

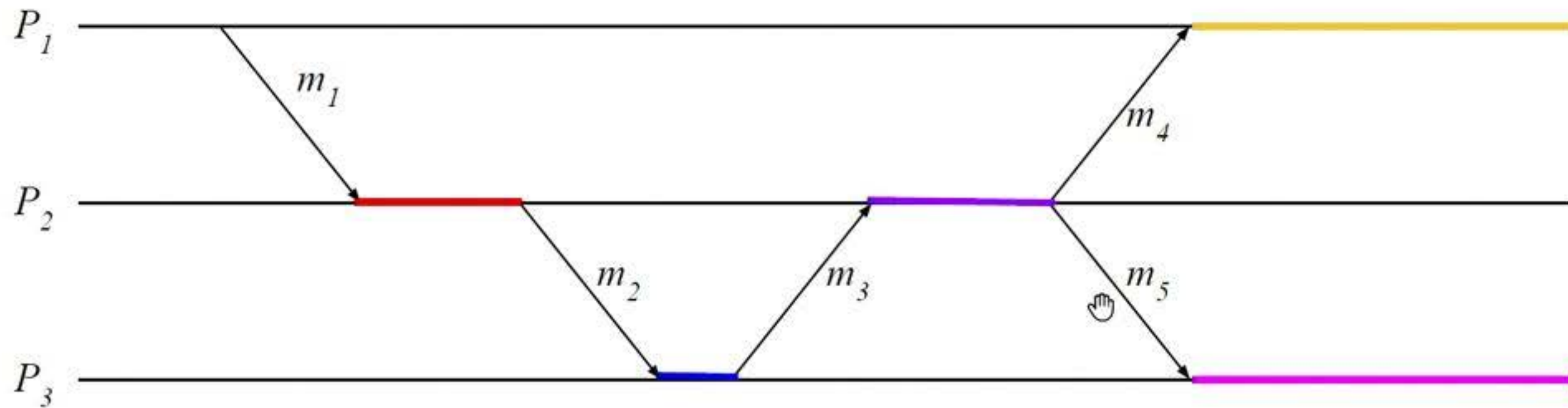
# Specification and Verification of Multi-Paxos

---

Saksham Chand,  
 Stony Brook University  
11/15/2019

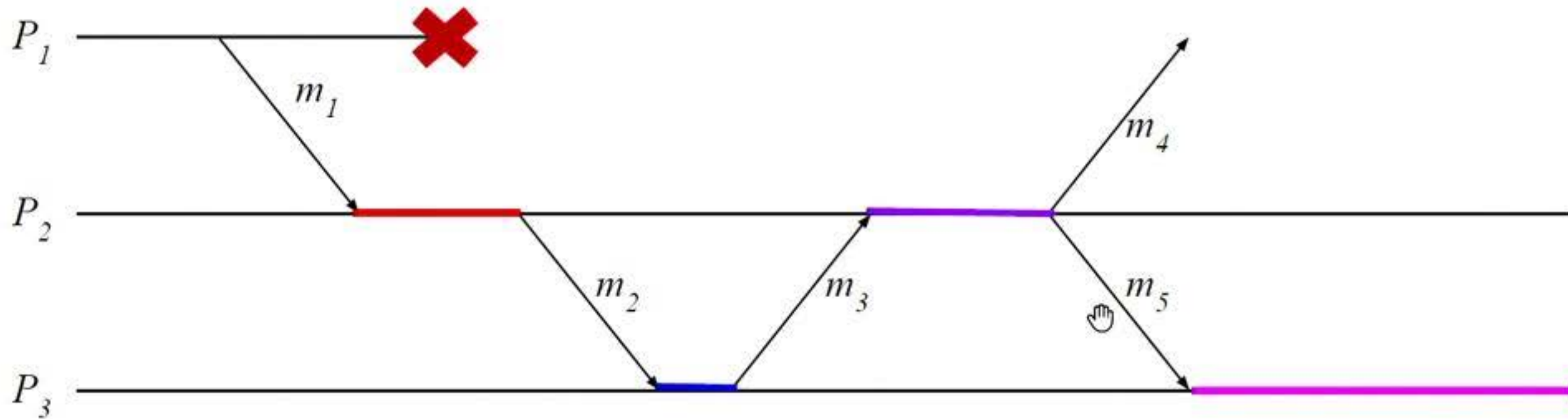
# System

- **Distributed system** = Processes + Communication Channel
- Processes communicate by *message-passing*



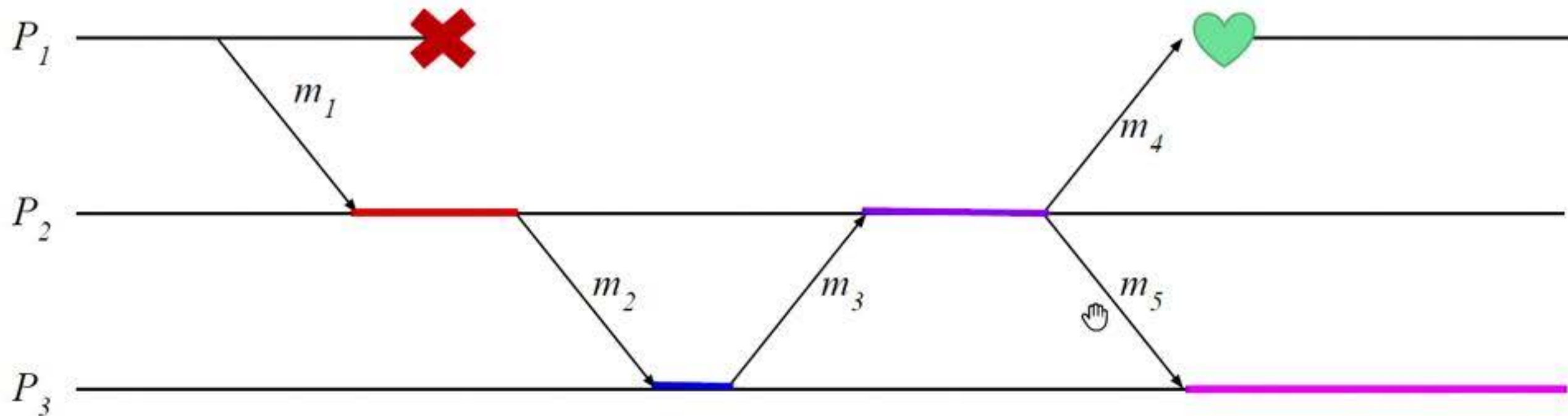
# System

- **Distributed system** = Processes + Communication Channel
- Processes communicate by *message-passing*
- Processes may **crash**



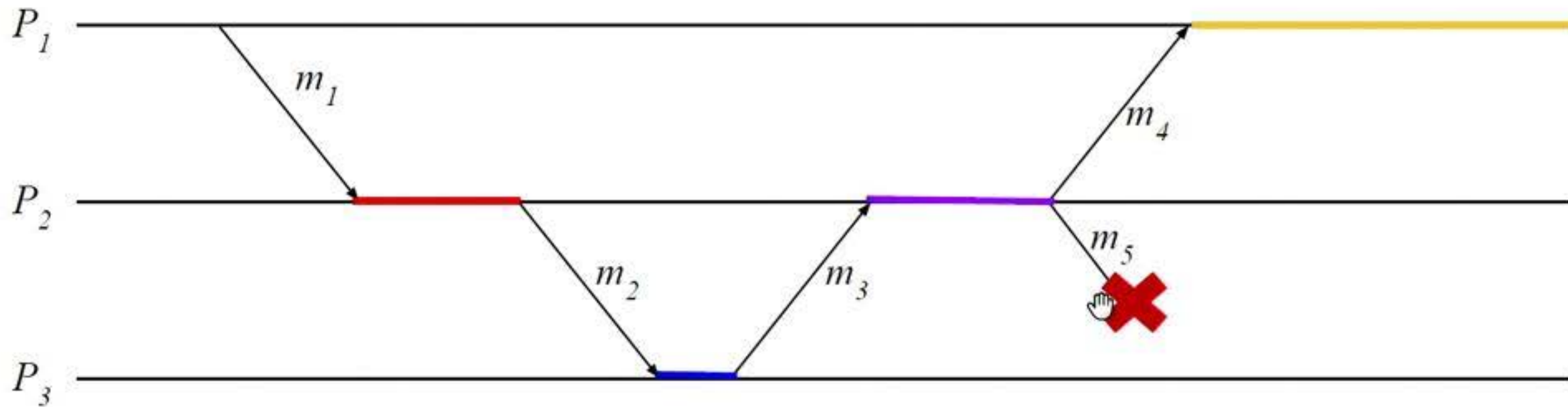
# System

- **Distributed system** = Processes + Communication Channel
- Processes communicate by *message-passing*
- Processes may crash **and later recover**



# System

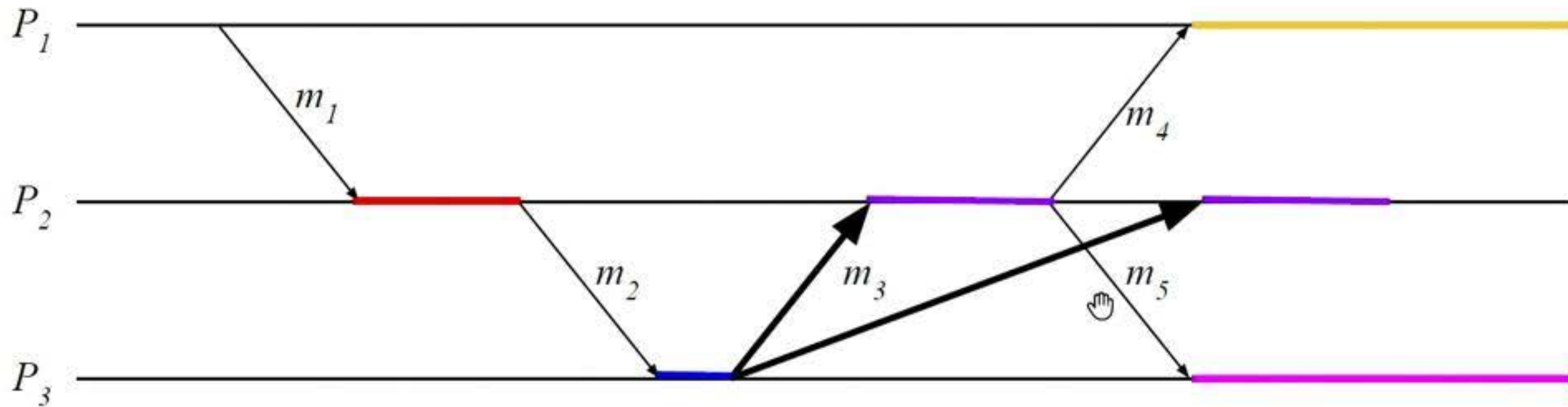
- **Distributed system** = Processes + Communication Channel
- Processes communicate by *message-passing*
- Processes may crash and later recover
- Messages may get **lost**





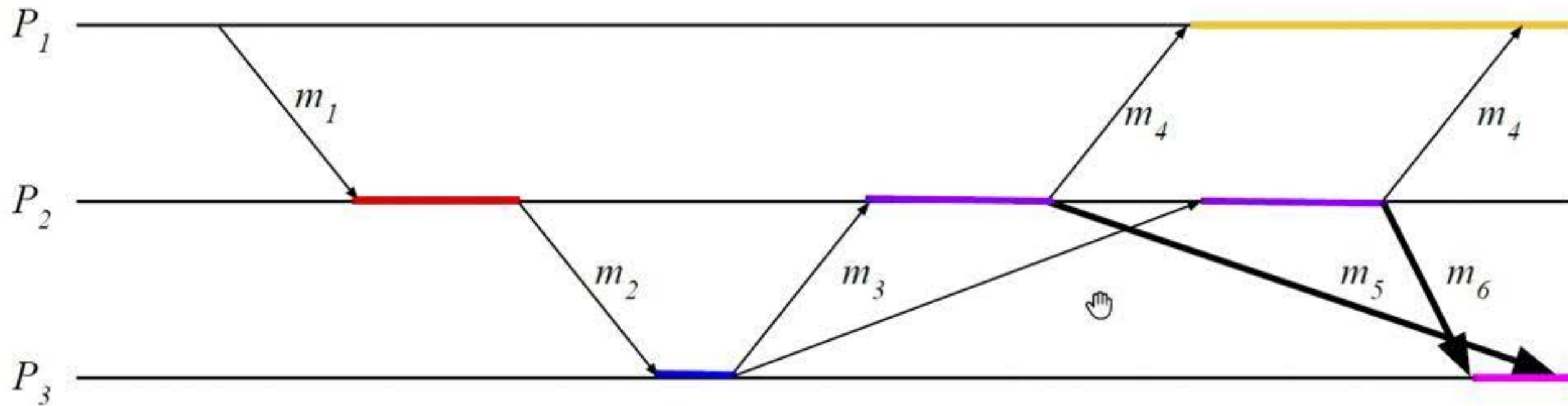
# System

- **Distributed system** = Processes + Communication Channel
- Processes communicate by *message-passing*
- Processes may crash and later recover
- Messages may get lost,  **duplicated**



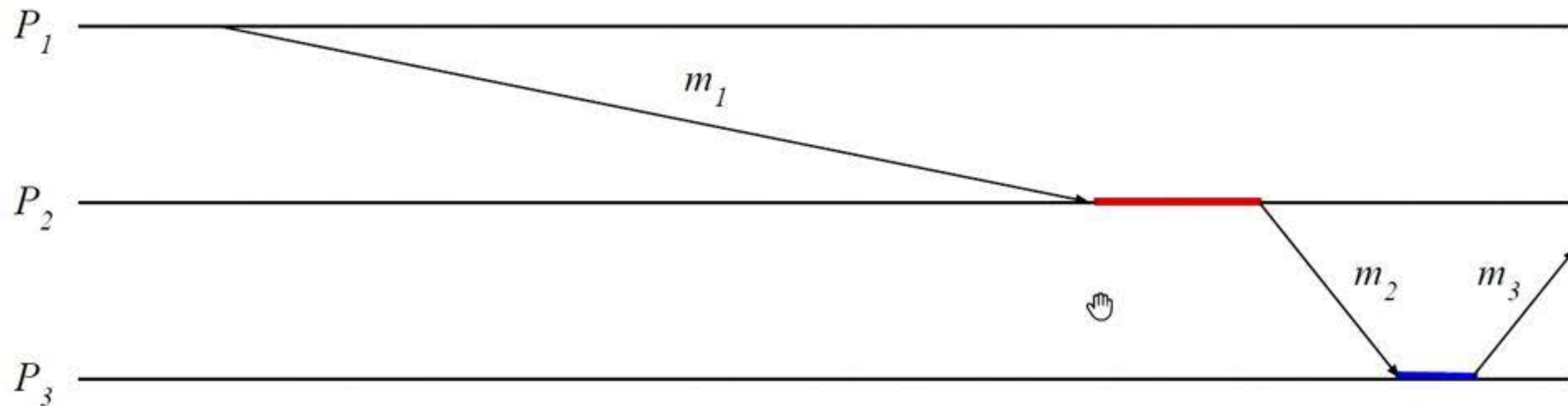
# System

- **Distributed system** = Processes + Communication Channel
- Processes communicate by *message-passing*
- Processes may crash and later recover
- Messages may get lost, duplicated, **received out-of-order**



# System

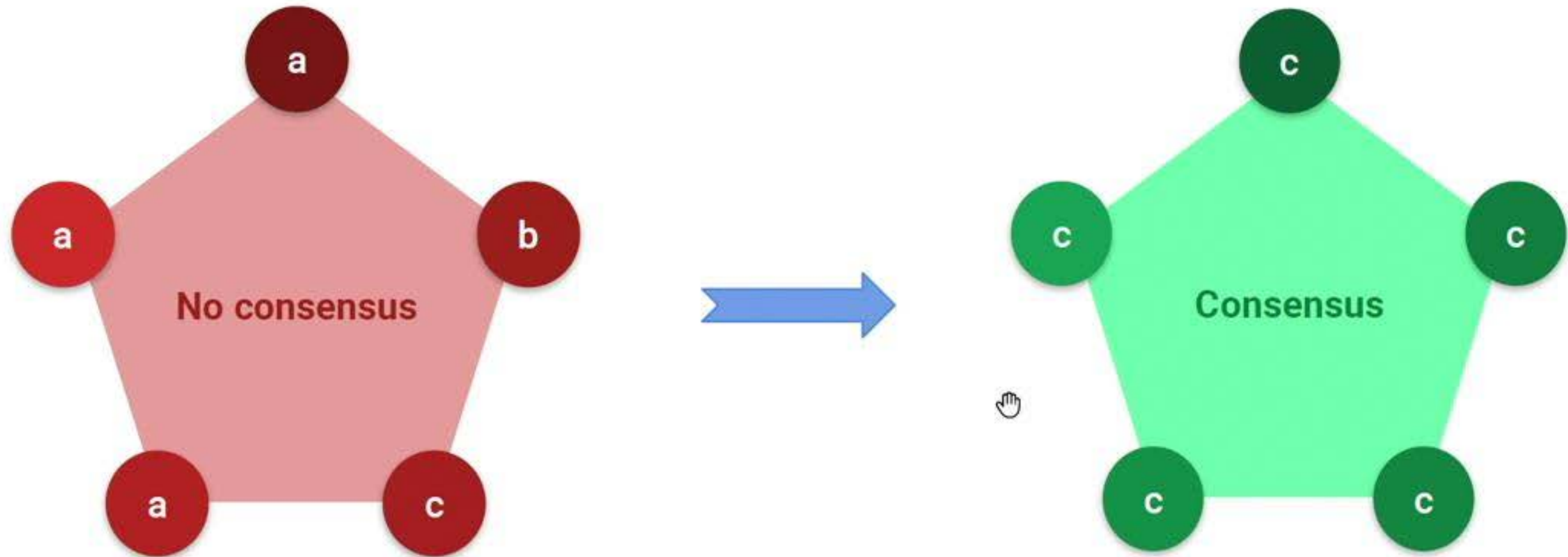
- **Distributed system** = Processes + Communication Channel
- Processes communicate by *message-passing*
- Processes may crash and later recover
- Messages may get lost, duplicated, received out-of-order, or **delayed indefinitely**





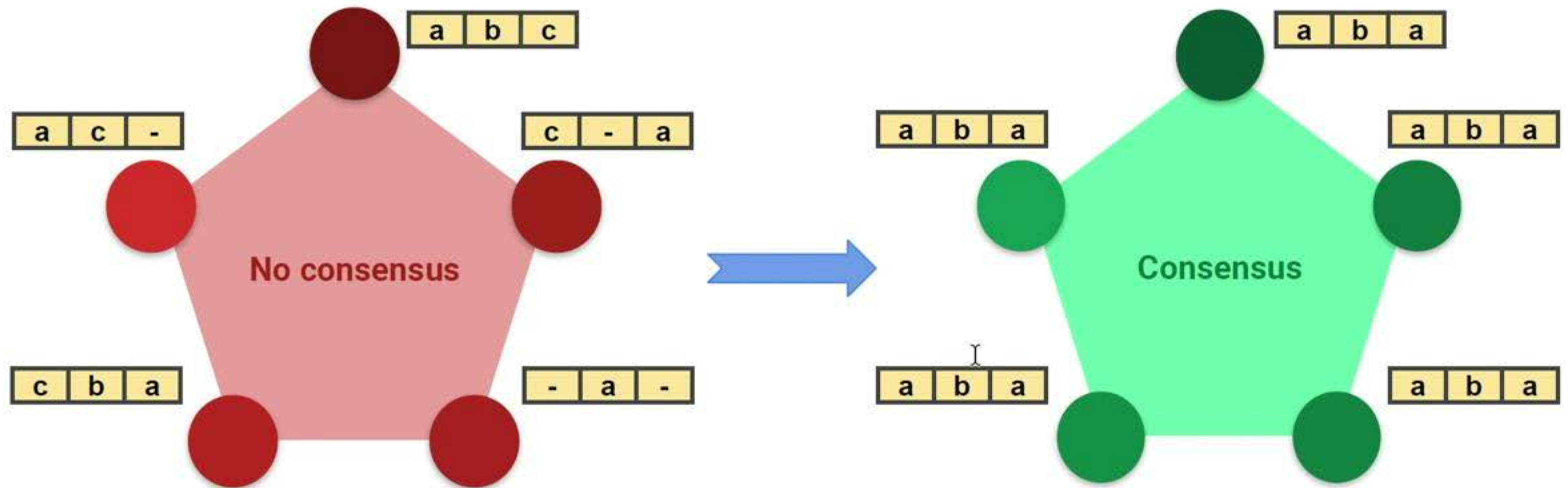
# The Problem of Consensus

- **Single-value Consensus:** Processes need to agree on **a single value**



# The Problem of Consensus

- **Multi-value Consensus:** Processes need to agree on a sequence of values



# Properties for consensus algorithms<sup>†</sup>

- **Safety:** At most a single value is chosen

$$\forall v_1, v_2 \in \text{Values}: \text{Chosen}(v_1) \wedge \text{Chosen}(v_2) \Rightarrow v_1 = v_2$$

- **Liveness:** Under some stable conditions, a value is eventually chosen.
  - [FLP]\* result states that in general consensus cannot be solved in an asynchronous system where even one process can fail. Thus, liveness is subject to some stable conditions.



<sup>†</sup>Presented for single-value consensus

\*[FLP]Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. 1985. Impossibility of distributed consensus with one faulty process. J. ACM 32, 2 (April 1985), 374-382. DOI=<http://dx.doi.org/10.1145/3149.214121>

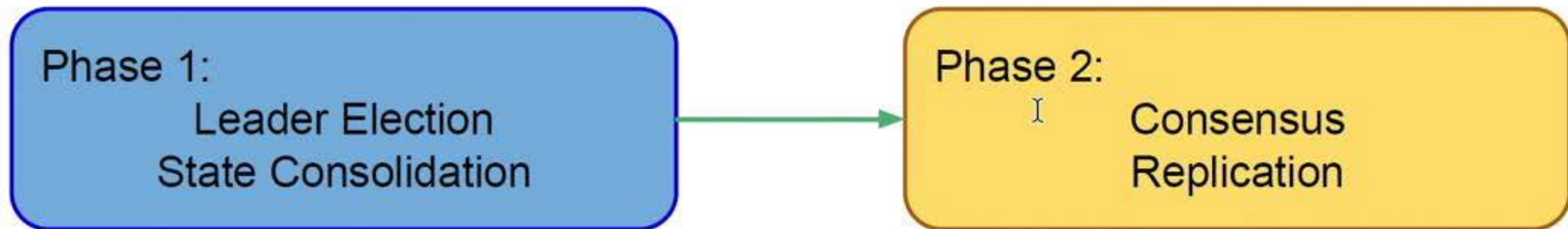


# Famous consensus algorithms

|                   | Name              | Description                                                                            | Language used         |
|-------------------|-------------------|----------------------------------------------------------------------------------------|-----------------------|
| Group Membership  | 1 VS-ISIS         | Reliable group communication, Birman-Joseph 1987                                       | English (items)       |
|                   | 2 VS-ISIS2        | Virtual synchrony, Birman-Joseph 1987                                                  | English               |
|                   | 3 EVS             | Extended Virtual Synchrony for network partition, Amir et al 1995                      | pseudocode            |
|                   | 4 Paxos-VS        | Virtually synchronous Paxos, Birman-Malkhi-van Reness 2012                             | pseudocode            |
| Primary-Backup    | 5 VR              | Viewstamped replication, Oki-Liskov 1988                                               | pseudocode (coarse)   |
|                   | 6 VR-Revisit      | VR revisited, Liskov 2012                                                              | English (items)       |
| PAXOS             | 7 Paxos-Synod     | Paxos in part-time parliament, Lamport 1998                                            | TLA (single-value)    |
|                   | 8 Paxos-Basic     | Single-value Paxos, Lamport 2001                                                       | English (items)       |
|                   | 9 Paxos-Fast      | Single-value Paxos with replicas proposing, Lamport 2006                               | English (items), TLA+ |
|                   | 10 Paxos-Vertical | Single-value Paxos with external starting of leader election, Lamport-Malkhi-Zhou 2009 | PlusCal               |
| Failure Detectors | 11 ACT            | Single-value consensus with failure detection, Aguilera-Chen-Toueg 2000                | pseudocode            |
|                   | 12 ACT-Store      | ACT with stable storage, same reference as above                                       | pseudocode            |

# Paxos at a high-level

- Assigns 2 roles to processes:
  - **Proposer,  $\mathcal{P}$** : Propose values that can be chosen
  - **Acceptor,  $\mathcal{A}$** : Vote on proposed values
- Assumes a Quorum set,  $\mathcal{Q} \subseteq \mathbf{P}(\mathcal{A})$  such that  $\forall Q_1, Q_2 \in \mathcal{Q}: Q_1 \cap Q_2 \neq \emptyset$
- 2 phase voting based algorithm:





# Lamport's English description of Paxos<sup>†</sup>

Putting the actions of the proposer and acceptor together, we see that the algorithm operates in the following two phases.

**Phase 1.** (a) A proposer selects a proposal number  $n$  and sends a *prepare* request with number  $n$  to a majority of acceptors.

(b) If an acceptor receives a *prepare* request with number  $n$  greater than that of any *prepare* request to which it has already responded, then it responds to the request with a promise not to accept any more proposals numbered less than  $n$  and with the highest-numbered proposal (if any) that it has accepted.

**Phase 2.** (a) If the proposer receives a response to its *prepare* requests (numbered  $n$ ) from a majority of acceptors, then it sends an *accept* request to each of those acceptors for a proposal numbered  $n$  with a value  $v$ , where  $v$  is the value of the highest-numbered proposal among the responses, or is any value if the responses reported no proposals.

(b) If an acceptor receives an *accept* request for a proposal numbered  $n$ , it accepts the proposal unless it has already responded to a *prepare* request having a number greater than  $n$ .

<sup>†</sup>Lamport, L. (2001). Paxos made simple. ACM Sigact News, 32(4), 18-25.



# Paxos in TLA+ by Lamport et al.<sup>†</sup>

```
Phase1a(b) == /\ ~ \E m \in msgs : (m.type = "1a") /\ (m.bal = b)
              /\ Send([type |> "1a", bal |> b])
              /\ UNCHANGED <<maxVBal, maxBal, maxVal>>
```

```
Phase1b(a) == \E m \in msgs :
  /\ m.type = "1a"
  /\ m.bal > maxBal[a]
  /\ Send([type |> "1b", bal |> m.bal, maxVBal |> maxVBal[a],
           maxVal |> maxVal[a], acc |> a])
  /\ maxBal' = [maxBal EXCEPT ![a] = m.bal]
  /\ UNCHANGED <<maxVBal, maxVal>>
```

```
Phase2a(b) ==
  /\ ~ \E m \in msgs : (m.type = "2a") /\ (m.bal = b)
  /\ \E v \in Values :
    /\ \E Q \in Quorums :
      \E S \in SUBSET {m \in msgs : (m.type = "1b") /\ (m.bal = b)} :
        /\ \A a \in Q : \E m \in S : m.acc = a
        /\ \/\ \A m \in S : m.maxVBal = -1
        /\ \E c \in 0..(b-1) :
          /\ \A m \in S : m.maxVBal =< c
          /\ \E m \in S : /\ m.maxVBal = c
                          /\ m.maxVal = v
        /\ Send([type |> "2a", bal |> b, val |> v])
  /\ UNCHANGED <<maxBal, maxVBal, maxVal>>

Phase2b(a) ==
  \E m \in msgs :
    /\ m.type = "2a"
    /\ m.bal >= maxBal[a]
    /\ Send([type |> "2b", bal |> m.bal, val |> m.val, acc |> a])
    /\ maxVBal' = [maxVBal EXCEPT ![a] = m.bal]
    /\ maxBal' = [maxBal EXCEPT ![a] = m.bal]
    /\ maxVal' = [maxVal EXCEPT ![a] = m.val]
```

<sup>†</sup>Lamport, L., Merz, S., Doligez, D.: A TLA+ specification of the Paxos Consensus algorithm



# Lamport's English description of Paxos<sup>†</sup>

Putting the actions of the proposer and acceptor together, we see that the algorithm operates in the following two phases.

**Phase 1.** (a) A proposer selects a proposal number  $n$  and sends a *prepare* request with number  $n$  to a majority of acceptors.

(b) If an acceptor receives a *prepare* request with number  $n$  greater than that of any *prepare* request to which it has already responded, then it responds to the request with a promise not to accept any more proposals numbered less than  $n$  and with the highest-numbered proposal (if any) that it has accepted.

**Phase 2.** (a) If the proposer receives a response to its *prepare* requests (numbered  $n$ ) from a majority of acceptors, then it sends an *accept* request to each of those acceptors for a proposal numbered  $n$  with a value  $v$ , where  $v$  is the value of the highest-numbered proposal among the responses, or is any value if the responses reported no proposals.

(b) If an acceptor receives an *accept* request for a proposal numbered  $n$ , it accepts the proposal unless it has already responded to a *prepare* request having a number greater than  $n$ .

<sup>†</sup>Lamport, L. (2001). Paxos made simple. ACM Sigact News, 32(4), 18-25.



# Paxos in TLA+ by Lamport et al.<sup>†</sup>

```
Phase1a(b) == /\ ~ \E m \in msgs : (m.type = "1a") /\ (m.bal = b)
              /\ Send([type |-> "1a", bal |-> b])
              /\ UNCHANGED <<maxVBal, maxBal, maxVal>>
```

```
Phase1b(a) == \E m \in msgs :
              /\ m.type = "1a"
              /\ m.bal > maxBal[a]
              /\ Send([type |-> "1b", bal |-> m.bal, maxVBal |-> maxVBal[a],
                      maxVal |-> maxVal[a], acc |-> a])
              /\ maxBal' = [maxBal EXCEPT ![a] = m.bal]
              /\ UNCHANGED <<maxVBal, maxVal>>
```

```
Phase2a(b) ==
              /\ ~ \E m \in msgs : (m.type = "2a") /\ (m.bal = b)
              /\ \E v \in Values :
                  /\ \E Q \in Quorums :
                      \E S \in SUBSET {m \in msgs : (m.type = "1b") /\ (m.bal = b)} :
                          /\ \A a \in Q : \E m \in S : m.acc = a
                          /\ \/\ \A m \in S : m.maxVBal = -1
                          /\ \E c \in 0..(b-1) :
                              /\ \A m \in S : m.maxVBal =< c
                              /\ \E m \in S : /\ m.maxVBal = c
                                  /\ m.maxVal = v
                          /\ Send([type |-> "2a", bal |-> b, val |-> v])
              /\ UNCHANGED <<maxBal, maxVBal, maxVal>>

Phase2b(a) == \E m \in msgs :
              /\ m.type = "2a"
              /\ m.bal >= maxBal[a]
              /\ Send([type |-> "2b", bal |-> m.bal, val |-> m.val, acc |-> a])
              /\ maxVBal' = [maxVBal EXCEPT ![a] = m.bal]
              /\ maxBal' = [maxBal EXCEPT ![a] = m.bal]
              /\ maxVal' = [maxVal EXCEPT ![a] = m.val]
```

<sup>†</sup>Lamport, L., Merz, S., Doligez, D.: A TLA+ specification of the Paxos Consensus algorithm



# Moving forward

- ***What's done:***

Formal specification with proof of safety of the exact phases of *single*-value Paxos in a general, high-level direct language like TLA+

- ***What's lacking:***

Formal specification with proof of safety of the exact phases of *multi*-value Paxos (also called *Multi-Paxos*) in a general, high-level direct language like TLA+

# **Formal Verification *of* Multi-Paxos *for* Distributed Consensus<sup>†</sup>**

<sup>†</sup>Chand, S., Liu Y.A., Stoller, S.D.: Formal Verification of Multi-Paxos for Distributed Consensus. In: Proceedings of the 21st International Symposium on Formal Methods, pp. 119–136. Springer (2016)

# Contributions

- Complete **formal specification** of Multi-Paxos in TLA+, minimally extending Lamport et al.'s specification for single-value Paxos.
- **Machine-checked proof of safety**, using TLA Proof System (TLAPS), with complete invariants, and a systematic proof method.
- Extending the specification and proof with **Preemption**, showing method for extending existing specification and proof.





# Specification

- Minimally extends Lamport et al.'s specification for single-value consensus
- Adds **slots** to become multi. For example,
  - **Before:** Propose value  $v$
  - **After:** Propose  $\langle \text{value}, \text{slot} \rangle$  pairs:  $\langle v_1, \mathbf{3} \rangle, \langle v_2, \mathbf{5} \rangle, \langle v_3, \mathbf{10} \rangle, \dots$
- Changes
  - Scalars  $\rightarrow$  Vectors
    - $\text{apples} = 5 \rightarrow \text{apples} = \langle 1, 2, 2 \rangle$
  - Sets of scalars  $\rightarrow$  Sets of vectors
    - $\text{apples} = \{\text{fuji}, \text{gala}, \text{honeycrisp}\} \rightarrow \text{apples} = \{\langle \text{fuji}, 1 \rangle, \langle \text{gala}, 2 \rangle, \langle \text{honeycrisp}, 2 \rangle\}$
  - Records of scalars  $\rightarrow$  Records of sets of vectors
    - $[\text{apples} \mapsto 5] \rightarrow [\text{apples} \mapsto \{\langle \text{fuji}, 1 \rangle, \langle \text{gala}, 2 \rangle, \langle \text{honeycrisp}, 2 \rangle\}]$

# Phase1a

## Basic Paxos

$$\begin{aligned}
 Phase1a(b \in \mathcal{B}) &\triangleq \\
 &\wedge \nexists m \in msgs : \wedge m.type = \text{"1a"} \\
 &\quad \wedge m.bal = b \\
 &\wedge Send([type \mapsto \text{"1a"}, \\
 &\quad bal \mapsto b]) \\
 &\wedge \text{UNCHANGED} \langle maxVBal, maxBal, maxVal \rangle
 \end{aligned}$$

## Multi-Paxos

$$\begin{aligned}
 Phase1a(p \in \mathcal{P}) &\triangleq \exists b \in \mathcal{B} : \\
 &\wedge \nexists m \in msgs : \wedge m.type = \text{"1a"} \\
 &\quad \wedge m.bal = b \\
 &\wedge Send([type \mapsto \text{"1a"}, \underline{from \mapsto p}, \\
 &\quad \quad \quad \underline{bal \mapsto b}]) \\
 &\wedge \underline{pBal' = [pBal \text{ EXCEPT } ![p] = b]} \\
 &\wedge \text{UNCHANGED} \langle aBal, aVoted \rangle
 \end{aligned}$$



# Phase1b

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase1b(a \in \mathcal{A}) \triangleq$<br>$\exists m \in msgs :$<br>$\wedge m.type = \text{"1a"}$<br>$\wedge m.bal > maxBal[a]$<br>$\wedge Send([type \mapsto \text{"1b"},$<br>$\quad acc \mapsto a,$<br>$\quad bal \mapsto m.bal,$<br>$\quad maxVBal \mapsto maxVBal[a],$<br>$\quad maxVal \mapsto maxVal[a]])$<br>$\wedge maxBal' = [maxBal \text{ EXCEPT } ![a] = m.bal]$<br>$\wedge \text{UNCHANGED} \langle maxVBal, maxVal \rangle$ | $Phase1b(a \in \mathcal{A}) \triangleq$<br>$\exists m \in msgs :$<br>$\wedge m.type = \text{"1a"}$<br>$\wedge m.bal > aBal[a]$<br>$\wedge Send([type \mapsto \text{"1b"},$<br>$\quad from \mapsto a,$<br>$\quad bal \mapsto m.bal,$<br>$\quad \underline{voted \mapsto aVoted[a]})$<br>$\wedge aBal' = [aBal \text{ EXCEPT } ![a] = m.bal]$<br>$\wedge \text{UNCHANGED} \langle pBal, aVoted \rangle$ |



# Phase1a

## Basic Paxos

$$\begin{aligned}
 Phase1a(b \in \mathcal{B}) &\triangleq \\
 &\wedge \nexists m \in msgs : \wedge m.type = \text{"1a"} \\
 &\quad \wedge m.bal = b \\
 &\wedge Send([type \mapsto \text{"1a"}, \\
 &\quad bal \mapsto b]) \\
 &\wedge \text{UNCHANGED} \langle maxVBal, maxBal, maxVal \rangle
 \end{aligned}$$

## Multi-Paxos

$$\begin{aligned}
 Phase1a(p \in \mathcal{P}) &\triangleq \exists b \in \mathcal{B} : \\
 &\wedge \nexists m \in msgs : \wedge m.type = \text{"1a"} \\
 &\quad \wedge m.bal = b \\
 &\wedge Send([type \mapsto \text{"1a"}, \underline{from \mapsto p}, \\
 &\quad \quad \quad \underline{bal \mapsto b}]) \\
 &\wedge \underline{pBal' = [pBal \text{ EXCEPT } ![p] = b]} \\
 &\wedge \text{UNCHANGED} \langle aBal, aVoted \rangle
 \end{aligned}$$

# Phase1b

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase1b(a \in \mathcal{A}) \triangleq$<br>$\exists m \in msgs :$<br>$\wedge m.type = \text{"1a"}$<br>$\wedge m.bal > maxBal[a]$<br>$\wedge Send([type \mapsto \text{"1b",}$<br>$\quad acc \mapsto a,$<br>$\quad bal \mapsto m.bal,$<br>$\quad maxVBal \mapsto maxVBal[a],$<br>$\quad maxVal \mapsto maxVal[a]])$<br>$\wedge maxBal' = [maxBal \text{ EXCEPT } ![a] = m.bal]$<br>$\wedge \text{UNCHANGED} \langle maxVBal, maxVal \rangle$ | $Phase1b(a \in \mathcal{A}) \triangleq$<br>$\exists m \in msgs :$<br>$\wedge m.type = \text{"1a"}$<br>$\wedge m.bal > aBal[a]$<br>$\wedge Send([type \mapsto \text{"1b",}$<br>$\quad from \mapsto a,$<br>$\quad bal \mapsto m.bal,$<br>$\quad \underline{voted \mapsto aVoted[a]})$<br>$\wedge aBal' = [aBal \text{ EXCEPT } ![a] = m.bal]$<br>$\wedge \text{UNCHANGED} \langle pBal, aVoted \rangle$ |



# Phase2a

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase2a(b \in \mathcal{B}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = b$ $\wedge \exists v \in \mathcal{V} :$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = b\} :$ $\wedge \forall a \in Q : \exists m \in S : m.acc = a$ $\wedge \forall m \in S : m.maxVBal = -1$ $\vee \exists c \in 0..(b-1) :$ $\wedge \forall m \in S : m.maxVBal \leq c$ $\wedge \exists m \in S : (m.maxVBal = c) \\ \wedge m.maxVal = v$ $\wedge Send([type \mapsto \text{"2a"}, bal \mapsto b, val \mapsto v])$ $\wedge \text{UNCHANGED} \langle maxBal, maxVBal, maxVal \rangle$ | $Phase2a(p \in \mathcal{P}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = pBal[p]$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = pBal[p]\} :$ $\wedge \forall a \in Q : \exists m \in S : m.from = a$ <div style="border: 1px solid green; padding: 5px; margin: 5px 0;"> <math display="block">\wedge Send([type \mapsto \text{"2a"}, \\ from \mapsto p, \\ bal \mapsto pBal[p], \\ propSV \mapsto PropSV(\text{UNION} \\ \{m.voted : m \in S\})])</math> </div> $\wedge \text{UNCHANGED} \langle pBal, aBal, aVoted \rangle$ |



# Phase1a

## Basic Paxos

$$\begin{aligned}
 Phase1a(b \in \mathcal{B}) &\triangleq \\
 &\wedge \nexists m \in msgs : \wedge m.type = \text{"1a"} \\
 &\quad \wedge m.bal = b \\
 &\wedge Send([type \mapsto \text{"1a"}, \\
 &\quad bal \mapsto b]) \\
 &\wedge \text{UNCHANGED} \langle maxVBal, maxBal, maxVal \rangle
 \end{aligned}$$

## Multi-Paxos

$$\begin{aligned}
 Phase1a(p \in \mathcal{P}) &\triangleq \exists b \in \mathcal{B} : \\
 &\wedge \nexists m \in msgs : \wedge m.type = \text{"1a"} \\
 &\quad \wedge m.bal = b \\
 &\wedge Send([type \mapsto \text{"1a"}, \underline{from \mapsto p}, \\
 &\quad \quad \quad \underline{bal \mapsto b}]) \\
 &\wedge \underline{pBal' = [pBal \text{ EXCEPT } ![p] = b]} \\
 &\wedge \text{UNCHANGED} \langle aBal, aVoted \rangle
 \end{aligned}$$

# Phase2a

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase2a(b \in \mathcal{B}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = b$ $\wedge \exists v \in \mathcal{V} :$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = b\} :$ $\wedge \forall a \in Q : \exists m \in S : m.acc = a$ $\wedge \forall m \in S : m.maxVBal = -1$ $\vee \exists c \in 0..(b-1) :$ $\wedge \forall m \in S : m.maxVBal \leq c$ $\wedge \exists m \in S : (m.maxVBal = c) \\ \wedge m.maxVal = v$ $\wedge Send([type \mapsto \text{"2a"}, bal \mapsto b, val \mapsto v])$ $\wedge \text{UNCHANGED} \langle maxBal, maxVBal, maxVal \rangle$ | $Phase2a(p \in \mathcal{P}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = pBal[p]$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = pBal[p]\} :$ $\wedge \forall a \in Q : \exists m \in S : m.from = a$ <div style="border: 1px solid green; padding: 5px; margin: 5px 0;"> <math display="block">\wedge Send([type \mapsto \text{"2a"}, \\ from \mapsto p, \\ bal \mapsto pBal[p], \\ propSV \mapsto PropSV(\text{UNION} \\ \{m.voted : m \in S\})])</math> </div> $\wedge \text{UNCHANGED} \langle pBal, aBal, aVoted \rangle$ |



# Phase2a

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase2a(b \in \mathcal{B}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = b$ $\wedge \exists v \in \mathcal{V} :$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = b\} :$ $\wedge \forall a \in Q : \exists m \in S : m.acc = a$ $\wedge \forall m \in S : m.maxVBal = -1$ $\vee \exists c \in 0..(b-1) :$ $\wedge \forall m \in S : m.maxVBal \leq c$ $\wedge \exists m \in S : (m.maxVBal = c) \\ \wedge m.maxVal = v$ $\wedge Send([type \mapsto \text{"2a"}, bal \mapsto b, val \mapsto v])$ $\wedge \text{UNCHANGED} \langle maxBal, maxVBal, maxVal \rangle$ | $Phase2a(p \in \mathcal{P}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = pBal[p]$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = pBal[p]\} :$ $\wedge \forall a \in Q : \exists m \in S : m.from = a$ $\wedge Send([type \mapsto \text{"2a"}, \\ from \mapsto p, \\ bal \mapsto pBal[p], \\ propSV \mapsto PropSV(\text{UNION} \\ \{m.voted : m \in S\})])$ $\wedge \text{UNCHANGED} \langle pBal, aBal, aVoted \rangle$ <p>where,</p> |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | $PropSV(T) \triangleq MaxSV(T) \cup NewSV(T)$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |



# Phase2a

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase2a(b \in \mathcal{B}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = b$ $\wedge \exists v \in \mathcal{V} :$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = b\} :$ $\wedge \forall a \in Q : \exists m \in S : m.acc = a$ $\wedge \forall m \in S : m.maxVBal = -1$ $\vee \exists c \in 0..(b-1) :$ $\wedge \forall m \in S : m.maxVBal \leq c$ $\wedge \exists m \in S : (m.maxVBal = c) \\ \wedge m.maxVal = v$ $\wedge Send([type \mapsto \text{"2a"}, bal \mapsto b, val \mapsto v])$ $\wedge \text{UNCHANGED} \langle maxBal, maxVBal, maxVal \rangle$ | $Phase2a(p \in \mathcal{P}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = pBal[p]$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = pBal[p]\} :$ $\wedge \forall a \in Q : \exists m \in S : m.from = a$ <div style="border: 1px solid green; padding: 5px; margin: 5px 0;"> <math display="block">\wedge Send([type \mapsto \text{"2a"}, \\ from \mapsto p, \\ bal \mapsto pBal[p], \\ propSV \mapsto PropSV(\text{UNION} \\ \{m.voted : m \in S\})])</math> </div> $\wedge \text{UNCHANGED} \langle pBal, aBal, aVoted \rangle$ |

# Phase2a

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase2a(b \in \mathcal{B}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = b$ $\wedge \exists v \in \mathcal{V} :$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = b\} :$ $\wedge \forall a \in Q : \exists m \in S : m.acc = a$ $\wedge \forall m \in S : m.maxVBal = -1$ $\vee \exists c \in 0..(b-1) :$ $\wedge \forall m \in S : m.maxVBal \leq c$ $\wedge \exists m \in S : (m.maxVBal = c) \\ \wedge m.maxVal = v$ $\wedge Send([type \mapsto \text{"2a"}, bal \mapsto b, val \mapsto v])$ $\wedge \text{UNCHANGED} \langle maxBal, maxVBal, maxVal \rangle$ | $Phase2a(p \in \mathcal{P}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = pBal[p]$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = pBal[p]\} :$ $\wedge \forall a \in Q : \exists m \in S : m.from = a$ $\wedge Send([type \mapsto \text{"2a"}, \\ from \mapsto p, \\ bal \mapsto pBal[p], \\ propSV \mapsto PropSV(\text{UNION} \\ \{m.voted : m \in S\})])$ $\wedge \text{UNCHANGED} \langle pBal, aBal, aVoted \rangle$ <p>where,</p> |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | $PropSV(T) \triangleq MaxSV(T) \cup NewSV(T)$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |



# Phase2a

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase2a(b \in \mathcal{B}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = b$ $\wedge \exists v \in \mathcal{V} :$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = b\} :$ $\wedge \forall a \in Q : \exists m \in S : m.acc = a$ $\wedge \forall m \in S : m.maxVBal = -1$ $\vee \exists c \in 0..(b-1) :$ $\wedge \forall m \in S : m.maxVBal \leq c$ $\wedge \exists m \in S : (m.maxVBal = c) \\ \wedge m.maxVal = v$ $\wedge Send([type \mapsto \text{"2a"}, bal \mapsto b, val \mapsto v])$ $\wedge \text{UNCHANGED} \langle maxBal, maxVBal, maxVal \rangle$ | $Phase2a(p \in \mathcal{P}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = pBal[p]$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = pBal[p]\} :$ $\wedge \forall a \in Q : \exists m \in S : m.from = a$ $\wedge Send([type \mapsto \text{"2a"}, \\ from \mapsto p, \\ bal \mapsto pBal[p], \\ propSV \mapsto PropSV(\text{UNION} \\ \{m.voted : m \in S\})])$ $\wedge \text{UNCHANGED} \langle pBal, aBal, aVoted \rangle$ <p>where,</p> $MaxBSV(T) \triangleq \{t \in T : \forall t2 \in T : \\ t2.slot = t.slot \Rightarrow t2.bal \leq t.bal\}$ $MaxSV(T) \triangleq \{[slot \mapsto t.slot, \\ val \mapsto t.val] : t \in MaxBSV(T)\}$ $PropSV(T) \triangleq MaxSV(T) \cup NewSV(T)$ |



# Phase2a

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase2a(b \in \mathcal{B}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = b$ $\wedge \exists v \in \mathcal{V} :$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = b\} :$ $\wedge \forall a \in Q : \exists m \in S : m.acc = a$ $\wedge \forall m \in S : m.maxVBal = -1$ $\vee \exists c \in 0..(b-1) :$ $\wedge \forall m \in S : m.maxVBal \leq c$ $\wedge \exists m \in S : (m.maxVBal = c) \\ \wedge m.maxVal = v$ $\wedge Send([type \mapsto \text{"2a"}, bal \mapsto b, val \mapsto v])$ $\wedge \text{UNCHANGED} \langle maxBal, maxVBal, maxVal \rangle$ | $Phase2a(p \in \mathcal{P}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = pBal[p]$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = pBal[p]\} :$ $\wedge \forall a \in Q : \exists m \in S : m.from = a$ $\wedge Send([type \mapsto \text{"2a"}, \\ from \mapsto p, \\ bal \mapsto pBal[p], \\ propSV \mapsto PropSV(\text{UNION} \\ \{m.voted : m \in S\})])$ $\wedge \text{UNCHANGED} \langle pBal, aBal, aVoted \rangle$ <p>where,</p> $MaxBSV(T) \triangleq \{t \in T : \forall t2 \in T : \\ t2.slot = t.slot \Rightarrow t2.bal \leq t.bal\}$ $MaxSV(T) \triangleq \{[slot \mapsto t.slot, \\ val \mapsto t.val] : t \in MaxBSV(T)\}$ $UnusedS(T) \triangleq \{s \in \mathcal{S} : \nexists t \in T : t.slot = s\}$ $NewSV(T) \triangleq \text{CHOOSE } D \subseteq [slot : \\ UnusedS(T), val : \mathcal{V}] : \forall d1, d2 \in D : \\ d1.slot = d2.slot \Rightarrow d1 = d2$ $PropSV(T) \triangleq MaxSV(T) \cup NewSV(T)$ |

# Phase1b

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase2b(a \in \mathcal{A}) \triangleq$<br>$\exists m \in msgs :$<br>$\wedge m.type = \text{"2a"}$<br>$\wedge m.bal \geq maxBal[a]$<br>$\wedge Send([type \mapsto \text{"2b"},$<br>$\quad acc \mapsto a,$<br>$\quad bal \mapsto m.bal,$<br>$\quad val \mapsto m.val])$<br>$\wedge maxBal' = [maxBal \text{ EXCEPT } ![a] = m.bal]$<br>$\wedge maxVBal' =$<br>$\quad [maxVBal \text{ EXCEPT } ![a] = m.bal]$<br>$\wedge maxVal' =$<br>$\quad [maxVal \text{ EXCEPT } ![a] = m.val]$ | $Phase2b(a \in \mathcal{A}) \triangleq$<br>$\exists m \in msgs :$<br>$\wedge m.type = \text{"2a"}$<br>$\wedge m.bal \geq aBal[a]$<br>$\wedge Send([type \mapsto \text{"2b"},$<br>$\quad from \mapsto a,$<br>$\quad bal \mapsto m.bal,$<br>$\quad propSV \mapsto m.propSV])$<br>$\wedge aBal' = [aBal \text{ EXCEPT } ![a] = m.bal]$<br><div style="border: 1px solid green; padding: 5px;"> <math>\wedge aVoted' = [aVoted \text{ EXCEPT } ![a] =</math><br/> <math>\quad \cup \{[bal \mapsto m.bal, slot \mapsto d.slot,</math><br/> <math>\quad \quad val \mapsto d.val] : d \in m.propSV\}</math><br/> <math>\quad \cup \{e \in aVoted[a] :</math><br/> <math>\quad \quad \nexists r \in m.propSV : e.slot = r.slot\}]</math> </div> $\wedge \text{UNCHANGED } \langle pBal \rangle$ |



# Phase2a

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase2a(b \in \mathcal{B}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = b$ $\wedge \exists v \in \mathcal{V} :$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = b\} :$ $\wedge \forall a \in Q : \exists m \in S : m.acc = a$ $\wedge \forall m \in S : m.maxVBal = -1$ $\vee \exists c \in 0..(b-1) :$ $\wedge \forall m \in S : m.maxVBal \leq c$ $\wedge \exists m \in S : (m.maxVBal = c) \\ \wedge m.maxVal = v$ $\wedge Send([type \mapsto \text{"2a"}, bal \mapsto b, val \mapsto v])$ $\wedge \text{UNCHANGED} \langle maxBal, maxVBal, maxVal \rangle$ | $Phase2a(p \in \mathcal{P}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = pBal[p]$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = pBal[p]\} :$ $\wedge \forall a \in Q : \exists m \in S : m.from = a$ $\wedge Send([type \mapsto \text{"2a"}, \\ from \mapsto p, \\ bal \mapsto pBal[p], \\ propSV \mapsto PropSV(\text{UNION} \\ \{m.voted : m \in S\})])$ $\wedge \text{UNCHANGED} \langle pBal, aBal, aVoted \rangle$ <p>where,</p> $MaxBSV(T) \triangleq \{t \in T : \forall t2 \in T : \\ t2.slot = t.slot \Rightarrow t2.bal \leq t.bal\}$ $MaxSV(T) \triangleq \{[slot \mapsto t.slot, \\ val \mapsto t.val] : t \in MaxBSV(T)\}$ $UnusedS(T) \triangleq \{s \in \mathcal{S} : \nexists t \in T : t.slot = s\}$ $NewSV(T) \triangleq \text{CHOOSE } D \subseteq [slot : \\ UnusedS(T), val : \mathcal{V}] : \forall d1, d2 \in D : \\ d1.slot = d2.slot \Rightarrow d1 = d2$ $PropSV(T) \triangleq MaxSV(T) \cup NewSV(T)$ |



# Phase1b

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase2b(a \in \mathcal{A}) \triangleq$<br>$\exists m \in msgs :$<br>$\wedge m.type = \text{"2a"}$<br>$\wedge m.bal \geq maxBal[a]$<br>$\wedge Send([type \mapsto \text{"2b"},$<br>$\quad acc \mapsto a,$<br>$\quad bal \mapsto m.bal,$<br>$\quad val \mapsto m.val])$<br>$\wedge maxBal' = [maxBal \text{ EXCEPT } ![a] = m.bal]$<br>$\wedge maxVBal' =$<br>$\quad [maxVBal \text{ EXCEPT } ![a] = m.bal]$<br>$\wedge maxVal' =$<br>$\quad [maxVal \text{ EXCEPT } ![a] = m.val]$ | $Phase2b(a \in \mathcal{A}) \triangleq$<br>$\exists m \in msgs :$<br>$\wedge m.type = \text{"2a"}$<br>$\wedge m.bal \geq aBal[a]$<br>$\wedge Send([type \mapsto \text{"2b"},$<br>$\quad from \mapsto a,$<br>$\quad bal \mapsto m.bal,$<br>$\quad propSV \mapsto m.propSV])$<br>$\wedge aBal' = [aBal \text{ EXCEPT } ![a] = m.bal]$<br>$\wedge aVoted' = [aVoted \text{ EXCEPT } ![a] =$<br>$\quad \cup \{[bal \mapsto m.bal, slot \mapsto d.slot,$<br>$\quad \quad val \mapsto d.val] : d \in m.propSV\}$<br>$\quad \cup \{e \in aVoted[a] :$<br>$\quad \quad \nexists r \in m.propSV : e.slot = r.slot\}]$<br>$\wedge \text{UNCHANGED } \langle pBal \rangle$ |

# Phase2a

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase2a(b \in \mathcal{B}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = b$ $\wedge \exists v \in \mathcal{V} :$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge m.bal = b\} :$ $\wedge \forall a \in Q : \exists m \in S : m.acc = a$ $\wedge \forall m \in S : m.maxVBal = -1$ $\vee \exists c \in 0..(b-1) :$ $\wedge \forall m \in S : m.maxVBal \leq c$ $\wedge \exists m \in S : (m.maxVBal = c)$ $\wedge m.maxVal = v$ $\wedge Send([type \mapsto \text{"2a"}, bal \mapsto b, val \mapsto v])$ $\wedge \text{UNCHANGED} \langle maxBal, maxVBal, maxVal \rangle$ | $Phase2a(p \in \mathcal{P}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = pBal[p]$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge m.bal = pBal[p]\} :$ $\wedge \forall a \in Q : \exists m \in S : m.from = a$ $\wedge Send([type \mapsto \text{"2a"},$ $from \mapsto p,$ $bal \mapsto pBal[p],$ $propSV \mapsto PropSV(\text{UNION}$ $\{m.voted : m \in S\}))$ $\wedge \text{UNCHANGED} \langle pBal, aBal, aVoted \rangle$ <p>where,</p> $MaxBSV(T) \triangleq \{t \in T : \forall t2 \in T : \\ t2.slot = t.slot \Rightarrow t2.bal \leq t.bal\}$ $MaxSV(T) \triangleq \{[slot \mapsto t.slot,$ $val \mapsto t.val] : t \in MaxBSV(T)\}$ $UnusedS(T) \triangleq \{s \in \mathcal{S} : \nexists t \in T : t.slot = s\}$ $NewSV(T) \triangleq \text{CHOOSE } D \subseteq [slot :$ $UnusedS(T), val : \mathcal{V}] : \forall d1, d2 \in D :$ $d1.slot = d2.slot \Rightarrow d1 = d2$ $PropSV(T) \triangleq MaxSV(T) \cup NewSV(T)$ |



# Phase1b

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase2b(a \in \mathcal{A}) \triangleq$<br>$\exists m \in msgs :$<br>$\wedge m.type = \text{"2a"}$<br>$\wedge m.bal \geq maxBal[a]$<br>$\wedge Send([type \mapsto \text{"2b"},$<br>$\quad acc \mapsto a,$<br>$\quad bal \mapsto m.bal,$<br>$\quad val \mapsto m.val])$<br>$\wedge maxBal' = [maxBal \text{ EXCEPT } ![a] = m.bal]$<br>$\wedge maxVBal' =$<br>$\quad [maxVBal \text{ EXCEPT } ![a] = m.bal]$<br>$\wedge maxVal' =$<br>$\quad [maxVal \text{ EXCEPT } ![a] = m.val]$ | $Phase2b(a \in \mathcal{A}) \triangleq$<br>$\exists m \in msgs :$<br>$\wedge m.type = \text{"2a"}$<br>$\wedge m.bal \geq aBal[a]$<br>$\wedge Send([type \mapsto \text{"2b"},$<br>$\quad from \mapsto a,$<br>$\quad bal \mapsto m.bal,$<br>$\quad propSV \mapsto m.propSV])$<br>$\wedge aBal' = [aBal \text{ EXCEPT } ![a] = m.bal]$<br>$\wedge aVoted' = [aVoted \text{ EXCEPT } ![a] =$<br>$\quad \cup \{[bal \mapsto m.bal, slot \mapsto d.slot,$<br>$\quad \quad val \mapsto d.val] : d \in m.propSV\}$<br>$\quad \cup \{e \in aVoted[a] :$<br>$\quad \quad \nexists r \in m.propSV : e.slot = r.slot\}]$<br>$\wedge \text{UNCHANGED } \langle pBal \rangle$ |

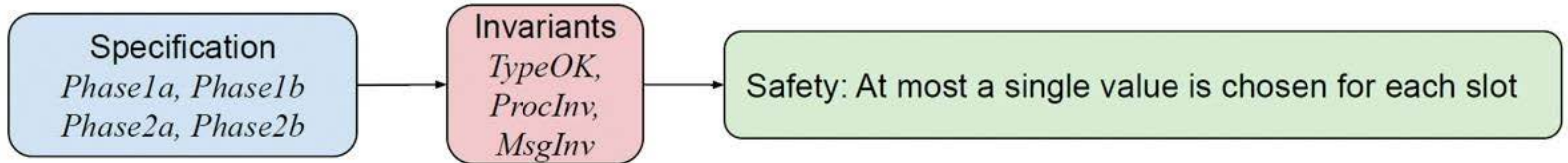
# Phase2a

| Basic Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Multi-Paxos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase2a(b \in \mathcal{B}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = b$ $\wedge \exists v \in \mathcal{V} :$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = b\} :$ $\wedge \forall a \in Q : \exists m \in S : m.acc = a$ $\wedge \forall m \in S : m.maxVBal = -1$ $\vee \exists c \in 0..(b-1) :$ $\wedge \forall m \in S : m.maxVBal \leq c$ $\wedge \exists m \in S : (m.maxVBal = c) \\ \wedge m.maxVal = v$ $\wedge Send([type \mapsto \text{"2a"}, bal \mapsto b, val \mapsto v])$ $\wedge \text{UNCHANGED} \langle maxBal, maxVBal, maxVal \rangle$ | $Phase2a(p \in \mathcal{P}) \triangleq$ $\wedge \nexists m \in msgs : \wedge m.type = \text{"2a"} \\ \wedge m.bal = pBal[p]$ $\wedge \exists Q \in \mathcal{Q}, S \subseteq \{m \in msgs : m.type = \text{"1b"} \wedge \\ m.bal = pBal[p]\} :$ $\wedge \forall a \in Q : \exists m \in S : m.from = a$ $\wedge Send([type \mapsto \text{"2a"}, \\ from \mapsto p, \\ bal \mapsto pBal[p], \\ propSV \mapsto PropSV(\text{UNION} \\ \{m.voted : m \in S\})])$ $\wedge \text{UNCHANGED} \langle pBal, aBal, aVoted \rangle$ <p>where,</p> $MaxBSV(T) \triangleq \{t \in T : \forall t2 \in T : \\ t2.slot = t.slot \Rightarrow t2.bal \leq t.bal\}$ $MaxSV(T) \triangleq \{[slot \mapsto t.slot, \\ val \mapsto t.val] : t \in MaxBSV(T)\}$ $UnusedS(T) \triangleq \{s \in \mathcal{S} : \nexists t \in T : t.slot = s\}$ $NewSV(T) \triangleq \text{CHOOSE } D \subseteq [slot : \\ UnusedS(T), val : \mathcal{V}] : \forall d1, d2 \in D : \\ d1.slot = d2.slot \Rightarrow d1 = d2$ $PropSV(T) \triangleq MaxSV(T) \cup NewSV(T)$ |

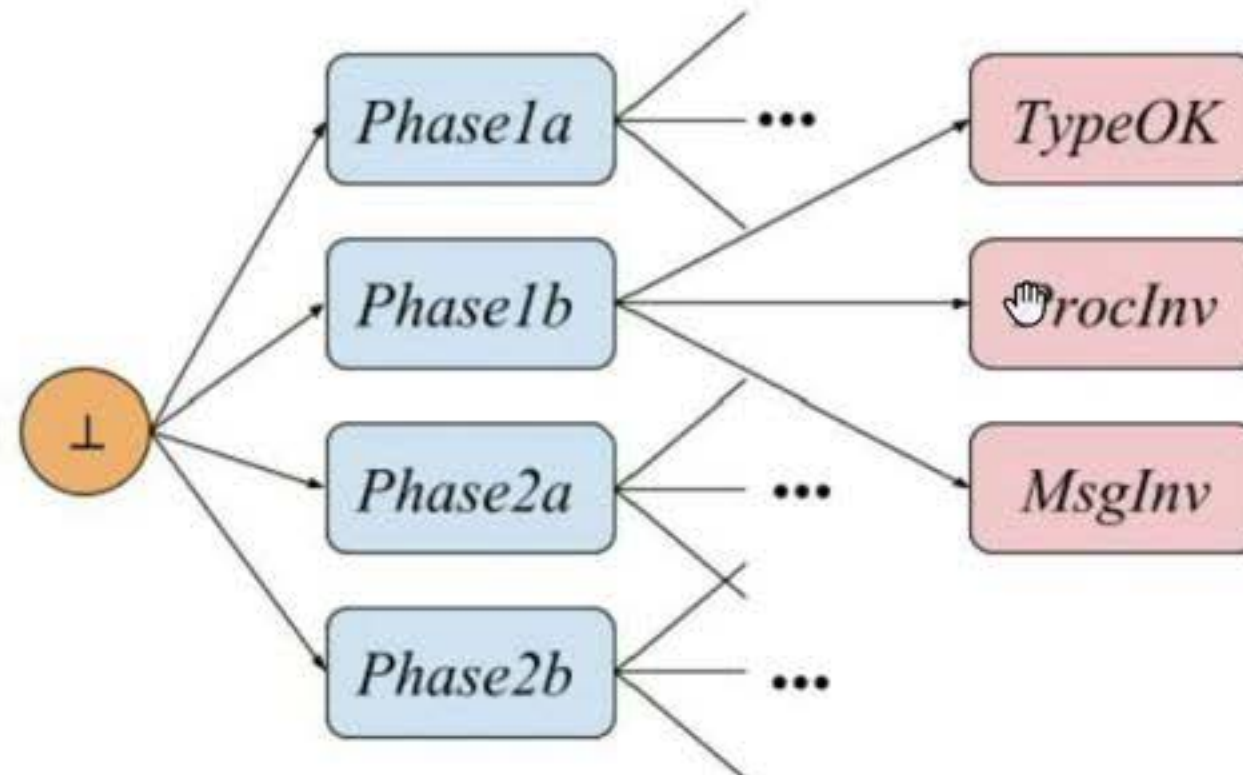


# Verification

- Similar proof strategy and skeleton to Lamport et al.'s



- Hierarchical proof by induction



# Overview of Results

| Metric              | Single-value Paxos | Multi-Paxos |          | Multi-Paxos w/ Preemption |          |
|---------------------|--------------------|-------------|----------|---------------------------|----------|
|                     | Value              | Value       | Increase | Value                     | Increase |
| Specification size  | 52                 | 55          | 6%       | 74                        | 35%      |
| Proof size          | 310                | 787         | 154%     | 831                       | 6%       |
| No. of obligations  | 239                | 779         | 226%     | 825                       | 6%       |
| Proof check time(s) | 16                 | 58          | 263%     | 110                       | 90%      |

- Increase in proof parameters for Multi-Paxos due to complications due to vectors
- Conflated increase in proof check time for Multi-Paxos w/Preemption due to a new library import



# Can we do better?

- Can the **Specification** be even more high-level?
- Can the **Invariants** be made to more easily follow from the specification?



# Can we do better?

- Can the **Specification** be even more high-level?
- Can the **Invariants** be made to more easily follow from the specification?

Use **History Variables!!!**





# **Simpler Specifications *and* Easier Proofs *using* History Variables<sup>†</sup>**

<sup>†</sup>Chand, S., & Liu, Y. A. Simpler specifications and easier proofs of distributed algorithms using history variables. In NASA Formal Methods Symposium (pp. 70-86). Springer (2018).

# Contributions

- Demonstrate a **Systematic Style** to write specifications of distributed algorithms using message history variables - *sent* and *received* (in TLA+).
- A **Method** to systematically derive important invariants of the algorithm.
- Case studies:
  - Single-value Paxos for single-value consensus by Lamport et al. [1]
  - Multi-Paxos for multi-value consensus by Chand et al. [2]
  - Multi-Paxos with preemption by Chand et al. [2]

[1] Lamport, L., Merz, S., Doligez, D. Paxos.tla. [github.com/tlaplus/v1-tlapm/blob/master/examples/paxos/Paxos.tla](https://github.com/tlaplus/v1-tlapm/blob/master/examples/paxos/Paxos.tla)

[2] Chand, S., Liu, Y. A., and Stoller, S. D. Formal verification of multi-paxos for distributed consensus. In FM 2016: Formal Methods: 21st International Symposium, pages 119–136. Springer, 2016

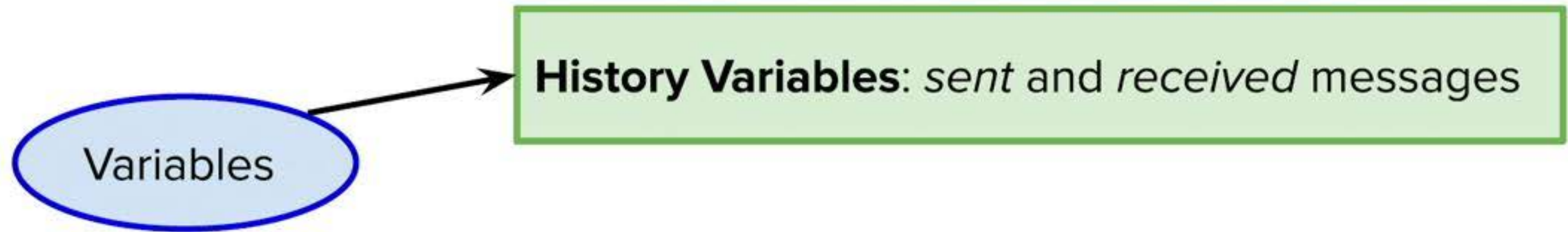


# Results

| Metric              | Basic Paxos |       |      | Multi-Paxos |                                                                                          |      | Multi-Paxos w/ Preemption |       |      |
|---------------------|-------------|-------|------|-------------|------------------------------------------------------------------------------------------|------|---------------------------|-------|------|
|                     | Lam         | Us    | Decr | Cha         | Us                                                                                       | Decr | Cha                       | Us    | Decr |
| Specification size  | 52          | 39    | 25%  | 55          | 42                                                                                       | 24%  | 74                        | 52    | 30%  |
| Proof size          | 310         | 227   | 27%  | 787         | 520                                                                                      | 34%  | 831                       | 538   | 35%  |
| No. of invariants   | 15          | 5, 1^ | 60%  | 16          | 7, 1^                                                                                    | 50%  | 17                        | 7, 1^ | 53%  |
| Proof check time(s) | 16          | 12    | 25%  | 58          | 28  | 52%  | 110                       | 30    | 73%  |

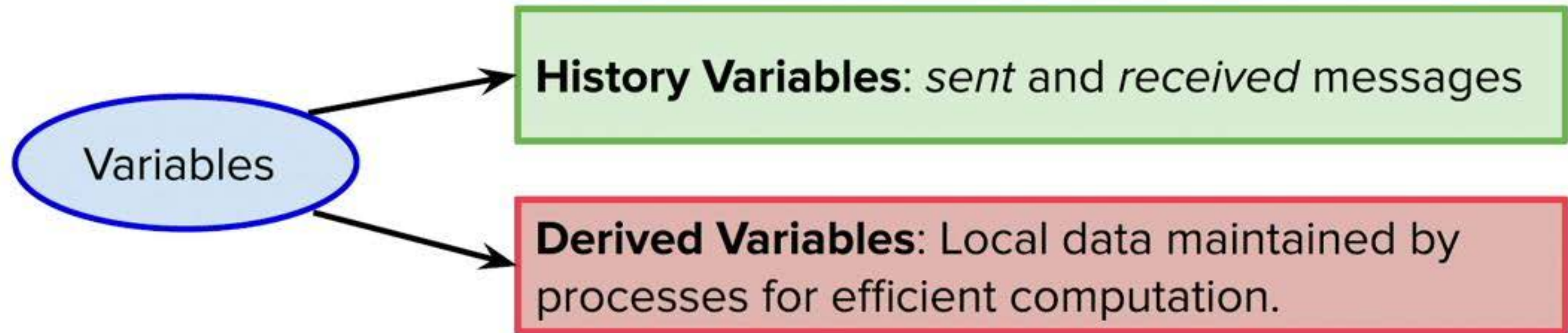
Lam: Lamport et al. [1], Cha: Chand et al. [2], Us: Using History Variables only, Decr: percentage of decrease.

# History and Derived Variables

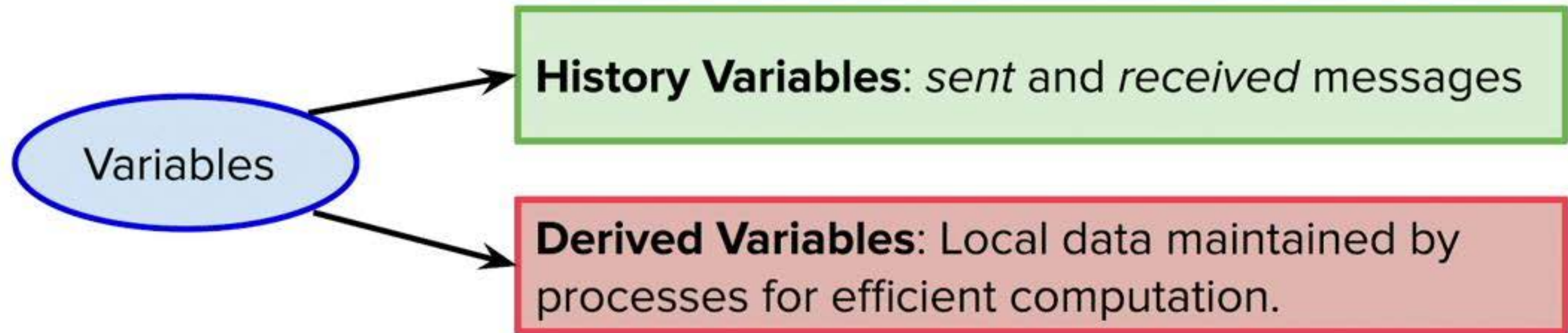




# History and Derived Variables



# History and Derived Variables



- BUT, local data is updated upon receiving some<sup>hand</sup> set of messages
- Thus, **derived variables** are functions of **history variables**
- So, write specifications using **history variables *only***

# Specification of Basic Paxos

**Phase 1a.** A proposer selects a proposal number  $b$  and sends a **1a** request with number  $b$  to a majority of acceptors.

Lamport et al.'s

$Phase1a(b \in \mathcal{B}) \triangleq$   
 $\wedge \nexists m \in sent : (m.type = \text{"1a"}) \wedge (m.bal = b)$   
 $\wedge Send([type \mapsto \text{"1a"}, bal \mapsto b])$   
 $\wedge \text{UNCHANGED} \langle maxVBal, maxBal, maxVal \rangle$

Using *sent* only

$Phase1a(b \in \mathcal{B}) \triangleq$   
 $\text{Send}([type \mapsto \text{"1a"}, bal \mapsto b])$



**Phase 1b.** If an acceptor receives a 1a request with number  $bal$  greater than that of any 1a request to which it has already responded, then it responds to the request with a promise not to accept any more proposals numbered less than  $bal$  and with the highest-numbered proposal (if any) that it has accepted.

| Lamport et al.'s                                                                                                                                                                                                                                                                                                                                                                            | Using <i>sent</i> only                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $Phase1b(a \in \mathcal{A}) \triangleq$ $\exists m \in sent :$ $\wedge m.type = \text{"1a"}$ $\wedge m.bal > maxBal[a]$<br>$\wedge Send([type \mapsto \text{"1b"},$ $acc \mapsto a, bal \mapsto m.bal,$ $maxVBal \mapsto maxVBal[a],$ $maxVal \mapsto maxVal[a]])$<br>$\wedge maxBal' =$ $[maxBal \text{ EXCEPT } ![a] = m.bal]$ $\wedge \text{UNCHANGED } \langle maxVBal, maxVal \rangle$ | $Phase1b(a \in \mathcal{A}) \triangleq$ $\exists m \in sent, r \in max\_prop(a) :$ $\wedge m.type = \text{"1a"}$ $\wedge \forall m2 \in sent : m2.type \in \{\text{"1b"}, \text{"2b"}\} \wedge$ $m2.acc = a \Rightarrow m.bal > m2.bal$ $\wedge Send([type \mapsto \text{"1b"},$ $acc \mapsto a, bal \mapsto m.bal,$ $maxVBal \mapsto r.bal,$ $maxVal \mapsto r.val])$<br>$2bs(a) \triangleq \{m \in sent : m.type = \text{"2b"} \wedge m.acc = a\}$ $max\_prop(a) \triangleq$ $\text{IF } 2bs(a) = \emptyset \text{ THEN } \{[bal \mapsto -1, val \mapsto \perp]\}$ $\text{ELSE } \{m \in 2bs(a) : \forall m2 \in 2bs(a) : m.bal \geq m2.bal\}$ |

**Phase 2a.** If the proposer receives a response to its 1a requests (numbered  $b$ ) from a majority of acceptors, then it sends a 2a request to each of those acceptors for a proposal numbered  $b$  with a value  $v$ , where  $v$  is the value of the highest-numbered proposal among the 1b responses, or is any value if the responses reported no proposals.

Lamport et al.'s

$$\begin{aligned}
 &Phase2a(b \in \mathcal{B}) \triangleq \\
 &\wedge \nexists m \in sent : m.type = \text{"2a"} \wedge m.bal = b \\
 &\wedge \exists v \in \mathcal{V}, Q \in \mathcal{Q}, S \subseteq \{m \in sent : \\
 &\quad m.type = \text{"1b"} \wedge m.bal = b\} : \\
 &\quad \wedge \forall a \in Q : \exists m \in S : m.acc = a \\
 &\quad \wedge \forall m \in S : m.maxVBal = -1 \\
 &\quad \vee \exists c \in 0..(b-1) : \\
 &\quad \quad \wedge \forall m \in S : m.maxVBal \leq c \\
 &\quad \quad \wedge \exists m \in S : m.maxVBal = c \\
 &\quad \quad \wedge m.maxVal = v \\
 &\quad \wedge Send([type \mapsto \text{"2a"}, bal \mapsto b, val \mapsto v]) \\
 &\wedge \text{UNCHANGED} \langle maxBal, maxVBal, \\
 &\quad maxVal \rangle
 \end{aligned}$$

Using *sent* only

$$\begin{aligned}
 &Phase2a(b \in \mathcal{B}) \triangleq \\
 &\wedge \nexists m \in sent : m.type = \text{"2a"} \wedge m.bal = b \\
 &\wedge \exists v \in \mathcal{V}, Q \in \mathcal{Q}, S \subseteq \{m \in sent : \\
 &\quad m.type = \text{"1b"} \wedge m.bal = b\} : \\
 &\quad \wedge \forall a \in Q : \exists m \in S : m.acc = a \\
 &\quad \wedge \forall m \in S : m.maxVBal = -1 \\
 &\quad \vee \exists c \in 0..(b-1) : \\
 &\quad \quad \wedge \forall m \in S : m.maxVBal \leq c \\
 &\quad \quad \wedge \exists m \in S : m.maxVBal = c \\
 &\quad \quad \wedge m.maxVal = v \\
 &\quad \wedge Send([type \mapsto \text{"2a"}, bal \mapsto b, val \mapsto v])
 \end{aligned}$$



**Phase 2b.** If an acceptor receives a **2a** request for a proposal numbered  $bal$ , it accepts the proposal unless it has already responded to a **1a** request having a number greater than  $bal$ .

Lamport et al.'s

$Phase2b(a \in \mathcal{A}) \triangleq$

$\exists m \in sent :$

$\wedge m.type = \text{"2a"}$

$\wedge m.bal \geq maxBal[a]$

$\wedge Send([type \mapsto \text{"2b"}, acc \mapsto a,$   
 $bal \mapsto m.bal, val \mapsto m.val])$

$\wedge maxBal' = [maxBal \text{ EXCEPT } ![a] = m.bal]$

$\wedge maxVBal' = [maxVBal \text{ EXCEPT } ![a] = m.bal]$

$\wedge maxVal' = [maxVal \text{ EXCEPT } ![a] = m.val]$

Using *sent* only

$Phase2b(a \in \mathcal{A}) \triangleq$

$\exists m \in sent :$

$\wedge m.type = \text{"2a"}$

$\wedge \forall m2 \in sent : m2.type \in \{\text{"1b"}, \text{"2b"}\} \wedge$   
 $m2.acc = a \Rightarrow m.bal \geq m2.bal$

$\wedge Send([type \mapsto \text{"2b"}, acc \mapsto a,$   
 $bal \mapsto m.bal, val \mapsto m.val])$

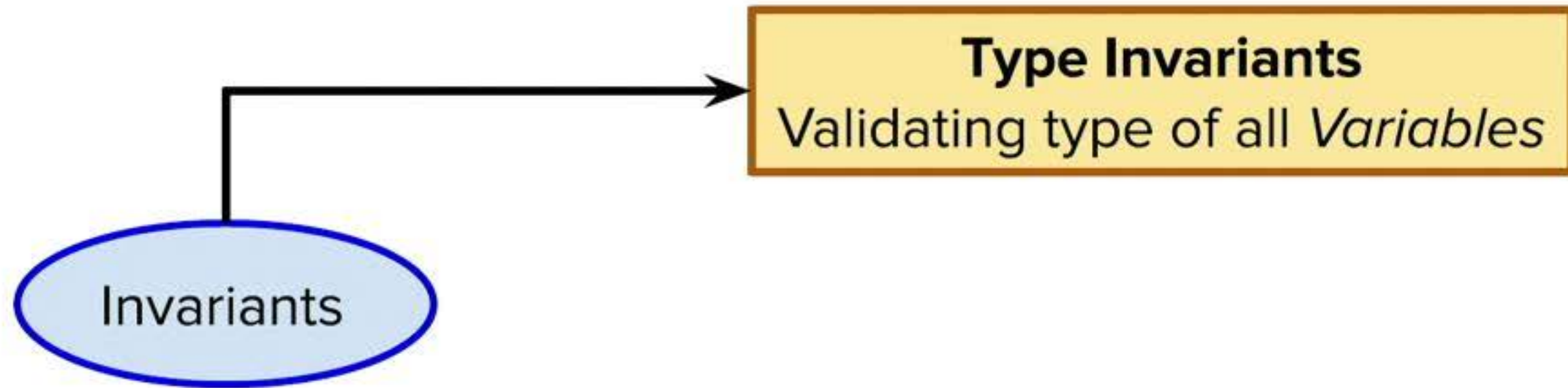


**Phase 1b.** If an acceptor receives a 1a request with number  $bal$  greater than that of any 1a request to which it has already responded, then it responds to the request with a promise not to accept any more proposals numbered less than  $bal$  and with the highest-numbered proposal (if any) that it has accepted.

Lamport et al.'s

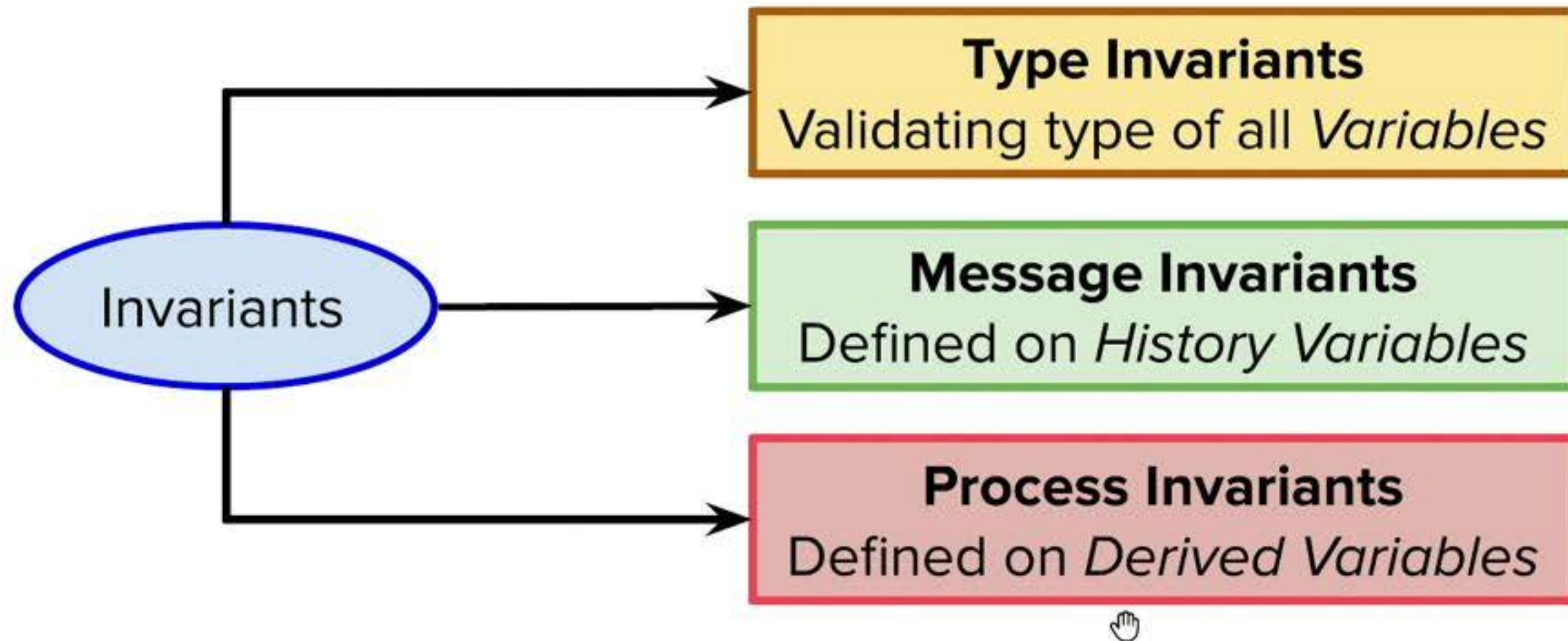
Using *sent* only
$$\begin{aligned} \text{Phase1b}(a \in \mathcal{A}) &\triangleq \\ \exists m \in \text{sent} : & \\ \wedge m.\text{type} = \text{"1a"} & \\ \wedge m.\text{bal} > \text{maxBal}[a] & \\ & \\ \wedge \text{Send}([type \mapsto \text{"1b"}, & \\ \quad acc \mapsto a, bal \mapsto m.bal, & \\ \quad \text{maxVBal} \mapsto \text{maxVBal}[a], & \\ \quad \text{maxVal} \mapsto \text{maxVal}[a]]) & \\ & \\ \wedge \text{maxBal}' = & \\ \quad [\text{maxBal} \text{ EXCEPT } ![a] = m.bal] & \\ \wedge \text{UNCHANGED } \langle \text{maxVBal}, \text{maxVal} \rangle & \end{aligned}$$
$$\begin{aligned}
\text{Phase1b}(a \in \mathcal{A}) &\triangleq \\
&\exists m \in \text{sent}, r \in \text{max\_prop}(a) : \\
&\quad \wedge m.\text{type} = \text{"1a"} \\
&\quad \wedge \forall m2 \in \text{sent} : m2.\text{type} \in \{\text{"1b"}, \text{"2b"}\} \wedge \\
&\quad \quad m2.\text{acc} = a \Rightarrow m.\text{bal} > m2.\text{bal} \\
&\quad \wedge \text{Send}([ \text{type} \mapsto \text{"1b"}, \\
&\quad \quad \text{acc} \mapsto a, \text{bal} \mapsto m.\text{bal}, \\
&\quad \quad \text{maxVBal} \mapsto r.\text{bal}, \\
&\quad \quad \text{maxVal} \mapsto r.\text{val} ]) \\
&\quad \quad \quad \text{👉} \\
2bs(a) &\triangleq \{ m \in \text{sent} : m.\text{type} = \text{"2b"} \wedge m.\text{acc} = a \} \\
\text{max\_prop}(a) &\triangleq \\
&\text{IF } 2bs(a) = \emptyset \text{ THEN } \{ [ \text{bal} \mapsto -1, \text{val} \mapsto \perp ] \} \\
&\text{ELSE } \{ m \in 2bs(a) : \forall m2 \in 2bs(a) : m.\text{bal} \geq m2.\text{bal} \}
\end{aligned}$$

# Invariants

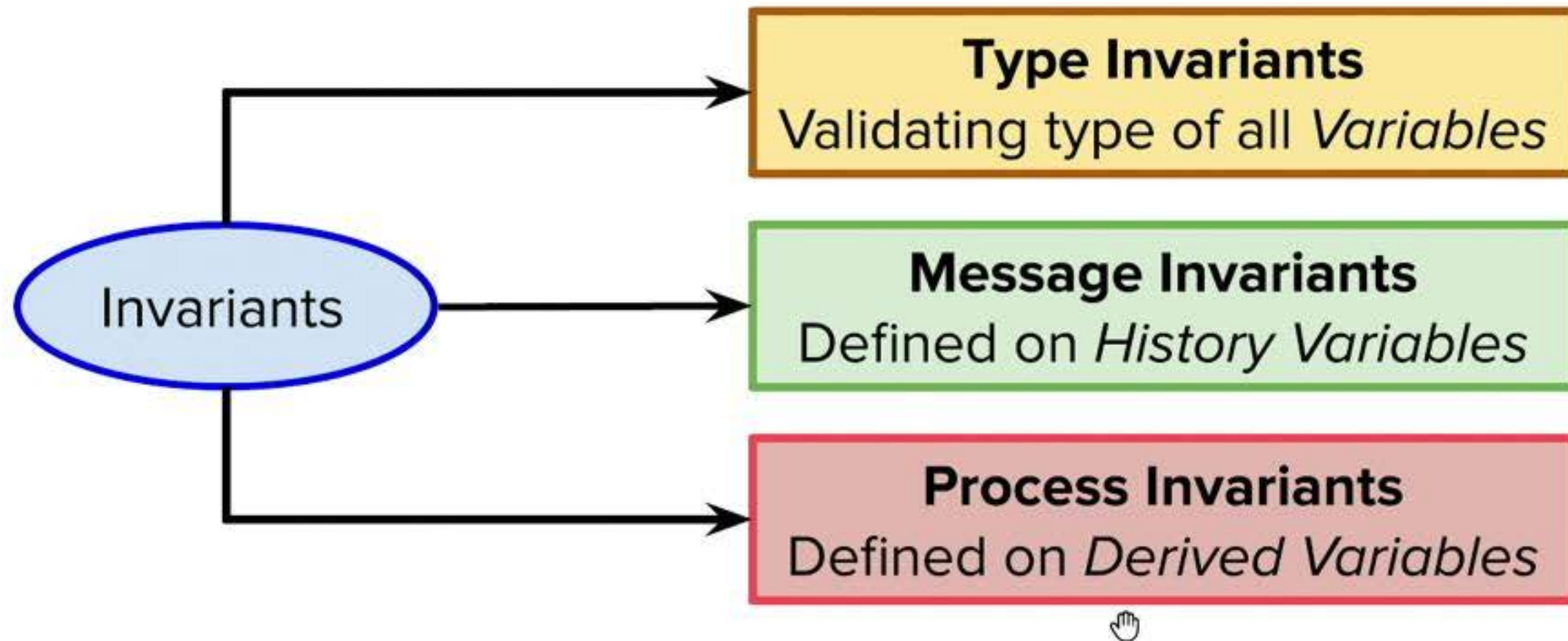




# Invariants



# Invariants



## Advantages with History Variables:

- 1) No derived variables  $\Rightarrow$  Fewer invariants (~50% fewer in our case studies)
- 2) Systematically derive message invariants (Could derive all but 1 in our case studies)

# 3-Step Method for deriving Message Invariants

## 1. Increment

- a. Look at the increment made to *sent* by an action

**Reply.** Upon receiving a request numbered  $n$ , a process replies with an acknowledgement with the same number  $n$ .

$$\begin{aligned} \text{Reply}(p \in \mathcal{P}) &\triangleq \\ &\exists m \in \text{sent} : \\ &\quad \wedge \text{Receive}(m, p) \\ &\quad \wedge m.\text{type} = \text{"req"} \\ &\quad \wedge \text{Send}([type \mapsto \text{"ack"}, \\ &\quad \quad reqnum \mapsto m.reqnum, \\ &\quad \quad to \mapsto m.from]) \end{aligned}$$



# 3-Step Method for Deriving Message Invariants

1. Increment
  - Look at the increment made to *sent* by an action
2. Analyze
  - Analyze and connect the **contents of msg** and the **body of the action**

$$\begin{aligned}\phi(msg) = & \exists m \in sent: \\ & m.type = \text{"req"} \wedge \\ & msg.reqnum = m.reqnum \wedge \\ & msg.to = m.from\end{aligned}$$

**Reply.** Upon receiving a request numbered  $n$ , a process replies with an acknowledgement with the same number  $n$ .

$$\begin{aligned}Reply(p \in \mathcal{P}) & \triangleq \\ & \exists m \in sent : \\ & \quad \wedge Receive(m, p) \\ & \quad \wedge m.type = \text{"req"} \\ & \quad \wedge Send([type \mapsto \text{"ack"}, \\ & \quad \quad reqnum \mapsto m.reqnum, \\ & \quad \quad to \mapsto m.from])\end{aligned}$$

# 3-Step Method for Deriving Message Invariants

## 1. Increment

- Look at the increment made to *sent* by an action

## 2. Analyze

- Analyze and connect the contents of *msg* and the body of the action

## 3. Assimilate

- Use the properties found in step 2 to derive an invariant:

$$\begin{aligned} \text{MsgInvAcc} &\triangleq \forall \text{msg} \in \text{sent}: \text{msg.type} = \text{"ack"} \Rightarrow \phi(\text{msg}) \\ \phi(\text{msg}) &= \exists m \in \text{sent}: m.\text{type} = \text{"req"} \wedge \text{msg.to} = m.\text{from} \wedge \text{msg.reqnum} = \\ &\quad m.\text{reqnum} \end{aligned}$$

## 2b Message Invariant

$Phase2b(a \in \mathcal{A}) \triangleq$

$\exists m \in sent :$

$\wedge \underline{m.type = "2a"}$

$\wedge \forall m2 \in sent : m2.type \in \{"1b", "2b"\} \wedge$

$m2.acc = a \Rightarrow m.bal \geq m2.bal$

$\wedge Send([type \mapsto "2b", acc \mapsto a,$

$\underline{bal \mapsto m.bal, val \mapsto m.val}])$



$\forall msg \in sent : msg.type = "2b" \Rightarrow$

$\exists m \in sent :$

$\wedge \underline{m.type = "2a"}$

$\wedge \underline{msg.bal = m.bal}$

$\wedge \underline{msg.val = m.val}$





## 2a Message Invariant

$Phase2a(b \in \mathcal{B}) \triangleq$   
 $\wedge \forall m \in sent : m.type = \text{"2a"} \Rightarrow m.bal \neq b$   
 $\wedge \exists v \in \mathcal{V}, Q \in \mathcal{Q}, S \subseteq \{m \in sent :$   
     $m.type = \text{"1b"} \wedge m.bal = b\} :$   
     $\wedge \forall a \in Q : \exists m \in S : m.acc = a$   
     $\wedge \forall m \in S : m.maxVBal = -1$   
     $\vee \exists c \in 0..(b-1) :$   
         $\wedge \forall m \in S : m.maxVBal \leq c$   
         $\wedge \exists m \in S : \wedge m.maxVBal = c$   
         $\wedge m.maxVal = v$   
 $\wedge Send([type \mapsto \text{"2a"}, \text{bal} \mapsto b, val \mapsto v])$



$\forall msg \in sent : msg.type = \text{"2a"} \Rightarrow$   
 $\forall m \in sent :$   
     $\wedge \text{m.type} = \text{"2a"}$   
     $\wedge msg.bal = m.bal$   
     $\Rightarrow msg = m$



# 1b Message Invariant

$Phase1b(a \in \mathcal{A}) \triangleq$

$\exists m \in sent, r \in max\_prop(a) :$

$\wedge m.type = \text{"1a"}$

$\wedge \forall m2 \in sent : m2.type \in \{\text{"1b"}, \text{"2b"}\} \wedge$   
 $m2.acc = a \Rightarrow m.bal > m2.bal$

$\wedge Send([type \mapsto \text{"1b"},$   
 $acc \mapsto a, bal \mapsto m.bal,$   
 $maxVBal \mapsto r.bal,$   
 $maxVal \mapsto r.val])$

$2bs(a) \triangleq \{m \in sent : m.type = \text{"2b"} \wedge m.acc = a\}$

$max\_prop(a) \triangleq$

**IF**  $2bs(a) = \emptyset$  **THEN**  $\{[bal \mapsto -1, val \mapsto \perp]\}$

**ELSE**  $\{m \in 2bs(a) : \forall m2 \in 2bs(a) : m.bal \geq m2.bal\}$



$\forall msg \in sent : msg.type = \text{"1b"} \Rightarrow$

$\forall msg.maxVBal = -1$

$\forall \exists m \in sent :$

$\wedge m.type = \text{"2b"}$

$\wedge msg.acc = m.acc$

$\exists \wedge$   $msg.maxVBal = m.bal$

$\wedge msg.maxVal = m.val$



# 1b Message Invariant using max

$Phase1b(a \in \mathcal{A}) \triangleq$

$\exists m \in sent, r \in max\_prop(a) :$

$\wedge m.type = \text{"1a"}$

$\wedge \forall m2 \in sent : m2.type \in \{\text{"1b"}, \text{"2b"}\} \wedge$   
 $m2.acc = a \Rightarrow m.bal > m2.bal$

$\wedge Send([type \mapsto \text{"1b"},$   
 $acc \mapsto a, \underline{bal \mapsto m.bal},$   
 $\underline{maxVBal \mapsto r.bal},$   
 $\underline{maxVal \mapsto r.val}])$

$2bs(a) \triangleq \{m \in sent : m.type = \text{"2b"} \wedge m.acc = a\}$

$\underline{max\_prop(a) \triangleq}$

**IF**  $2bs(a) = \emptyset$  **THEN**  $\{[bal \mapsto -1, val \mapsto \perp]\}$

**ELSE**  $\{m \in 2bs(a) : \forall m2 \in 2bs(a) : \underline{m.bal \geq m2.bal}\}$



$\forall msg \in sent : msg.type = \text{"1b"} \Rightarrow$

$\forall \underline{b2 \in (msg.maxVBal, msg.bal)} :$

$\nexists m \in sent :$

$\wedge \underline{m.type = \text{"2b"}}$

$\wedge \underline{msg.acc = m.acc}$



$\wedge \underline{m.bal = b2}$



# 1b Message Invariant

$Phase1b(a \in \mathcal{A}) \triangleq$

$\exists m \in sent, r \in max\_prop(a) :$

$\wedge m.type = \text{"1a"}$

$\wedge \forall m2 \in sent : m2.type \in \{\text{"1b"}, \text{"2b"}\} \wedge$   
 $m2.acc = a \Rightarrow m.bal > m2.bal$

$\wedge Send([type \mapsto \text{"1b"},$   
 $acc \mapsto a, bal \mapsto m.bal,$   
 $maxVBal \mapsto r.bal,$   
 $maxVal \mapsto r.val])$

$2bs(a) \triangleq \{m \in sent : m.type = \text{"2b"} \wedge m.acc = a\}$

$max\_prop(a) \triangleq$

**IF**  $2bs(a) = \emptyset$  **THEN**  $\{[bal \mapsto -1, val \mapsto \perp]\}$

**ELSE**  $\{m \in 2bs(a) : \forall m2 \in 2bs(a) : m.bal \geq m2.bal\}$



$\forall msg \in sent : msg.type = \text{"1b"} \Rightarrow$

$\forall msg.maxVBal = -1$

$\forall \exists m \in sent :$

$\wedge m.type = \text{"2b"}$

$\wedge msg.acc = m.acc$

$\exists \wedge$   $msg.maxVBal = m.bal$

$\wedge msg.maxVal = m.val$

# 1b Message Invariant using max

$Phase1b(a \in \mathcal{A}) \triangleq$

$\exists m \in sent, r \in max\_prop(a) :$

$\wedge m.type = \text{"1a"}$

$\wedge \forall m2 \in sent : m2.type \in \{\text{"1b"}, \text{"2b"}\} \wedge$   
 $m2.acc = a \Rightarrow m.bal > m2.bal$

$\wedge Send([type \mapsto \text{"1b"},$   
 $acc \mapsto a, \underline{bal \mapsto m.bal},$   
 $\underline{maxVBal \mapsto r.bal},$   
 $\underline{maxVal \mapsto r.val}])$

$2bs(a) \triangleq \{m \in sent : m.type = \text{"2b"} \wedge m.acc = a\}$

$\underline{max\_prop(a) \triangleq}$

**IF**  $2bs(a) = \emptyset$  **THEN**  $\{[bal \mapsto -1, val \mapsto \perp]\}$

**ELSE**  $\{m \in 2bs(a) : \forall m2 \in 2bs(a) : \underline{m.bal \geq m2.bal}\}$



$\forall msg \in sent : msg.type = \text{"1b"} \Rightarrow$

$\forall \underline{b2 \in (msg.maxVBal, msg.bal)} :$

$\nexists m \in sent :$

$\wedge \underline{m.type = \text{"2b"}}$

$\wedge \underline{msg.acc = m.acc}$



$\wedge \underline{m.bal = b2}$

# Summary

- Distributed systems are complex and difficult to reason about.
- **Distributed consensus** is a fundamental problem and **Paxos** is a well-known algorithm for it.
- We discussed a formal specification and **safety** proof of **multi-value Paxos** in **TLA+** and **TLAPS** respectively.
- We discussed a systematic method that uses **history variables** to specify and verify distributed algorithms.





# 1b Message Invariant using max

$Phase1b(a \in \mathcal{A}) \triangleq$   
 $\exists m \in sent, r \in max\_prop(a) :$   
 $\wedge m.type = \text{"1a"}$   
 $\wedge \forall m2 \in sent : m2.type \in \{\text{"1b"}, \text{"2b"}\} \wedge$   
 $m2.acc = a \Rightarrow m.bal > m2.bal$   
 $\wedge Send([type \mapsto \text{"1b"},$   
 $acc \mapsto a, \underline{bal \mapsto m.bal},$   
 $\underline{maxVBal \mapsto r.bal},$   
 $\underline{maxVal \mapsto r.val}])$

$2bs(a) \triangleq \{m \in sent : m.type = \text{"2b"} \wedge m.acc = a\}$   
 $\underline{max\_prop(a) \triangleq}$   
 $\text{IF } 2bs(a) = \emptyset \text{ THEN } \{[bal \mapsto -1, val \mapsto \perp]\}$   
 $\text{ELSE } \{m \in 2bs(a) : \forall m2 \in 2bs(a) : \underline{m.bal \geq m2.bal}\}$



$\forall msg \in sent : msg.type = \text{"1b"} \Rightarrow$   
 $\forall \underline{b2 \in (msg.maxVBal, msg.bal)} :$   
 $\nexists m \in sent :$   
 $\wedge \underline{m.type = \text{"2b"}}$   
 $\wedge \underline{msg.acc = m.acc}$   
 $\wedge \underline{m.bal = b2}$

# Summary

- Distributed systems are complex and difficult to reason about.
- **Distributed consensus** is a fundamental problem and **Paxos** is a well-known algorithm for it.
- We discussed a formal specification and **safety** proof of **multi-value Paxos** in **TLA+** and **TLAPS** respectively.
- We discussed a systematic method that uses **history variables** to specify and verify distributed algorithms.

# THANKS!

Q + A





File Explorer window showing the Downloads folder. The address bar shows the path: Users > Sam > Downloads. The left sidebar shows the navigation pane with 'Downloads' selected. The main pane displays a list of files and folders organized by date.

| Name                                         | Date modified       |
|----------------------------------------------|---------------------|
| Today (5)                                    |                     |
| Unconfirmed 854190.crdownload                | 11/15/2019 10:23 AM |
| Spec Verif MP.pdf                            | 11/15/2019 10:22 AM |
| PaxosHistVarTLAPS_scratchpad.pdf             | 11/15/2019 8:33 AM  |
| PaxosTLAPS_work_scratchpad.pdf               | 11/15/2019 6:42 AM  |
| Spec Verif MP (1).pdf                        | 11/15/2019 10:23 AM |
| Yesterday (2)                                |                     |
| PaxosTLAPS_work (3).pdf                      |                     |
| SakshamChand_CV.pdf                          |                     |
| Earlier this week (11)                       |                     |
| proofs-master.zip                            |                     |
| Overleaf Projects (2 items).zip              |                     |
| PaxosHistVarTLAPS-work.zip                   |                     |
| g-1145.pdf                                   |                     |
| pgfmodulesshapes.code.tex                    |                     |
| output.log                                   | 11/11/2019 5:47 AM  |
| PaxosHistVarTLAPS_work.pdf                   | 11/11/2019 5:44 AM  |
| proofs-master                                | 11/13/2019 1:37 PM  |
| PaxosHistVarTLAPS-work                       | 11/13/2019 1:27 PM  |
| Saksham Academic CV and Cover Letter - Si... | 11/13/2019 11:32 AM |
| Overleaf Projects (2 items)                  | 11/13/2019 8:03 AM  |
| Last week (8)                                |                     |



Adobe Acrobat Pro DC interface showing the 'Export PDF' sidebar. The sidebar includes a search bar, a 'Share' button, and a 'Search 'Add Text'' input field. The 'Export PDF' section is active, showing options to convert PDF files to Word or Excel Online. The selected PDF file is 'Spec Verif MP (1).pdf'. The 'Convert to' dropdown is set to 'Microsoft Word (\*.docx)'. The 'Document Language' is set to 'English (U.S.)'. A large blue 'Convert' button is visible at the bottom of the sidebar. The main content area shows a document with a 64% zoom level and various editing tools.



TLA+ Toolbox

File Edit Window TLC Model Checker TLA Proof Manager Help

Spec Explorer

AIMD [ AIMD.tla ]

MC [ MC.tla ]

MultiPaxos [ MultiPaxos.tla ]

MultiPaxosCha [ MultiPaxosCh

MultiPaxosUs\_1 [ MultiPaxosU

MultiPaxosUs [ MultiPaxosUs.t

modules

FiniteSets

MultiPaxosUs

TLAPS

models

Model\_1

PaxosCopy [ PaxosCopy.tla ]

PaxosHistVarNFM18 [ PaxosH

PaxosLam [ PaxosLam.tla ]

MultiPaxos

C:\Users\Sam\Downloads\proofs-master\proofs-master\multi-paxos\MultiPaxos.tla

TLA Module

```
1 ----- MODULE MultiPaxos -----
2 (*****
3 (* This is a TLA+ specification and machine checked proof of the *)
4 (* MultiPaxos Consensus algorithm. *)
5 (*****
6 EXTENDS Integers, TLAPS
7 CONSTANTS Acceptors, Values, Quorums, Proposers
8 ASSUME QuorumAssumption == Quorums \subseteqq SUBSET Acceptors /\ \A Q1, Q2 \in Quorums
9 VARIABLES msgs,      /* Set of sent messages
10      aVoted,    /* Set of <<ballot, slot, value>> triples per acceptor that it has
11      aBal,      /* For each acceptor, the highest ballot seen by it.
12      pBal      /* Current ballot being run by proposers
13
14 Ballots == Nat
15 Slots == Nat
16 vars == <<msgs, aVoted, aBal, pBal>>
17 Send(m) == msgs' = msgs \cup {m}
18
19 (*****
20 (* Phase 1a: Executed by a proposer, it selects a ballot number on which *)
21 (* Phase 1a has never been initiated. This number is sent to any set of *)
22 (* acceptors which contains at least one quorum from Quorums. Trivially it *)
23 (* can be broadcasted to all Acceptors. For safety, any subset of *)
24 (* Acceptors would suffice. For liveness, a subset containing at least one *)
25 (* Quorum is needed. *)
26 (*****
27 Phase1a(p) == \E b \in Ballots:
28      /\ pBal' = [pBal EXCEPT! [p] = b]
29      /\ Send([tvpe |-> "1a". from |-> p. bal |-> b])
```

Parsing Errors

TLA+ Parser Error

line 121, col 35 to line 121, col 38 of module M

Unknown operator: 'tick'.

TLA+ Parser Error

line 125, col 20 to line 125, col 23 of module M

Unknown operator: 'tick'.

TLA+ Parser Error

line 125, col 12 to line 125, col 15 of module M

Unknown operator: 'tick'.

TLA+ Parser Error

line 126, col 92 to line 126, col 97 of module M

Unknown operator: 'mc\_log'.

TLA+ Parser Error

line 127, col 20 to line 127, col 25 of module M

Unknown operator: 'mc\_log'.

TLA+ Parser Error

line 128, col 29 to line 128, col 34 of module M

Unknown operator: 'mc\_log'.

TLA+ Parser Error

line 128, col 46 to line 128, col 49 of module M

Unknown operator: 'tick'.

Spec Status : error

Type here to search

11:16 AM 11/15/2019



TLA+ Toolbox

File Edit Window TLC Model Checker TLA Proof Manager Help

Spec Explorer

- AIMD [ AIMD.tla ]
- MC [ MC.tla ]
- MultiPaxos [ MultiPaxos.tla ]
- MultiPaxosCha [ MultiPaxosCha.tla ]
- MultiPaxosUs\_1 [ MultiPaxosUs.tla ]
- MultiPaxosUs [ MultiPaxosUs.tla ]
- PaxosCopy [ PaxosCopy.tla ]
- PaxosHistVarNFM18 [ PaxosHistVarNFM18.tla ]
  - modules
    - NaturalsInduction
    - PaxosHistVarNFM18
    - TLAPS
  - models
- PaxosLam [ PaxosLam.tla ]

PaxosHistVarNFM18

C:\Users\Sam\Downloads\proofs-master\proofs-master\basic-paxos\PaxosHistVarNFM18.tla

TLA Module

```
130 (*****)
131 (* WontVoteIn(a, b) is a predicate that implies that a has not voted and *)
132 (* never will vote in ballot b. *)
133 (*****)
134 WontVoteIn(a, b) == /\ \A v \in Values : ~ VotedForIn(a, v, b)
135                      /\ \E m \in sent: m.type \in {"1b", "2b"} /\ m.acc = a /\ m.bal > b
136
137 (*****)
138 (* The predicate SafeAt(v, b) implies that no value other than perhaps v *)
139 (* has been or ever will be chosen in any ballot numbered less than b. *)
140 (*****)
141 SafeAt(v, b) ==
142   \A b2 \in 0..(b-1) :
143     \E Q \in Quorums :
144       \A a \in Q : VotedForIn(a, v, b2) \/ WontVoteIn(a, b2)
145
146 MsgInv ==
147   \A m \in sent :
148     /\ m.type = "1b" => /\ VotedForIn(m.acc, m.maxVal, m.maxVbal) \/ m.maxVbal = -1
149                        /\ \A b \in m.maxVbal+1..m.bal-1: ~\E v \in Values: VotedForIn(m.acc, v, b)
150     /\ m.type = "2a" => /\ SafeAt(m.val, m.bal)
151                        /\ \A m2 \in sent : (m2.type = "2a" /\ m2.bal = m.bal) => m2 = m
152     /\ m.type = "2b" => \E m2 \in sent : /\ m2.type = "2a"
153                                           /\ m2.bal = m.bal
154                                           /\ m2.val = m.val
155
156 Inv == TypeOK /\ MsgInv
157
158 (*****)
```

141:1

Spec Status: parsed

Type here to search

11:17 AM 11/15/2019





50



51



52

## 1b Message Invariant using max

$$\text{Phase1b}(a \in \mathcal{A}) \triangleq$$

$$\exists m \in \text{sent}, r \in \text{max\_prop}(a) :$$

$$\wedge m.\text{type} = \text{"1a"}$$

$$\wedge \forall m2 \in \text{sent} : m2.\text{type} \in \{\text{"1b"}, \text{"2b"}\} \wedge$$

$$m2.\text{acc} = a \Rightarrow m.\text{bal} > m2.\text{bal}$$

$$\wedge \text{Send}([ \text{type} \mapsto \text{"1b"},$$

$$\text{acc} \mapsto a, \text{bal} \mapsto m.\text{bal},$$

$$\text{maxVbal} \mapsto r.\text{bal},$$

$$\text{maxVal} \mapsto r.\text{val} ] )$$

$$2bs(a) \triangleq \{ m \in \text{sent} : m.\text{type} = \text{"2b"} \wedge m.\text{acc} = a \}$$

$$\text{max\_prop}(a) \triangleq$$

$$\text{IF } 2bs(a) = \emptyset \text{ THEN } \{ [ \text{bal} \mapsto -1, \text{val} \mapsto \perp ] \}$$

$$\text{ELSE } \{ m \in 2bs(a) : \forall m2 \in 2bs(a) : m.\text{bal} \geq m2.\text{bal} \}$$

$$\forall \text{msg} \in \text{sent} : \text{msg.type} = \text{"1b"} \Rightarrow$$

$$\forall b2 \in (\text{msg.maxVbal}, \text{msg.bal}) :$$

$$\exists m \in \text{sent} :$$

$$\wedge m.\text{type} = \text{"2b"}$$

$$\wedge \text{msg.acc} = m.\text{acc}$$

$$\wedge m.\text{bal} = b2$$

Search 'Add Text'

 Export PDF

Adobe Export PDF

Convert PDF Files to Word or Excel Online

Select PDF File

Spec Verif MP (1).pdf X

Convert to

Microsoft Word (\*.docx) v

Document Language:

English (U.S.) Change

Convert

Convert and edit PDFs with Acrobat Pro DC

Start Free Trial



TLA+ Toolbox

File Edit Window TLC Model Checker TLA Proof Manager Help

Spec Explorer

- AIMD [ AIMD.tla ]
- MC [ MC.tla ]
- MultiPaxos [ MultiPaxos.tla ]
- MultiPaxosCha [ MultiPaxosCha.tla ]
- MultiPaxosUs\_1 [ MultiPaxosUs.tla ]
- MultiPaxosUs [ MultiPaxosUs.tla ]
- PaxosCopy [ PaxosCopy.tla ]
- PaxosHistVarNFM18 [ PaxosHistVarNFM18.tla ]
  - modules
    - NaturalsInduction
    - PaxosHistVarNFM18
    - TLAPS
  - models
- PaxosLam [ PaxosLam.tla ]

PaxosHistVarNFM18

C:\Users\Sam\Downloads\proofs-master\proofs-master\basic-paxos\PaxosHistVarNFM18.tla

TLA Module

```
130 (*****)
131 (* WontVoteIn(a, b) is a predicate that implies that a has not voted and *)
132 (* never will vote in ballot b. *)
133 (*****)
134 WontVoteIn(a, b) == /\ \A v \in Values : ~ VotedForIn(a, v, b)
135                      /\ \E m \in sent: m.type \in {"1b", "2b"} /\ m.acc = a /\ m.bal > b
136
137 (*****)
138 (* The predicate SafeAt(v, b) implies that no value other than perhaps v *)
139 (* has been or ever will be chosen in any ballot numbered less than b. *)
140 (*****)
141 SafeAt(v, b) ==
142   \A b2 \in 0..(b-1) :
143     \E Q \in Quorums :
144       \A a \in Q : VotedForIn(a, v, b2) /\ WontVoteIn(a, b2)
145
146 MsgInv ==
147   \A m \in sent :
148     /\ m.type = "1b" => /\ VotedForIn(m.acc, m.maxVal, m.maxVbal) /\ m.maxVbal = -1
149                        /\ \A b \in m.maxVbal+1..m.bal-1: ~\E v \in Values: VotedForIn(m.acc, v, b)
150     /\ m.type = "2a" => /\ SafeAt(m.val, m.bal)
151                        /\ \A m2 \in sent : (m2.type = "2a" /\ m2.bal = m.bal) => m2 = m
152     /\ m.type = "2b" => \E m2 \in sent : /\ m2.type = "2a"
153                                           /\ m2.bal = m.bal
154                                           /\ m2.val = m.val
155
156 Inv == TypeOK /\ MsgInv
157
158
```

145 : 1

Spec Status : parsed

Type here to search

11:20 AM 11/15/2019



TLA+ Toolbox

File Edit Window TLC Model Checker TLA Proof Manager Help

Spec Explorer

- AIMD [ AIMD.tla ]
- MC [ MC.tla ]
- MultiPaxos [ MultiPaxos.tla ]
- MultiPaxosCha [ MultiPaxosCha.tla ]
- MultiPaxosUs\_1 [ MultiPaxosUs.tla ]
- MultiPaxosUs [ MultiPaxosUs.tla ]
- PaxosCopy [ PaxosCopy.tla ]
- PaxosHistVarNFM18 [ PaxosHistVarNFM18.tla ]
  - modules
    - NaturalsInduction
    - PaxosHistVarNFM18
    - TLAPS
  - models
  - PaxosLam [ PaxosLam.tla ]

PaxosHistVarNFM18

C:\Users\Sam\Downloads\proofs-master\proofs-master\basic-paxos\PaxosHistVarNFM18.tla

TLA Module

```
346 <1>3. QED
347 BY <1>1, <1>2, PTL DEF Spec
348
349 THEOREM Consistent == Spec => []Consistency
350 <1> USE DEF Ballots
351
352 <1>1. Inv => Consistency
353 <2> SUFFICES ASSUME Inv,
354     NEW v1 \in Values, NEW v2 \in Values,
355     NEW b1 \in Ballots, NEW b2 \in Ballots,
356     ChosenIn(v1, b1), ChosenIn(v2, b2),
357     b1 =< b2
358     PROVE v1 = v2
359 BY DEF Consistency, Chosen
360 <2>1. CASE b1 = b2
361 BY <2>1, VotedOnce, QuorumAssumption, SMTT(100) DEF ChosenIn, Inv
362 <2>2. CASE b1 < b2
363 <3>1. SafeAt(v2, b2)
364 BY VotedInv, QuorumAssumption DEF ChosenIn, Inv
365 <3>2. PICK Q2 \in Quorums :
366     \A a \in Q2 : VotedForIn(a, v2, b1) /\ WontVoteIn(a, b1)
367 BY <3>1, <2>2 DEF SafeAt
368 <3>3. PICK Q1 \in Quorums : \A a \in Q1 : VotedForIn(a, v1, b1)
369 BY DEF ChosenIn
370 <3>4. QED
371 BY <3>2, <3>3, QuorumAssumption, VotedOnce, Z3 DEF WontVoteIn, Inv
372 <2>3. QED
373 BY <2>1, <2>2
374
```

362 : 20

Spec Status : parsed

Type here to search

11:23 AM 11/15/2019



TLA+ Toolbox

File Edit Window TLC Model Checker TLA Proof Manager Help

Spec Explorer

AIMD [ AIMD.tla ]

MC [ MC.tla ]

MultiPaxos [ MultiPaxos.tla ]

MultiPaxosCha [ MultiPaxosCha.tla ]

MultiPaxosUs\_1 [ MultiPaxosUs.tla ]

MultiPaxosUs [ MultiPaxosUs.tla ]

PaxosCopy [ PaxosCopy.tla ]

PaxosHistVarNFM18 [ PaxosHistVarNFM18.tla ]

modules

NaturalsInduction

PaxosHistVarNFM18

TLAPS

models

PaxosLam [ PaxosLam.tla ]

PaxosHistVarNFM18

C:\Users\Sam\Downloads\proofs-master\proofs-master\basic-paxos\PaxosHistVarNFM18.tla

TLA Module

171 -----

172 (\*\*\*\*\*)

173 (\* The following lemma shows that (the invariant implies that) the \*)

174 (\* predicate SafeAt(v, b) is stable, meaning that once it becomes true, it \*)

175 (\* remains true throughout the rest of the execution. \*)

176 (\*\*\*\*\*)

177 **LEMMA** SafeAtStable == Inv /\ Next =>

178                      \A v \in Values, b \in Ballots:

179                      SafeAt(v, b) => SafeAt(v, b)'

180 <1> **SUFFICES ASSUME** Inv, Next,

181                      **NEW** v \in Values, **NEW** b \in Ballots, SafeAt(v, b)

182                      **PROVE** SafeAt(v, b)'

183 **OBVIOUS**

184 <1> **USE DEF** Send, Inv, Ballots

185 <1> **USE TRUE** /\ **TRUE**

186 <1>1. **ASSUME** **NEW** bb \in Ballots, Phase1a(bb)

187                      **PROVE** SafeAt(v, b)'

188 **BY** <1>1, SMT **DEF** SafeAt, Phase1a, VotedForIn, WontVoteIn

189 <1>2. **ASSUME** **NEW** a \in Acceptors, Phase1b(a)

190                      **PROVE** SafeAt(v, b)'

191 **BY** <1>2, QuorumAssumption, SMTT(60) **DEF** TypeOK, SafeAt, WontVoteIn, VotedForIn, Phase1b

192 <1>3. **ASSUME** **NEW** bb \in Ballots, Phase2a(bb)

193                      **PROVE** SafeAt(v, b)'

194 **BY** <1>3, QuorumAssumption, SMT **DEF** TypeOK, SafeAt, WontVoteIn, VotedForIn, Phase2a

195 <1>4. **ASSUME** **NEW** a \in Acceptors, Phase2b(a)

196                      **PROVE** SafeAt(v, b)'

197 <2>1. **PICK** m \in sent : Phase2b(a)!(m)

198 **BY** <1>4 **DEF** Phase2b

199 <2>2 \A aa \in Acceptors, bb \in Ballots, vv \in Values :

150 : 7

Spec Status : parsed

Windows Taskbar

Search: Type here to search

Taskbar Icons

System Tray



```

2204  /2\3  ASCII NEW a \in Accentors  Phase1b(a) DDIVE TypeOK!

```



TLA+ Toolbox

File Edit Window TLC Model Checker TLA Proof Manager Help

Spec Explorer

AIMD [ AIMD.tla ]

MC [ MC.tla ]

MultiPaxos [ MultiPaxos.tla ]

MultiPaxosCha [ MultiPaxosCha.tla ]

MultiPaxosUs\_1 [ MultiPaxosUs.tla ]

MultiPaxosUs [ MultiPaxosUs.tla ]

PaxosCopy [ PaxosCopy.tla ]

PaxosHistVarNFM18 [ PaxosHistVarNFM18 ]

modules

NaturalsInduction

PaxosHistVarNFM18

TLAPS

models

PaxosLam [ PaxosLam.tla ]

PaxosHistVarNFM18

C:\Users\Sam\Downloads\proofs-master\proofs-master\basic-paxos\PaxosHistVarNFM18.tla

TLA Module

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

~ votedForIn(mmm.acc, v, c) => ~ votedForIn(mmm.acc, v, c)

BY DEF Send, VotedForIn, MsgInv, TypeOK, Messages

<4>. QED

BY <4>1, <4>2, SafeAtStable, <2>1 DEF MsgInv, Send, TypeOK, Messages

<3>5. QED

BY <3>1, <3>2, <3>3, <3>4 DEF Next

<2>4. QED

BY <2>1, <2>3 DEF Inv

<1>3. QED

BY <1>1, <1>2, PTL DEF Spec

THEOREM Consistent == Spec => []Consistency

<1> USE DEF Ballots

<1>1. Inv => Consistency

<2> SUFFICES ASSUME Inv,

NEW v1 \in Values, NEW v2 \in Values,

NEW b1 \in Ballots, NEW b2 \in Ballots,

ChosenIn(v1, b1), ChosenIn(v2, b2),

b1 =< b2

PROVE v1 = v2

BY DEF Consistency, Chosen

<2>1. CASE b1 = b2

BY <2>1, VotedOnce, QuorumAssumption, SMTT(100) DEF ChosenIn, Inv

<2>2. CASE b1 < b2

<3>1. SafeAt(v2, b2)

BY VotedInv, QuorumAssumption DEF ChosenIn, Inv

<3>2. PICK Q2 \in Quorums :

\A a \in Q2 : VotedForIn(a, v2, b1) /\ WontVoteIn(a, b1)

221 : 35

Spec Status : parsed

Type here to search

11:24 AM 11/15/2019



TLA+ Toolbox

File Edit Window TLC Model Checker TLA Proof Manager Help

Spec Explorer

- AIMD [ AIMD.tla ]
- MC [ MC.tla ]
- MultiPaxos [ MultiPaxos.tla ]
- MultiPaxosCha [ MultiPaxosCha.tla ]
- MultiPaxosUs\_1 [ MultiPaxosUs.tla ]
- MultiPaxosUs [ MultiPaxosUs.tla ]
- PaxosCopy [ PaxosCopy.tla ]
- PaxosHistVarNFM18 [ PaxosHistVarNFM18.tla ]
  - modules
    - NaturalsInduction
    - PaxosHistVarNFM18
    - TLAPS
  - models
- PaxosLam [ PaxosLam.tla ]

PaxosHistVarNFM18

C:\Users\Sam\Downloads\proofs-master\proofs-master\basic-paxos\PaxosHistVarNFM18.tla

TLA Module

```
349 THEOREM Consistent == Spec => []Consistency
350 <1> USE DEF Ballots
351
352 <1>1. Inv => Consistency
353 <2> SUFFICES ASSUME Inv,
354     NEW v1 \in Values, NEW v2 \in Values,
355     NEW b1 \in Ballots, NEW b2 \in Ballots,
356     ChosenIn(v1, b1), ChosenIn(v2, b2),
357     b1 =< b2
358     PROVE v1 = v2
359     BY DEF Consistency, Chosen
360 <2>1. CASE b1 = b2
361     BY <2>1, VotedOnce, QuorumAssumption, SMTT(100) DEF ChosenIn, Inv
362 <2>2. CASE b1 < b2
363 <3>1. SafeAt(v2, b2)
364     BY VotedInv, QuorumAssumption DEF ChosenIn, Inv
365 <3>2. PICK Q2 \in Quorums :
366     \A a \in Q2 : VotedForIn(a, v2, b1) \/ WontVoteIn(a, b1)
367     BY <3>1, <2>2 DEF SafeAt
368 <3>3. PICK Q1 \in Quorums : \A a \in Q1 : VotedForIn(a, v1, b1)
369     BY DEF ChosenIn
370 <3>4. QED
371     BY <3>2, <3>3, QuorumAssumption, VotedOnce, Z3 DEF WontVoteIn, Inv
372 <2>3. QED
373     BY <2>1, <2>2
374
375 <1>2. QED
376     BY Invariant, <1>1, PTL
```

221 : 35

Spec Status : parsed

Type here to search

11:24 AM 11/15/2019



TLA+ Toolbox

File Edit Window TLC Model Checker TLA Proof Manager Help

Spec Explorer

- AIMD [ AIMD.tla ]
- MC [ MC.tla ]
- MultiPaxos [ MultiPaxos.tla ]
- MultiPaxosCha [ MultiPaxosCha.tla ]
- MultiPaxosUs\_1 [ MultiPaxosUs.tla ]
- MultiPaxosUs [ MultiPaxosUs.tla ]
- PaxosCopy [ PaxosCopy.tla ]
- PaxosHistVarNFM18 [ PaxosHistVarNFM18.tla ]
  - modules
    - NaturalsInduction
    - PaxosHistVarNFM18
    - TLAPS
  - models
- PaxosLam [ PaxosLam.tla ]

PaxosHistVarNFM18

C:\Users\Sam\Downloads\proofs-master\proofs-master\basic-paxos\PaxosHistVarNFM18.tla

TLA Module

```
67 /\ ~ \E m \in sent : (m.type = "2a") /\ (m.bal = b)
68 /\ \E v \in Values, Q \in Quorums, S \in SUBSET {m \in sent : m.type = "1b" /\ m.bal = b}:
69   /\ \A a \in Q : \E m \in S : m.acc = a
70   /\ \A m \in S : m.maxVBal = -1
71   /\ \E c \in 0..(b-1) :
72     /\ \A m \in S : m.maxVBal =< c
73     /\ \E m \in S : /\ m.maxVBal = c
74                       /\ m.maxVal = v
75   /\ Send([type |-> "2a", bal |-> b, val |-> v])
76
77 (*****)
78 (* Phase 2b: If an acceptor receives a 2a message for a ballot numbered *)
79 (* b, it votes for the message's value in ballot b unless it has already *)
80 (* responded to a 1a request for a ballot number greater than or equal to *)
81 (* b. *)
82 (*****)
83 Phase2b(a) ==
84   \E m \in sent :
85     /\ m.type = "2a"
86     /\ \A m2 \in sent: m2.type \in {"1b", "2b"} /\ m2.acc = a => m.bal >= m2.bal
87     /\ Send([type |-> "2b", bal |-> m.bal, val |-> m.val, acc |-> a])
88
89 Next == \A b \in Ballots : Phase1a(b) /\ Phase2a(b)
90         /\ \A a \in Acceptors : Phase1b(a) /\ Phase2b(a)
91
92 Spec == Init /\ [][Next]_vars
93 -----
94 (*****)
```

361 : 17

Spec Status : **parsed**

Type here to search

11:25 AM 11/15/2019



TLA+ Toolbox

File Edit Window TLC Model Checker TLA Proof Manager Help

Spec Explorer

- AIMD [ AIMD.tla ]
- MC [ MC.tla ]
- MultiPaxos [ MultiPaxos.tla ]
- MultiPaxosCha [ MultiPaxosCha.tla ]
- MultiPaxosUs\_1 [ MultiPaxosUs.tla ]
- MultiPaxosUs [ MultiPaxosUs.tla ]
- PaxosCopy [ PaxosCopy.tla ]
- PaxosHistVarNFM18 [ PaxosHistVarNFM18.tla ]
  - modules
    - NaturalsInduction
    - PaxosHistVarNFM18
    - TLAPS
  - models
  - PaxosLam [ PaxosLam.tla ]

PaxosHistVarNFM18

C:\Users\Sam\Downloads\proofs-master\proofs-master\basic-paxos\PaxosHistVarNFM18.tla

TLA Module

```
32 (*****)
```

```
33 (* Phase 1a: A leader selects a ballot number b and sends a 1a message *)
```

```
34 (* with ballot b to a majority of acceptors. It can do this only if it *)
```

```
35 (* has not already sent a 1a message for ballot b. *)
```

```
36 (*****)
```

```
37 Phase1a(b) == Send([type |-> "1a", bal |-> b])
```

```
38
```

```
39 (*****)
```

```
40 (* Phase 1b: If an acceptor receives a 1a message with ballot b greater *)
```

```
41 (* than that of any 1a message to which it has already responded, then it *)
```

```
42 (* responds to the request with a promise not to accept any more proposals *)
```

```
43 (* for ballots numbered less than b and with the highest-numbered ballot *)
```

```
44 (* (if any) for which it has voted for a value and the value it voted for *)
```

```
45 (* in that ballot. That promise is made in a 1b message. *)
```

```
46 (*****)
```

```
47 last_voted(a) == LET 2bs == {m \in sent: m.type = "2b" /\ m.acc = a}
```

```
48                      IN IF 2bs # {} THEN {m \in 2bs: \A m2 \in 2bs: m.bal >= m2.bal}
```

```
49                      ELSE {[bal |-> -1, val |-> None]}
```

```
50
```

```
51 Phase1b(a) ==
```

```
52   \E m \in sent, r \in last_voted(a):
```

```
53     /\ m.type = "1a"
```

```
54     /\ \A m2 \in sent: m2.type \in {"1b", "2b"} /\ m2.acc = a => m.bal > m2.bal
```

```
55     /\ Send([type |-> "1b", bal |-> m.bal,
```

```
56               maxVBal |-> r.bal, maxVal |-> r.val, acc |-> a])
```

```
57
```

```
58 (*****)
```

```
59 (* Phase 2a: If the leader receives a response to its 1b message (for *)
```

```
60 (* ballot b) from a quorum of acceptors, then it sends a 2a message to all *)
```

54 : 81

Spec Status : parsed

Type here to search

11:28 AM 11/15/2019





50



51



52

## 1b Message Invariant using max

$$\text{Phase1b}(a \in \mathcal{A}) \triangleq$$

$$\exists m \in \text{sent}, r \in \text{max\_prop}(a) :$$

$$\wedge m.\text{type} = \text{"1a"}$$

$$\wedge \forall m2 \in \text{sent} : m2.\text{type} \in \{\text{"1b"}, \text{"2b"}\} \wedge$$

$$m2.\text{acc} = a \Rightarrow m.\text{bal} > m2.\text{bal}$$

$$\wedge \text{Send}([ \text{type} \mapsto \text{"1b"},$$

$$\text{acc} \mapsto a, \text{bal} \mapsto m.\text{bal},$$

$$\text{maxVBal} \mapsto r.\text{bal},$$

$$\text{maxVal} \mapsto r.\text{val} ] )$$

$$2bs(a) \triangleq \{ m \in \text{sent} : m.\text{type} = \text{"2b"} \wedge m.\text{acc} = a \}$$

$$\text{max\_prop}(a) \triangleq$$

$$\text{IF } 2bs(a) = \emptyset \text{ THEN } \{ [ \text{bal} \mapsto -1, \text{val} \mapsto \perp ] \}$$

$$\text{ELSE } \{ m \in 2bs(a) : \forall m2 \in 2bs(a) : m.\text{bal} \geq m2.\text{bal} \}$$

$$\forall \text{msg} \in \text{sent} : \text{msg.type} = \text{"1b"} \Rightarrow$$

$$\forall b2 \in (\text{msg.maxVBal}, \text{msg.bal}) :$$

$$\exists m \in \text{sent} :$$

$$\wedge m.\text{type} = \text{"2b"}$$

$$\wedge \text{msg.acc} = m.\text{acc}$$

$$\wedge m.\text{bal} = b2$$

Search 'Add Text'

 Export PDF

Adobe Export PDF

Convert PDF Files to Word or Excel Online

Select PDF File

Spec Verif MP (1).pdf X

Convert to

Microsoft Word (\*.docx) v

Document Language:

English (U.S.) Change

Convert

Convert and edit PDFs with Acrobat Pro DC

Start Free Trial



TLA+ Toolbox

File Edit Window TLC Model Checker TLA Proof Manager Help

Spec Explorer

AIMD [ AIMD.tla ]

MC [ MC.tla ]

MultiPaxos [ MultiPaxos.tla ]

MultiPaxosCha [ MultiPaxosCha.tla ]

MultiPaxosUs\_1 [ MultiPaxosUs.tla ]

MultiPaxosUs [ MultiPaxosUs.tla ]

PaxosCopy [ PaxosCopy.tla ]

PaxosHistVarNFM18 [ PaxosHistVarNFM18.tla ]

modules

NaturalsInduction

PaxosHistVarNFM18

TLAPS

models

PaxosLam [ PaxosLam.tla ]

PaxosHistVarNFM18

C:\Users\Sam\Downloads\proofs-master\proofs-master\basic-paxos\PaxosHistVarNFM18.tla

TLA Module

```
44 (* (if any) for which it has voted for a value and the value it voted for *)
45 (* in that ballot. That promise is made in a 1b message. *)
46 (*****)
47 last_voted(a) == LET 2bs == {m \in sent: m.type = "2b" /\ m.acc = a}
48                  IN IF 2bs # {} THEN {m \in 2bs: \A m2 \in 2bs: m.bal >= m2.bal}
49                  ELSE {[bal |-> -1, val |-> None]}
50
51 Phase1b(a) ==
52   \E m \in sent, r \in last_voted(a):
53     /\ m.type = "1a"
54     /\ \A m2 \in sent: m2.type \in {"1b", "2b"} /\ m2.acc = a => m.bal > m2.bal
55     /\ Send([type |-> "1b", bal |-> m.bal,
56             maxVBal |-> r.bal, maxVal |-> r.val, acc |-> a])
57
58 (*****)
59 (* Phase 2a: If the leader receives a response to its 1b message (for *)
60 (* ballot b) from a quorum of acceptors, then it sends a 2a message to all *)
61 (* acceptors for a proposal in ballot b with a value v, where v is the *)
62 (* value of the highest-numbered proposal among the responses, or is any *)
63 (* value if the responses reported no proposals. The leader can send only *)
64 (* one 2a message for any ballot. *)
65 (*****)
66 Phase2a(b) ==
67   /\ ~ \E m \in sent : (m.type = "2a") /\ (m.bal = b)
68   /\ \E v \in Values, Q \in Quorums, S \in SUBSET {m \in sent : m.type = "1b" /\ m.bal = b}:
69     /\ \A a \in Q : \E m \in S : m.acc = a
70     /\ \A m \in S : m.maxVBal = -1
71     /\ \E c \in 0..(b-1) :
72       /\ \A m \in S : m.maxVBal = c
```

53 : 22

Spec Status : parsed

Windows Start Search

Taskbar Icons

System Tray