $\phi(v)$ is $\mathbf{t}$ for *at most* one v.
- $\phi(v)$ can be $\mathbf{f}$ or $\mathbf{b}$ for as many v as it wants!
- In particular, $\exists_{\mathbf{01}} v.\mathbf{b}$ is valid.

$\exists_{\mathbf{1}} v.\phi(v)$ is shorthand for $\exists_{\mathbf{01}} v.\phi(v) \wedge \exists v.\phi$.

- This means $\phi(v)$ is $\mathbf{t}$ for *at most* one v, and is $\mathbb{B}$ for *at least* one v.
- In particular, $\exists_{\mathbf{1}} v.\mathbf{b}$ is valid (specifically, $[\exists_{\mathbf{1}} v.\mathbf{b}]_w = \mathbf{b}$ always).

Correctness properties (informal; from literature)

These correctness properties for Bracha Broadcast are from pages 112 and 117 of *Introduction to reliable and secure distributed programming* (2nd ed):

1. Validity: If a correct process broadcasts some value v, then every correct process delivers v.
   The meaning of 'Correct' is not made immediately formal in the source citation.
2. No duplication: Every correct process delivers at most one value.
3. Integrity: If some process delivers a message v with sender p, and p is correct, then v was broadcast by p.
4. Consistency: If one correct process delivers v and another correct process delivers v', then $v = v'$.
5. Totality: If some correct process delivers v, then every correct process delivers v.

3-twined, axiomatically

Recall that a semitopology consists of a pair $(\mathrm{P}, \mathrm{Open})$ of a set of points P and a set of open sets $\mathrm{Open} \subseteq \mathit{pow}(\mathrm{P})$.

Call a semitoplogy 3-twined when any three nonempty open sets intersect (also called Q^3; see e.g. Definition 2 here).

Lemma: The following are equivalent [Lemma 5.3.1(2)]:

1. $(\mathrm{P}, \mathrm{Open})$ is 3-twined.
2. $\vDash (\boxdot\phi \wedge \boxdot\psi \wedge \boxdot\chi) \Rightarrow \Diamond(\phi \wedge \psi \wedge \chi)$ (for any ϕ, ψ, and χ).
3. $\vDash (\boxdot\phi \wedge \boxdot\psi) \Rightarrow \diamondsuit(\phi \wedge \psi)$.
4. $\vDash (\boxdot \mathbb{F}\,\phi \wedge \boxdot\phi) \Rightarrow \mathtt{T}\,\diamondsuit\phi$. *Correct on a quorum and valid on a quorum implies true on a contraquorum!*

$p \vDash \boxdot\phi$ means $\exists O \in \mathrm{Open}. \forall p' \in O. p' \vDash \phi$.

$p \vDash \phi \Rightarrow \phi'$ means $p \vDash \mathbb{B}\,\phi \Rightarrow p \vDash \mathbb{B}\,\phi'$
(if ϕ is valid then ϕ' is valid).

Correctness properties (in the logic)

Theorem: Any model of Declarative Bracha Broadcast over a 3-twined semitopology satisfies:

Validity:	$\models \Diamond \mathtt{brdcst}(v) \rightarrow \Box \mathtt{dlvr}(v)$
No duplication:	$\models \exists_{\mathbf{01}} v.\mathtt{dlvr}(v)$
Integrity:	$\models \mathtt{dlvr}(v) \rightarrow \Diamond \mathtt{brdcst}(v)$
Consistency:	$\models \exists_{\mathbf{01}} v.\Diamond \mathtt{dlvr}(v)$
Totality:	$\models \Diamond \mathtt{dlvr}(v) \rightarrow \Box \mathtt{dlvr}(v)$

- I don't explicitly represent messages! There are modal predicates evaluating to truth-values.
- p in the informal integrity property corresponds to $\Diamond$ in (**Integrity**) above.
- Asymmetry between (**Validity**) and (**Integrity**).
- (**BrBroadcast01**).

Conclusions

We can present distributed algorithms as axiomatic theories.

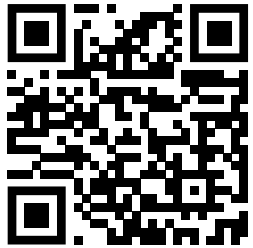
Paragraphs of English text get distilled down to *precise axiomatic assertions*. Difficult reasoning gets reduced to, or at least formalised as, formal logical reasoning.

It's a powerful technique. Axiomatic reasoning is not new, but it's not been applied in this way.

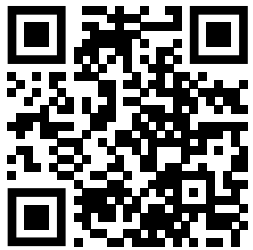
. . . and it works! I find it indispensable for my own understanding, and it's already helped to catch real bugs.

References

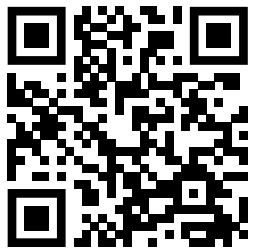
1. Declarative Bracha Broadcast (submitted): https://arxiv.org/abs/2512.21137.



2. Declarative Paxos journal paper (submitted): https://arxiv.org/abs/2502.00892.



3. Semitopologies journal paper (published): https://doi.org/10.1093/logcom/exae050.



4. Semitopologies book (published): collegepublications.co.uk/logic/?00056.

