Inferring Inductive Invariants from Phase Structures

Yotam Feldman James R. Wilcox Sharon Shoham Mooly Sagiv

TEL AVIV UNIVERSITY אוניברסיטת תל אביב

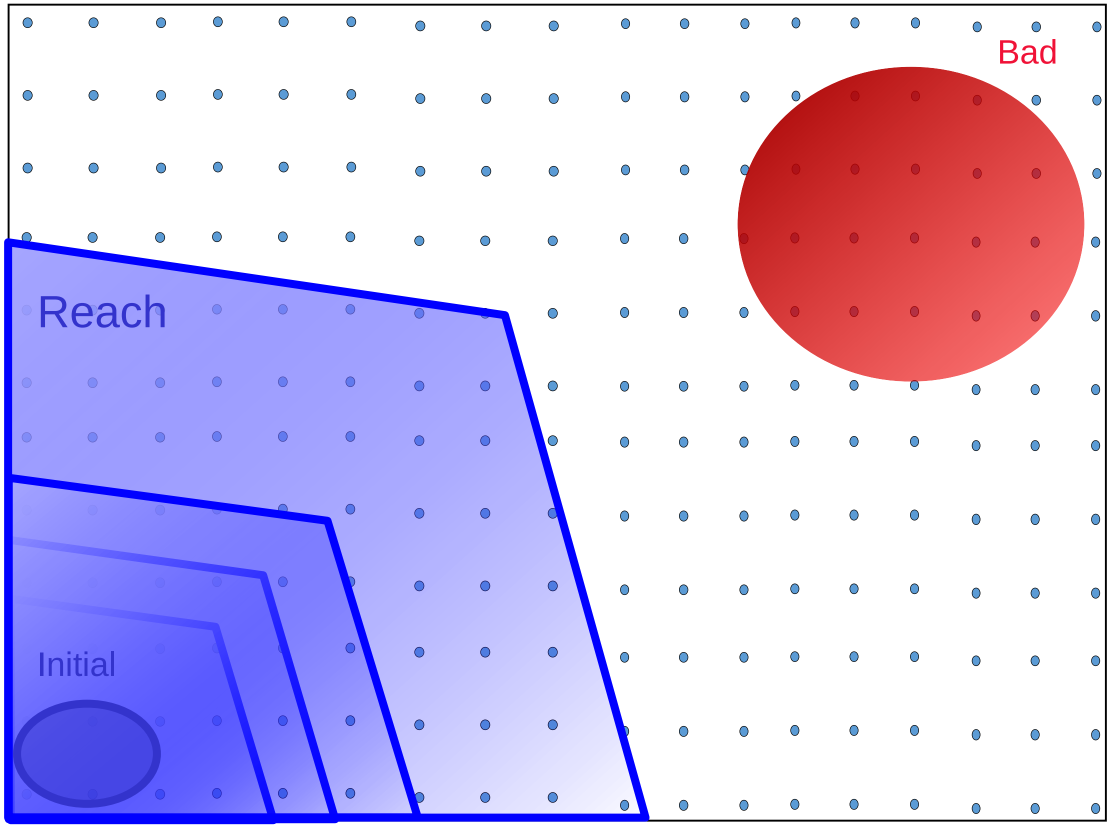
UNIVERSITY of WASHINGTON

TEL AVIV UNIVERSITY אוניברסיטת תל אביב

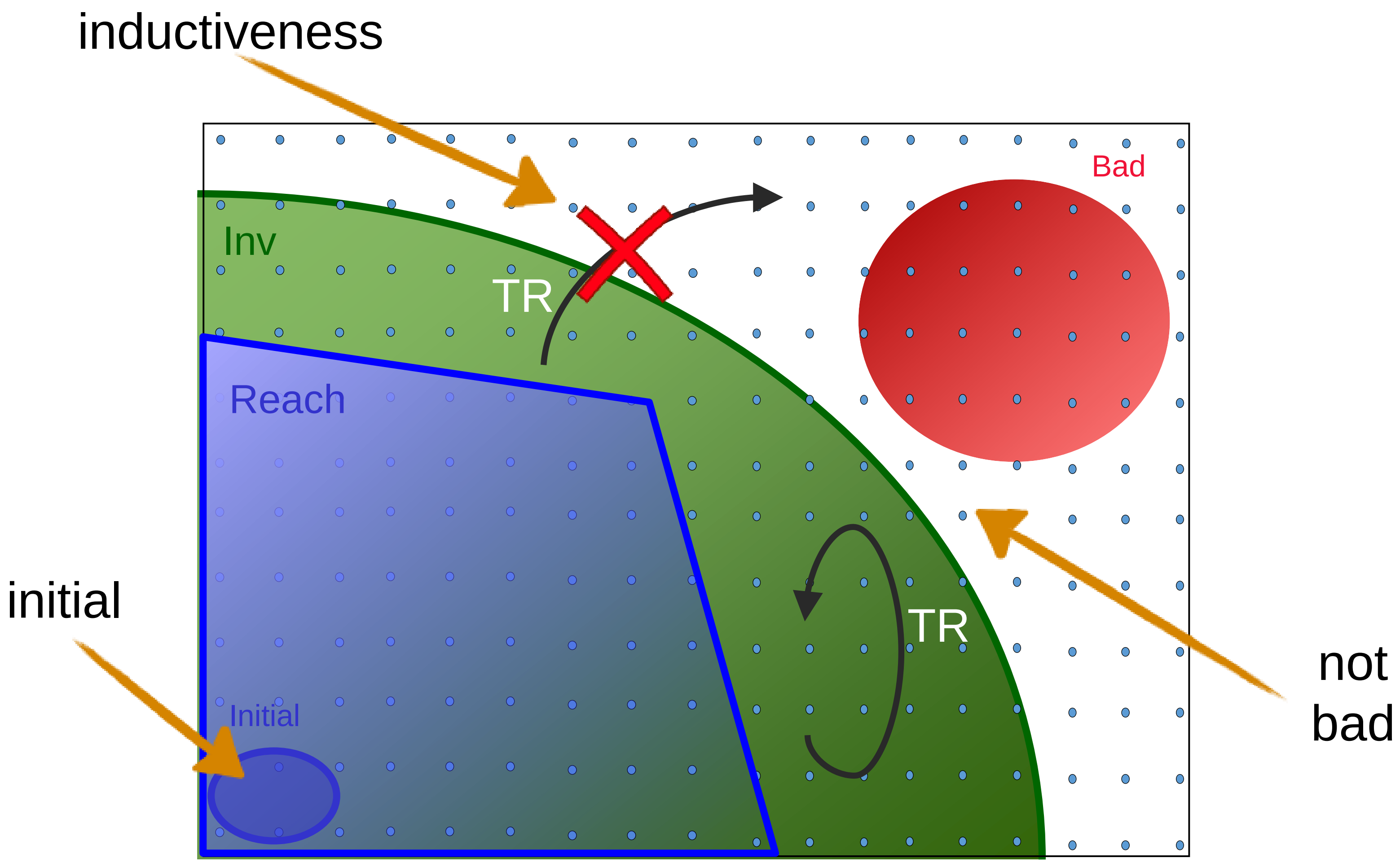
TEL AVIV UNIVERSITY אוניברסיטת תל אביב

@yotamfe, @wilcoxjay, @SagivMooly

Safety of Infinite-State Systems



Inductive Invariants



Distributed Protocols in EPR

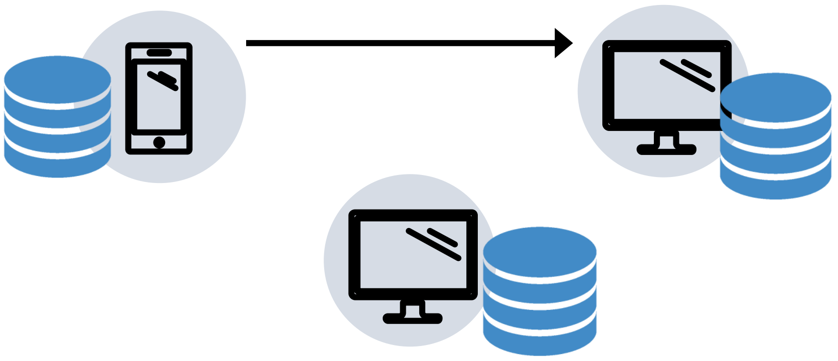
EPR: A decidable fragment of first order logic

Used for modelling distributed protocols

[Padon et al. PLDI'16, OOPSLA'17, POPL'18, Taube et al. PLDI'18, Berkovits et al. CAV'19]

Our focus: **universally quantified invariants for EPR distributed protocols**

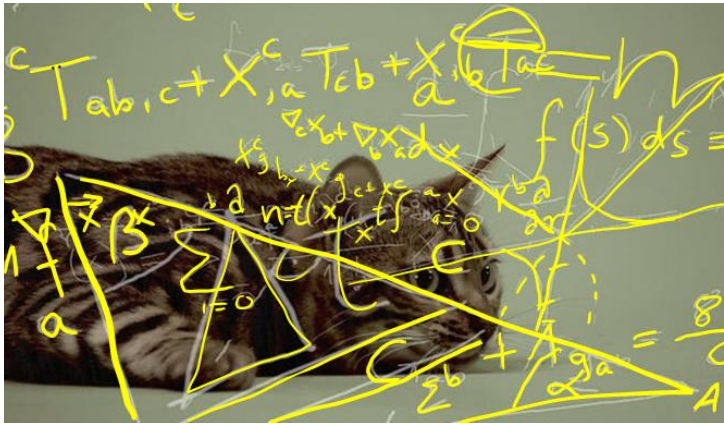
$\forall$



Proving with Inductive Invariants

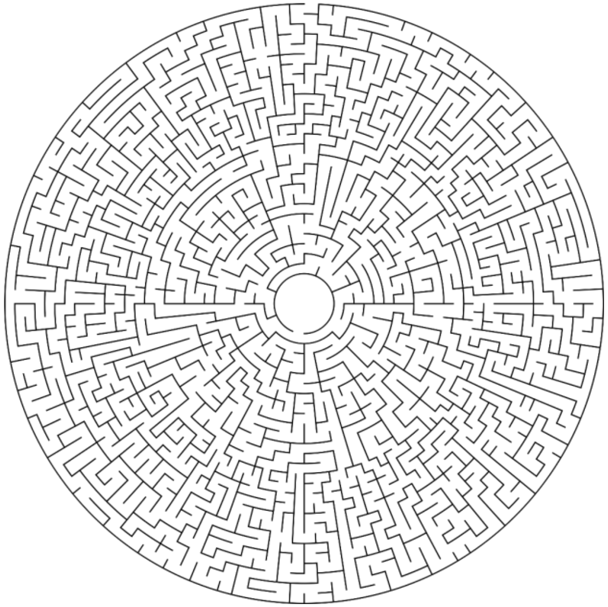
Deductive verification –
manually specify inductive invariant

Labor intensive



Invariant inference –
automatically search for invariant

Limited and fragile



Our Approach

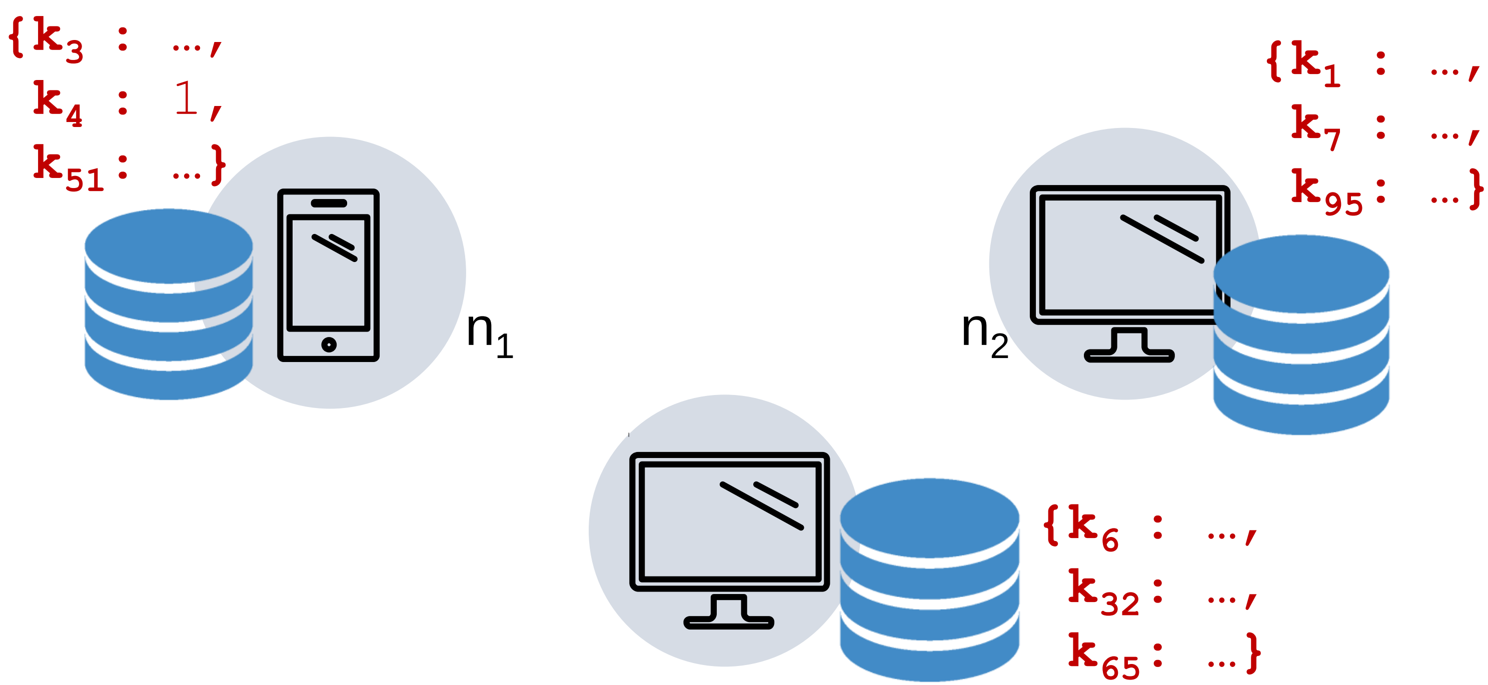
User-guided invariant inference – **manually** specify high-level **intuition**, **automatically** find **full** proof

1. Guide invariant inference using **phase structures**
2. Apply to inference of **universally quantified** invariants on **challenging distributed protocols** modelled in EPR

Example: Sharded Key-Value Store

State: modeled over global relations

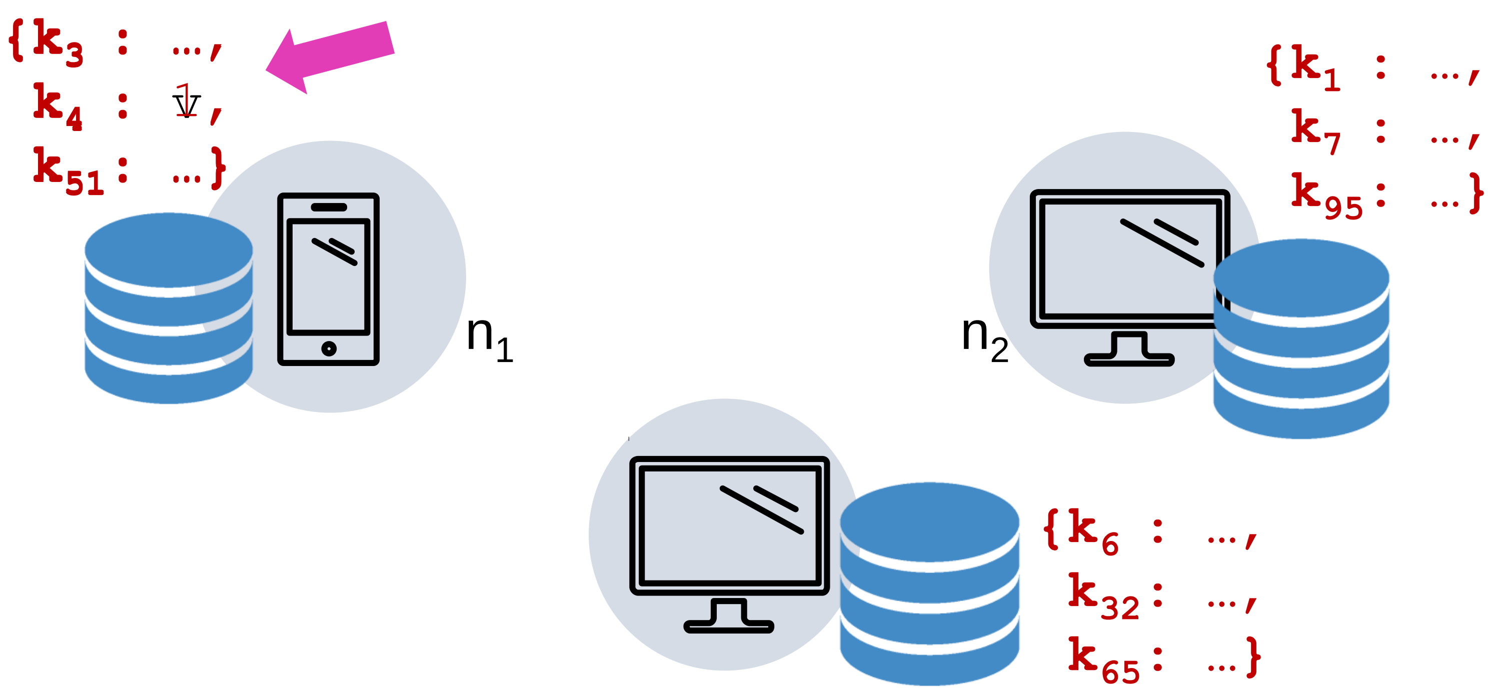
- local state
- network



[SOSP15] IronFleet: Proving Practical Distributed Systems Correct. C. Hawblitzel, J. Howell, M. Kapritsos, J.R. Lorch, B. Parno, M.L. Roberts, S. Setty, and B. Zill

Example: Sharded Key-Value Store

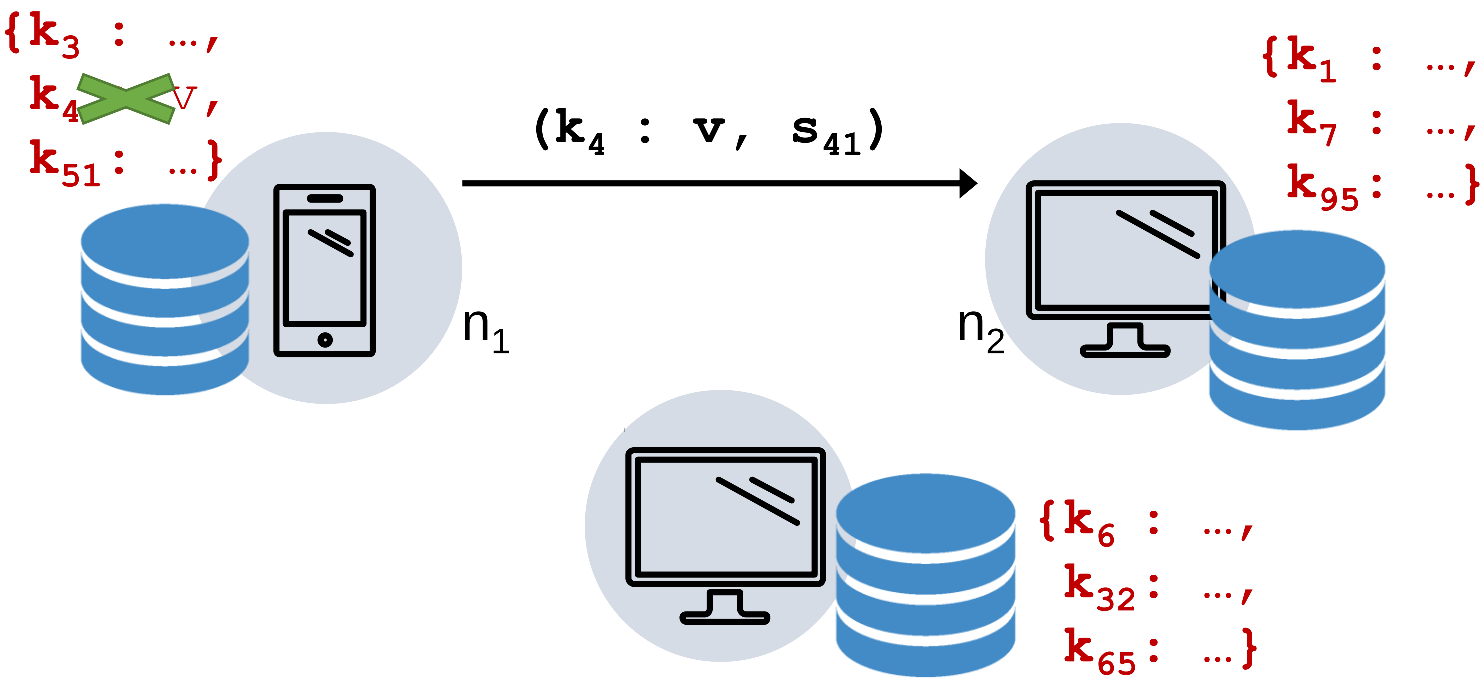
change local table:
table(n_1, k_4) := v



[SOSP15] IronFleet: Proving Practical Distributed Systems Correct. C. Hawblitzel, J. Howell, M. Kapritsos, J.R. Lorch, B. Parno, M.L. Roberts, S. Setty, and B. Zill

Example: Sharded Key-Value Store

reshard:
table(n_1, k_4) := ⊥
transfer_msg(n_1, n_2, k_4, v, s_{41}) := true

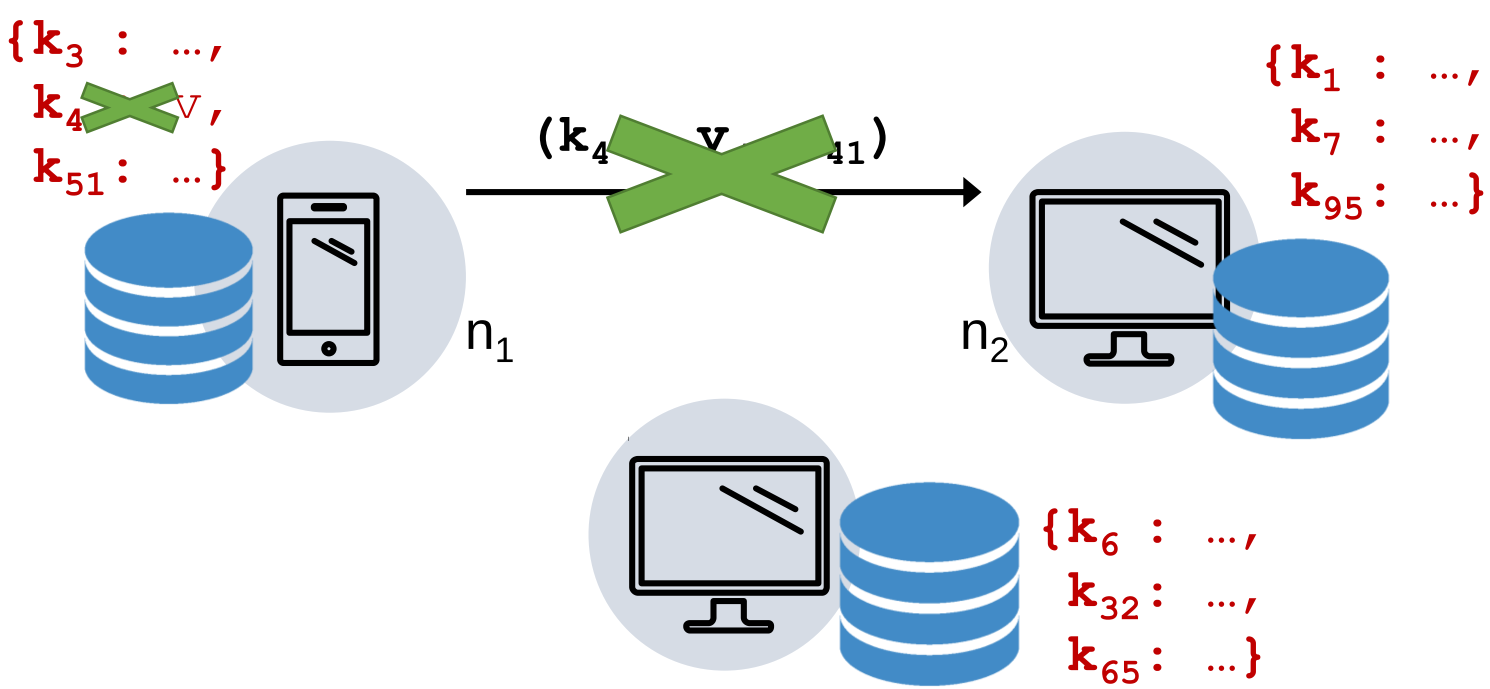


[SOSP15] IronFleet: Proving Practical Distributed Systems Correct. C. Hawblitzel, J. Howell, M. Kapritsos, J.R. Lorch, B. Parno, M.L. Roberts, S. Setty, and B. Zill

Example: Sharded Key-Value Store

drop transfer message:

transfer_msg(n_1, n_2, k_4, v, s_{41}) := false

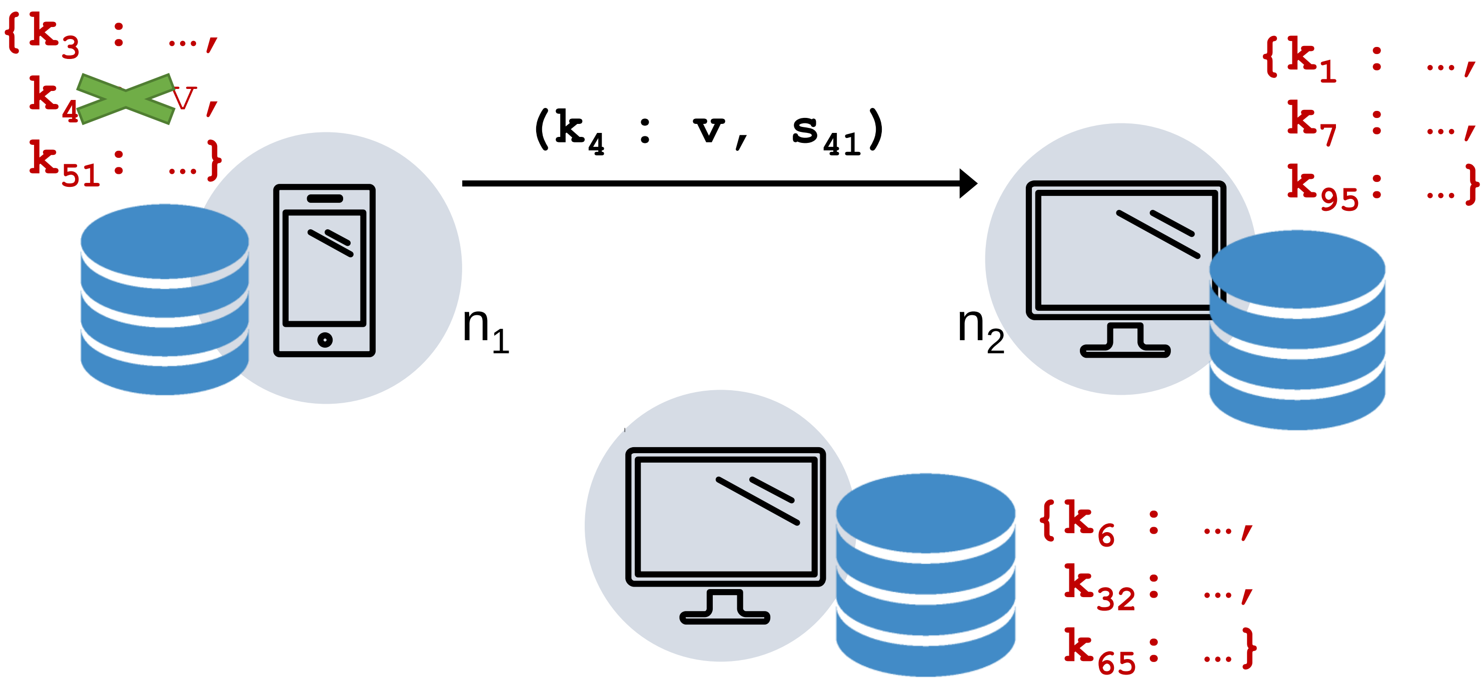


[SOSP15] IronFleet: Proving Practical Distributed Systems Correct. C. Hawblitzel, J. Howell, M. Kapritsos, J.R. Lorch, B. Parno, M.L. Roberts, S. Setty, and B. Zill

Example: Sharded Key-Value Store

retransmit:

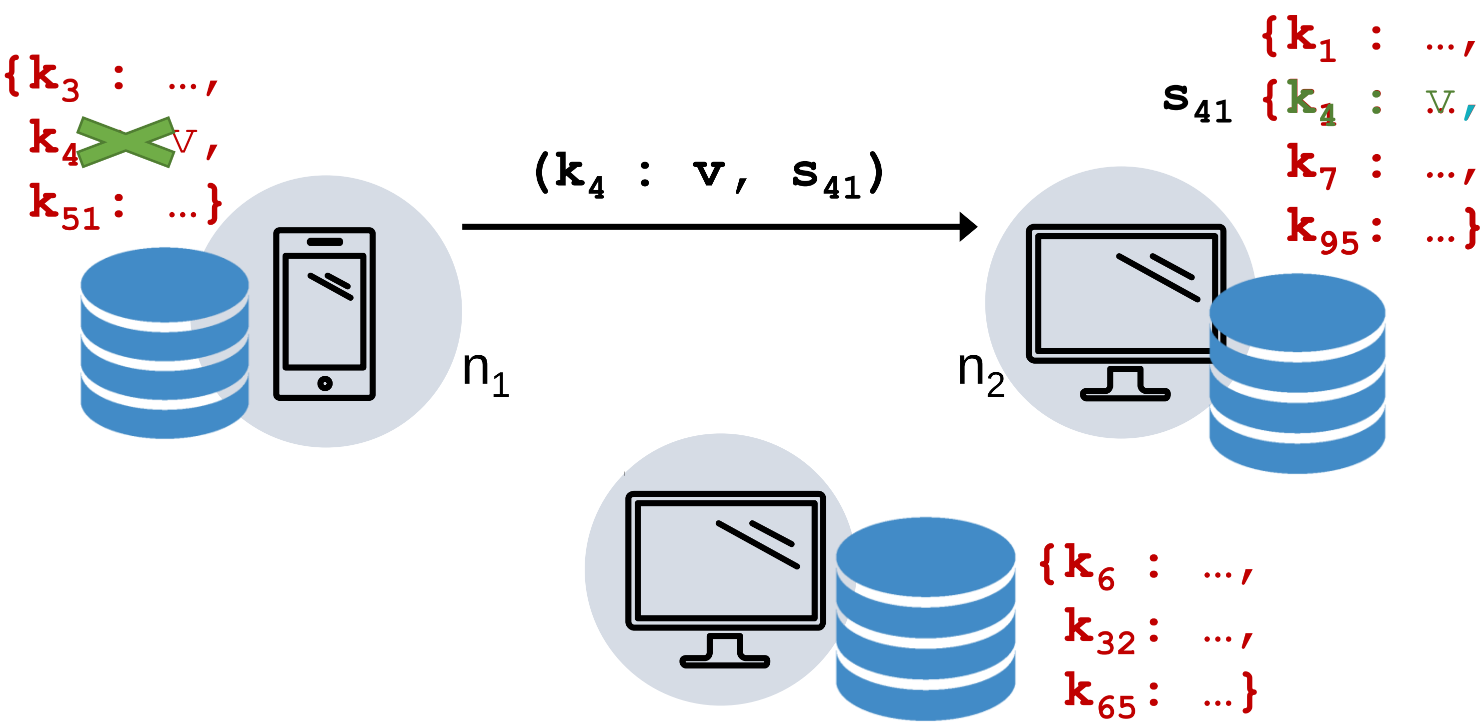
transfer_msg(n_1, n_2, k_4, v, s_{41}) := true



[SOSP15] IronFleet: Proving Practical Distributed Systems Correct. C. Hawblitzel, J. Howell, M. Kapritsos, J.R. Lorch, B. Parno, M.L. Roberts, S. Setty, and B. Zill

Example: Sharded Key-Value Store

recv transfer message:
table(n_2, k_4) := v; seq_recvd(n_2, n_1, s_{41}) := true
transfer_msg(n_1, n_2, k_4, v, s_{41}) := false

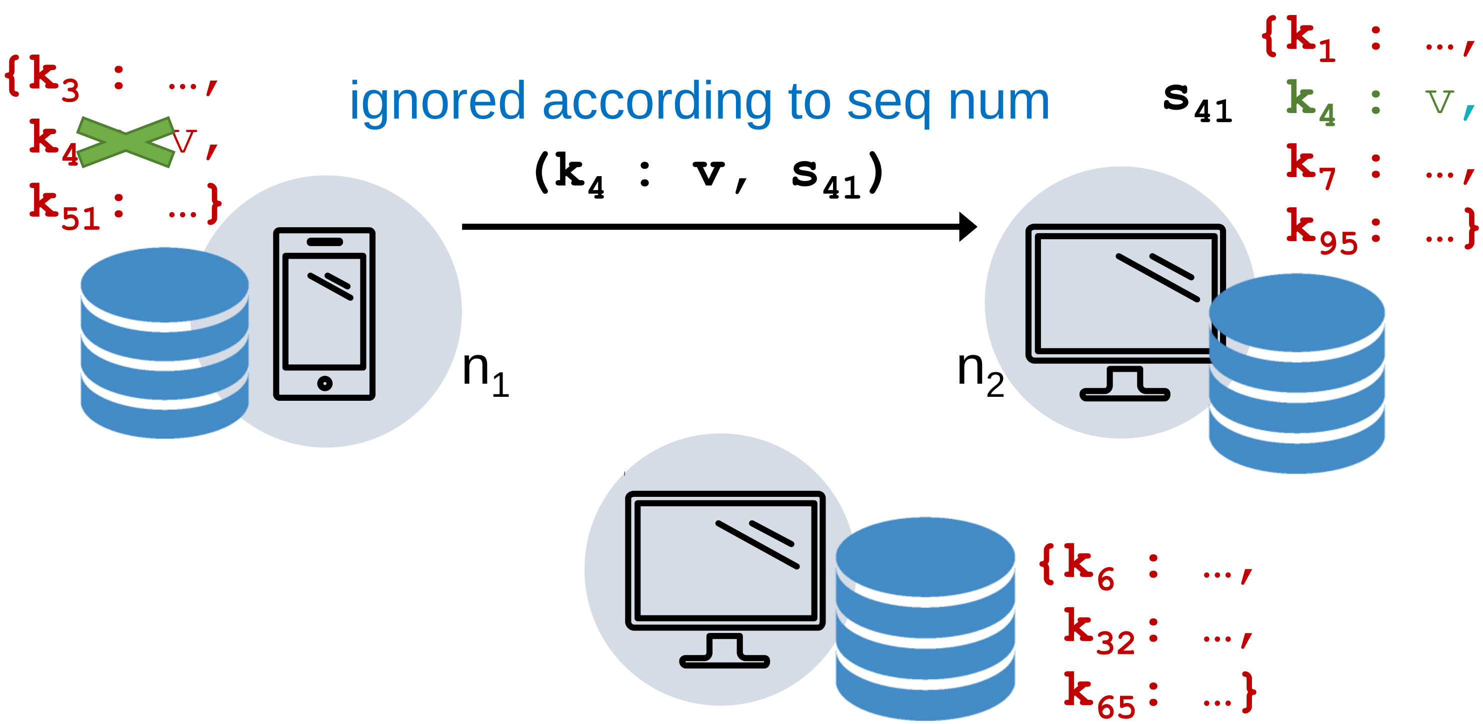


[SOSP15] IronFleet: Proving Practical Distributed Systems Correct. C. Hawblitzel, J. Howell, M. Kapritsos, J.R. Lorch, B. Parno, M.L. Roberts, S. Setty, and B. Zill

Example: Sharded Key-Value Store

retransmit:

transfer_msg(n_1, n_2, k_4, v, s_{41}) := true



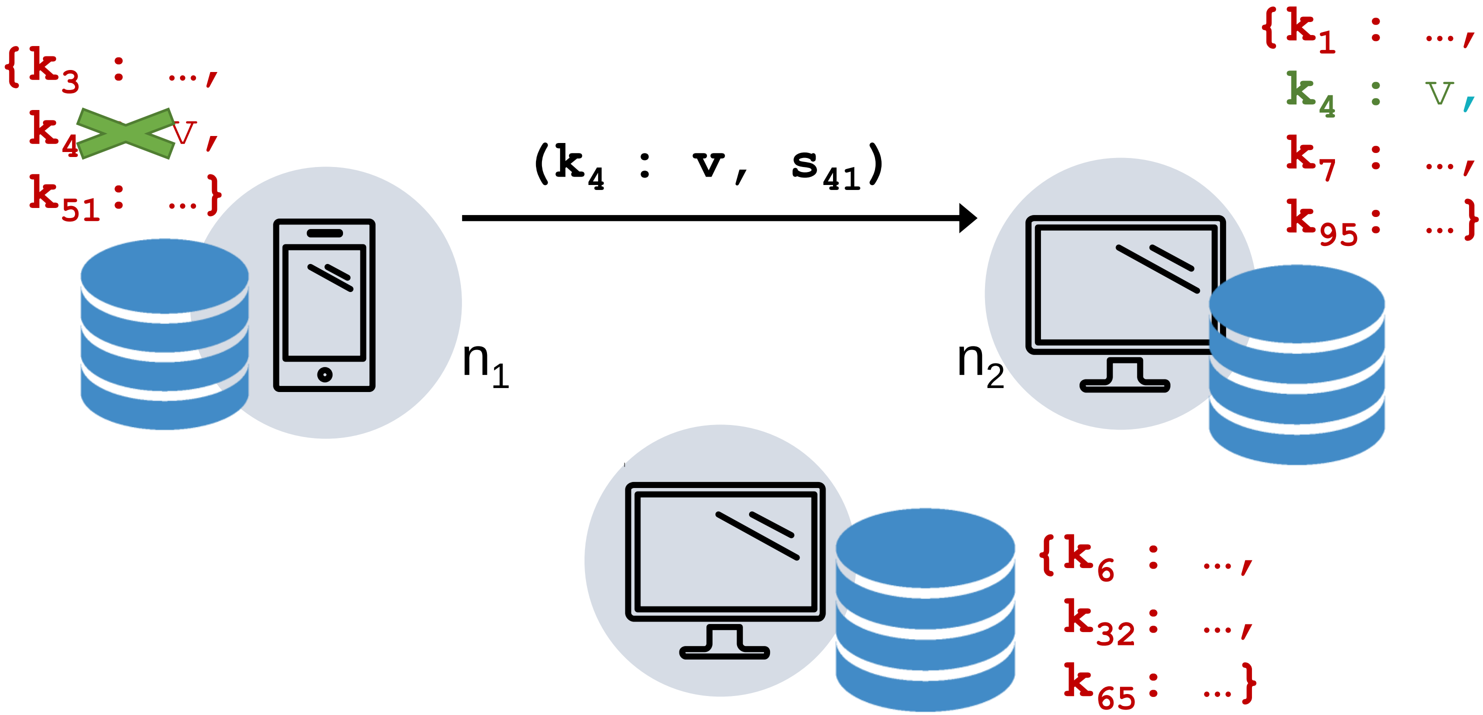
[SOSP15] IronFleet: Proving Practical Distributed Systems Correct. C. Hawblitzel, J. Howell, M. Kapritsos, J.R. Lorch, B. Parno, M.L. Roberts, S. Setty, and B. Zill

Example: Sharded Key-Value Store

Safety property:

$\forall n_1, n_2, k, v_1, v_2.$

$$\mathrm{table}(n_1, k, v_1) \wedge \mathrm{table}(n_2, k, v_2) \rightarrow v_1 = v_2$$



[SOSP15] IronFleet: Proving Practical Distributed Systems Correct. C. Hawblitzel, J. Howell, M. Kapritsos, J.R. Lorch, B. Parno, M.L. Roberts, S. Setty, and B. Zill

Deductive Verification for Sharded KV

invariant $\forall k, n_1, n_2, v_1, v_2.\ \texttt{table}(n_1, k, v_1) \land \texttt{table}(n_2, k, v_2) \rightarrow n_1 = n_2 \land v_1 = v_2$

invariant $\forall k, n_1, n_2.\ \texttt{owner}(n_1, k) \land \texttt{owner}(n_2, k) \rightarrow n_1 = n_2$
invariant $\forall k, n, v.\ \texttt{table}(n, k, v) \rightarrow \texttt{owner}(n, k)$

invariant $\forall k, \text{src}, \text{dst}, v, s, n.\ \neg(\texttt{transfer\_msg}(\text{src}, \text{dst}, k, v, s) \land \neg\texttt{seqnum\_recvd}(\text{dst}, \text{src}, s) \land \texttt{owner}(n, k))$
invariant $\forall k, \text{src}, \text{dst}, v, s, n.\ \neg(\texttt{unacked}(\text{src}, \text{dst}, k, v, s) \land \neg\texttt{seqnum\_recvd}(\text{dst}, \text{src}, s) \land \texttt{owner}(n, k))$

invariant $\forall k, \text{src}_1, \text{src}_2, \text{dst}_1, \text{dst}_2, v_1, v_2, s_1, s_2.\ \texttt{transfer\_msg}(\text{src}_1, \text{dst}_1, k, v_1, s_1) \land \neg\texttt{seqnum\_recvd}(\text{dst}_1, \text{src}_1, s_1)$
$\land\texttt{transfer\_msg}(\text{src}_2, \text{dst}_2, k, v_2, s_2) \land \neg\texttt{seqnum\_recvd}(\text{dst}_2, \text{src}_2, s_2) \rightarrow \text{src}_1 = \text{src}_2 \land \text{dst}_1 = \text{dst}_2 \land v_1 = v_2 \land s_1 = s_2$
invariant $\forall k, \text{src}_1, \text{src}_2, \text{dst}_1, \text{dst}_2, v_1, v_2, s_1, s_2. \texttt{transfer\_msg}(\text{src}_1, \text{dst}_1, k, v_1, s_1) \land \neg\texttt{seqnum\_recvd}(\text{dst}_1, \text{dst}_1, s_1)$
$\land\texttt{unacked}(\text{src}_2, \text{dst}_2, k, v_2, s_2) \land \neg\texttt{seqnum\_recvd}(\text{dst}_2, \text{src}_2, s_2) \rightarrow \text{src}_1 = \text{src}_2 \land \text{dst}_1 \land \text{dst}_2 \land v_1 = v_2 \land s_1 = s_2$
invariant $\forall \text{src}_1, \text{src}_2, \text{dst}_1, \text{dst}_2, v_1, v_2, s_1, s_2.\ \texttt{unacked}(\text{src}_1, \text{dst}_1, k, v_1, s_1) \land \neg\texttt{seqnum\_recvd}(\text{dst}_1, \text{src}_1, s_1)$
$\land\texttt{unacked}(\text{src}_2, \text{dst}_2, k, v_2, s_2) \land \neg\texttt{seqnum\_recvd}(\text{dst}_2, \text{src}_2, s_2) \rightarrow \text{src}_1 = \text{src}_2 \land \text{dst}_1 = \text{dst}_2 \land v_1 = v_2 \land s_1 = s_2$

Labor intensive

Invariant Inference for Sharded KV

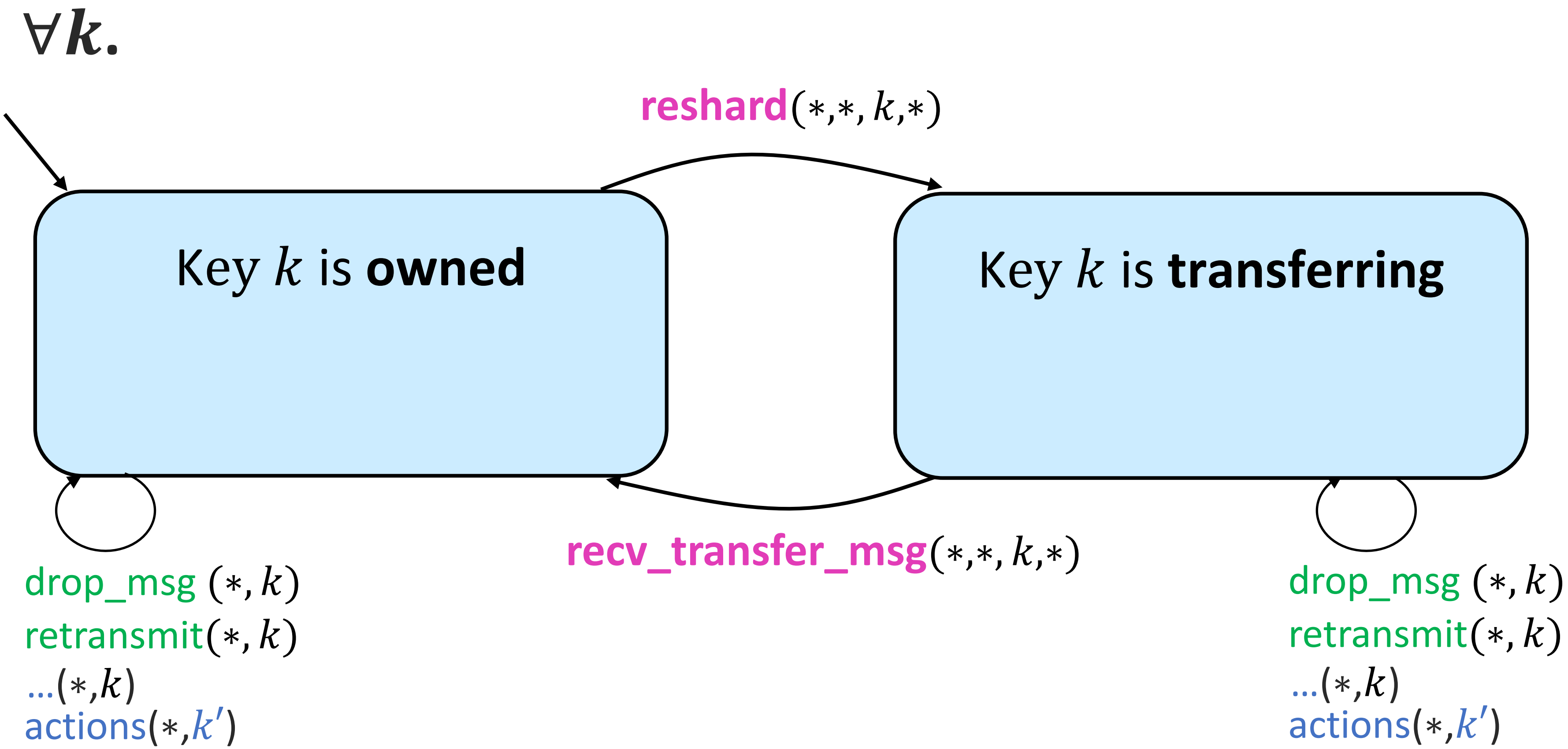
Protocol	Inductive Invariant Inference
…	…
Sharded KV	**failed to converge** in 1 hour in **13/16** Z3 seeds

Solution: guide using phase structure

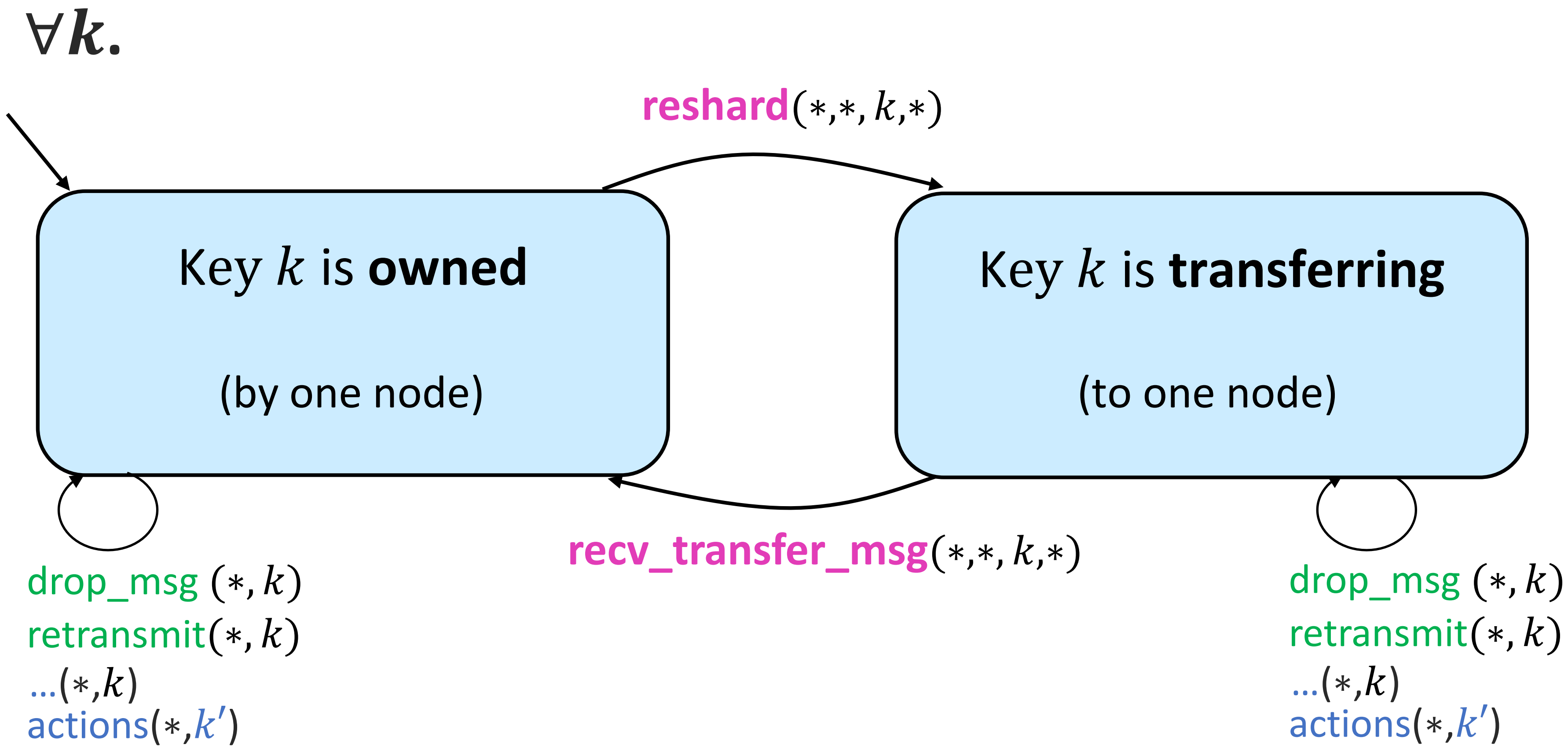
Limited and fragile

[CAV'15, JACM'17] Property-Directed Inference of Universal Invariants or Proving Their Absence, A. Karbyshev, N. Bjorner, S. Itzhaky, N. Rinetzky and S. Shoham.

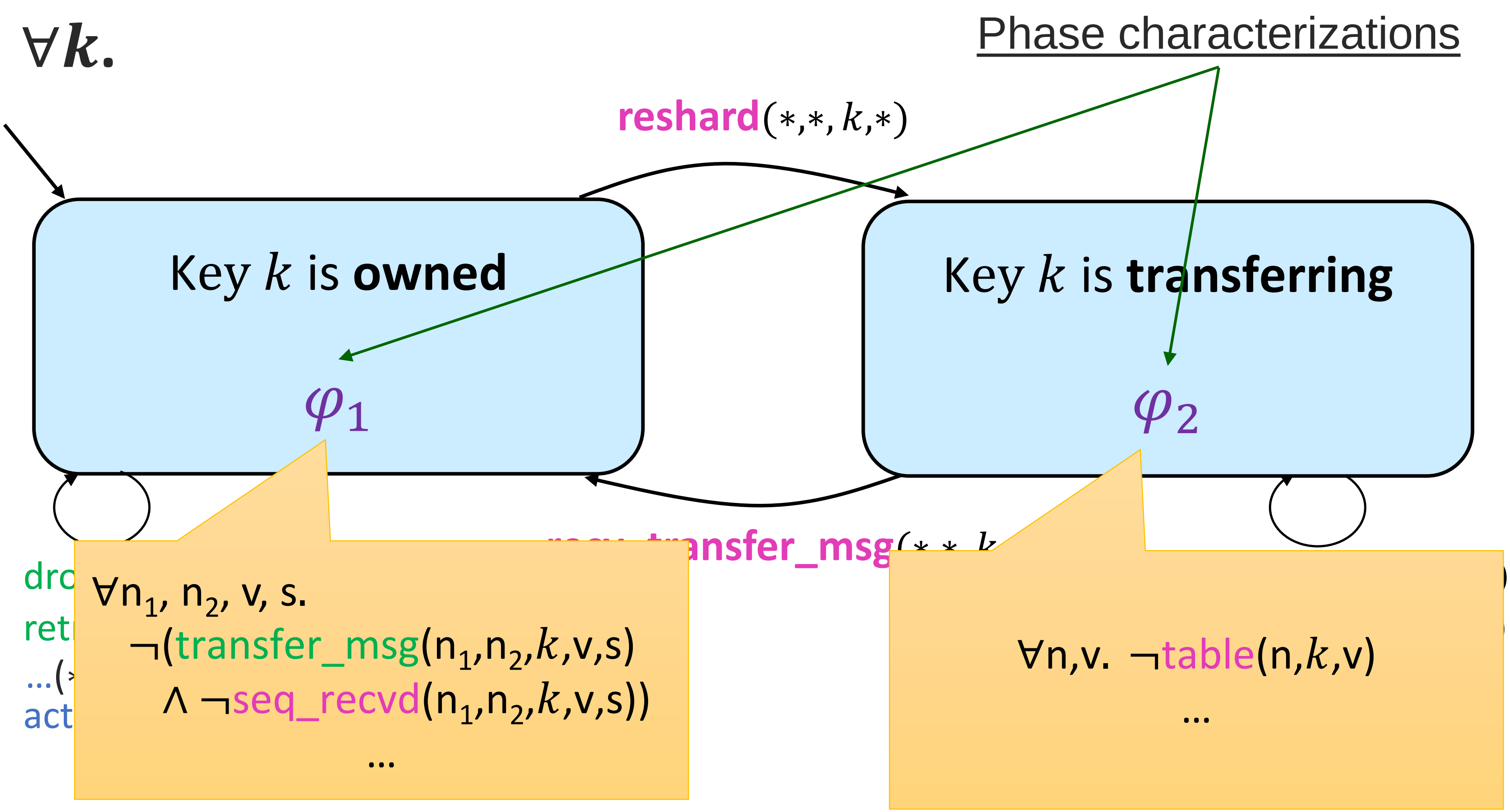
Phase Structure of Distributed KV's Proof



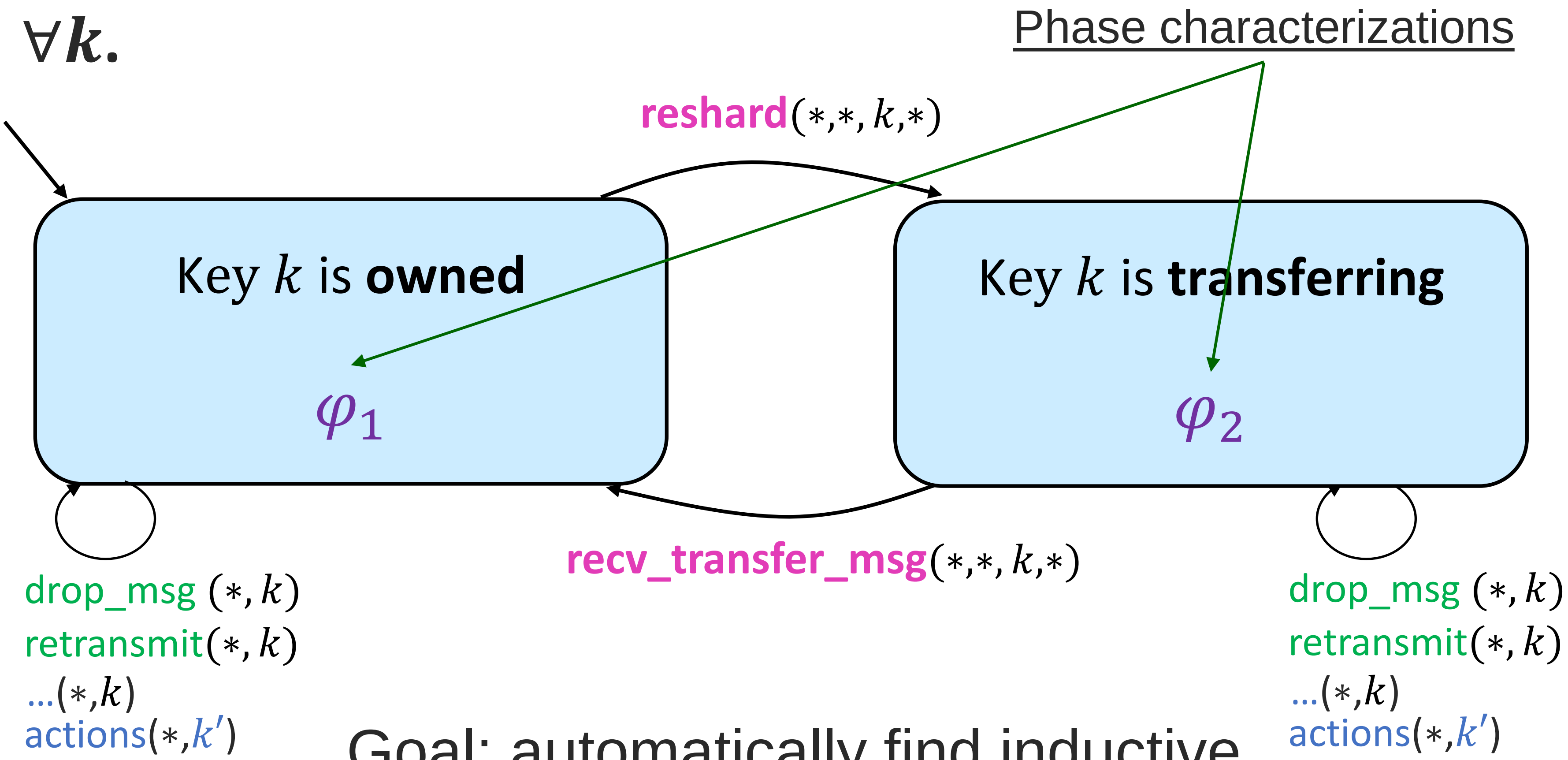
Phase Structure of Distributed KV's Proof



Phase Structure of Distributed KV's Proof

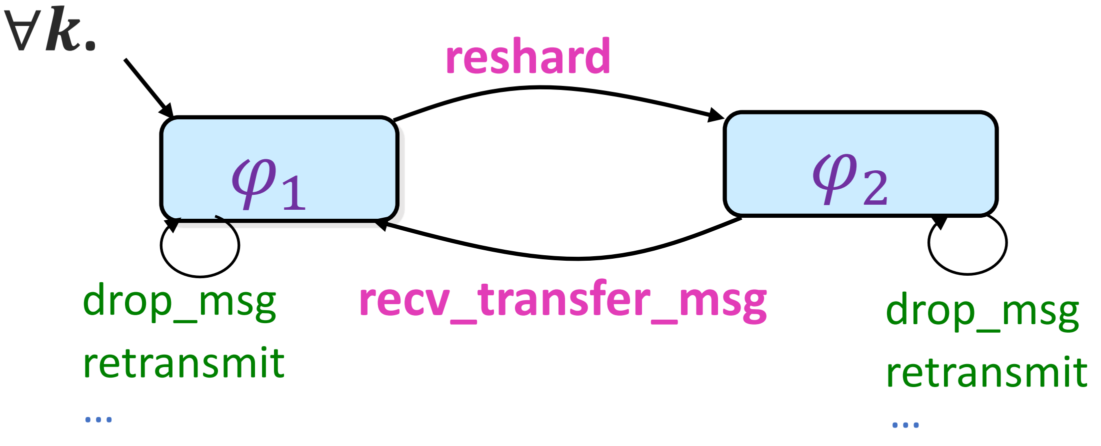


Phase Structure of Distributed KV's Proof



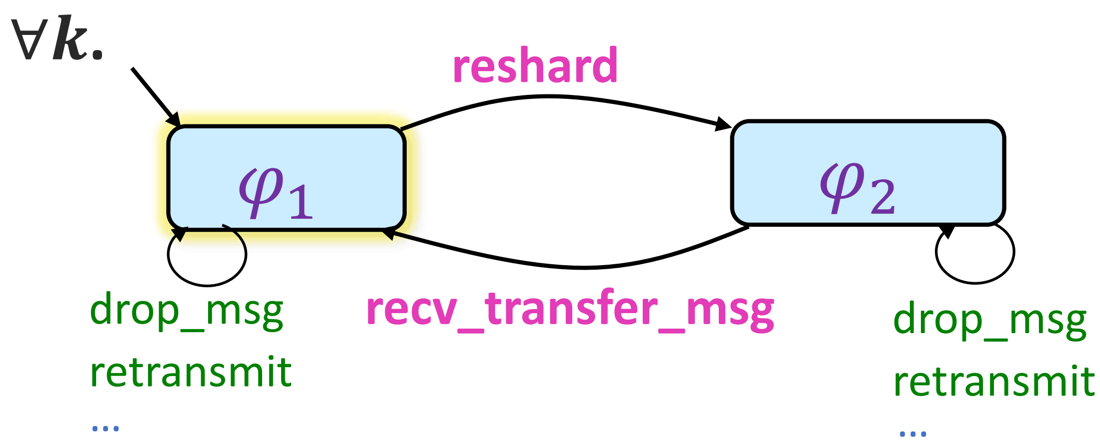
Goal: automatically find inductive phase characterizations

Inductive Phase Invariants



Inductive Phase Invariants

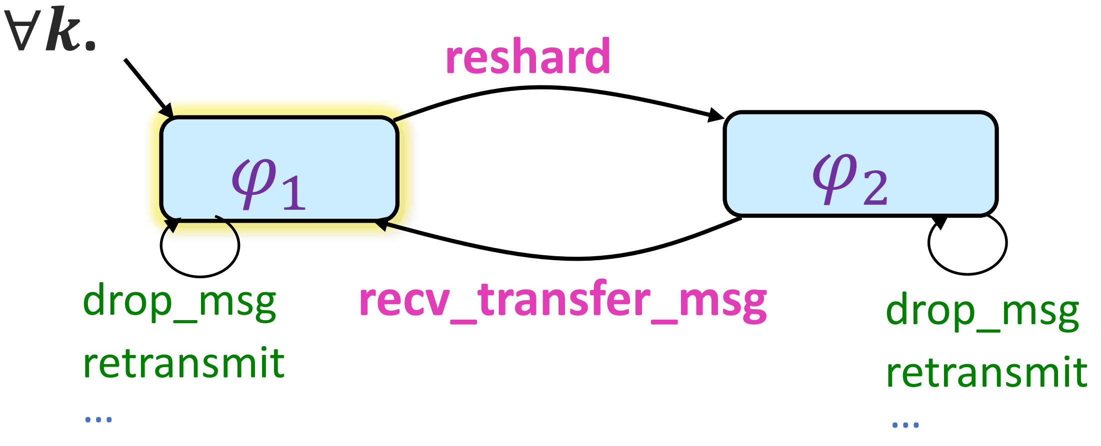
$$\text{Init} \Longrightarrow \varphi_1$$



Inductive Phase Invariants

$\text{Init} \Longrightarrow \varphi_1$

$\varphi_1 \Longrightarrow \text{Safety}, \ldots$

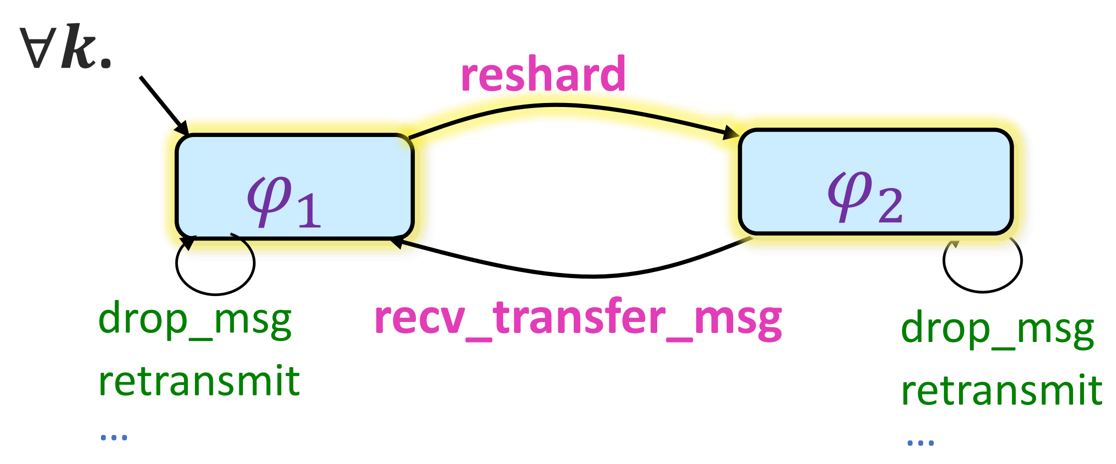


Inductive Phase Invariants

$\text{Init} \Longrightarrow \varphi_1$

$\varphi_1 \Longrightarrow \text{Safety}, \ldots$

$\varphi_1 \wedge \text{TR}_{\text{reshard}} \Longrightarrow \varphi_2{}', \ldots$



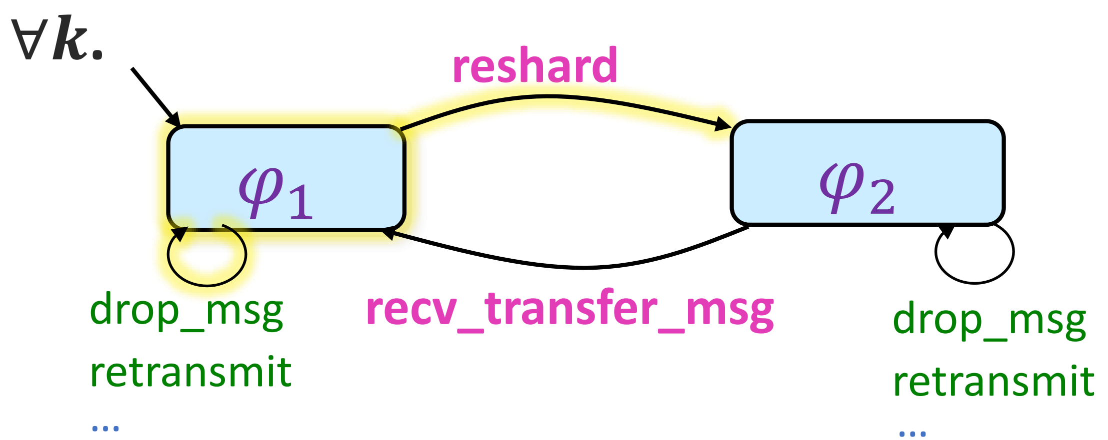
Inductive Phase Invariants

$\text{Init} \Longrightarrow \varphi_1$

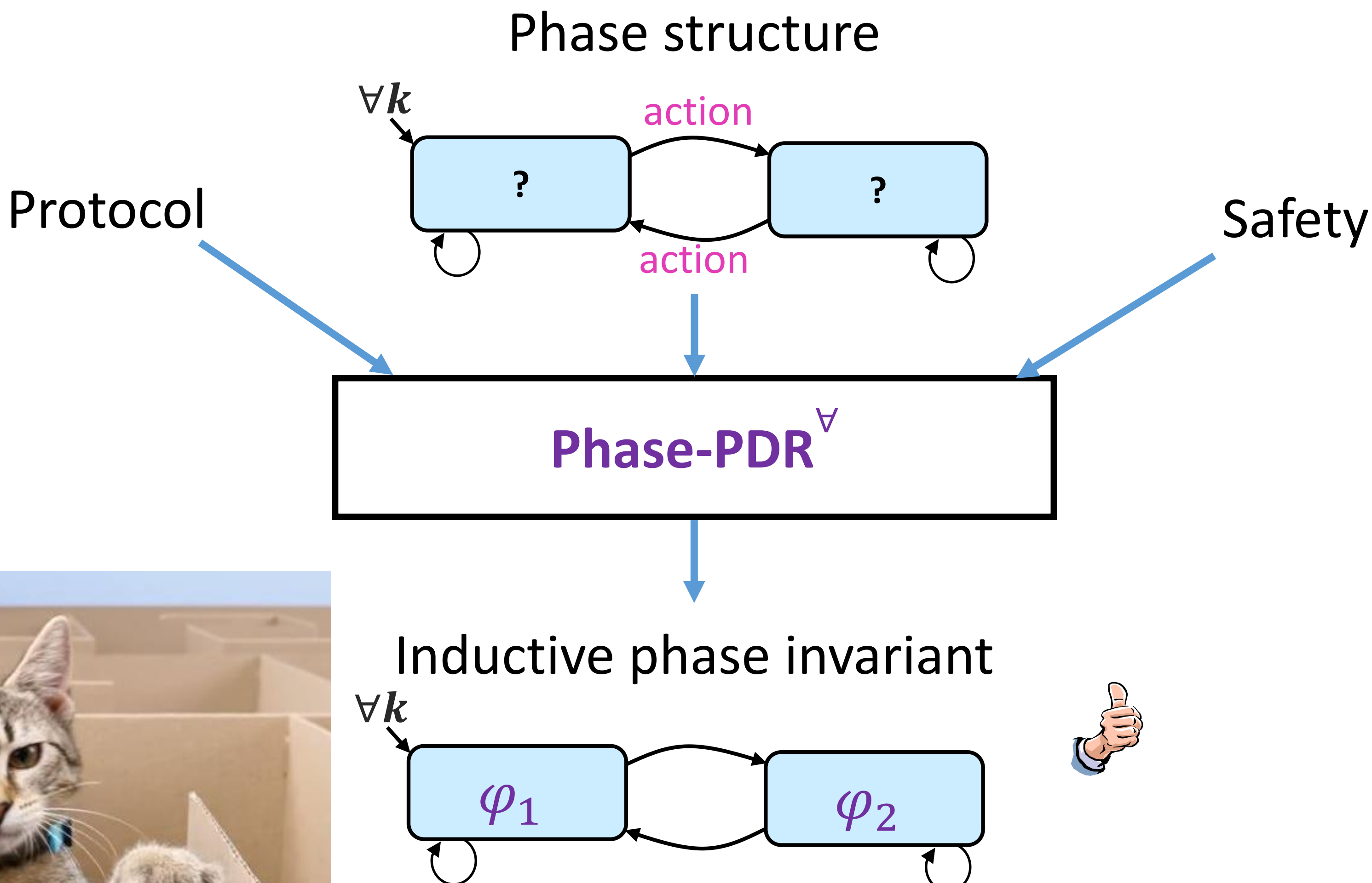
$\varphi_1 \Longrightarrow \text{Safety}, \ldots$

$\varphi_1 \wedge \text{TR}_{\text{reshard}} \Longrightarrow \varphi_2{}', \ldots$

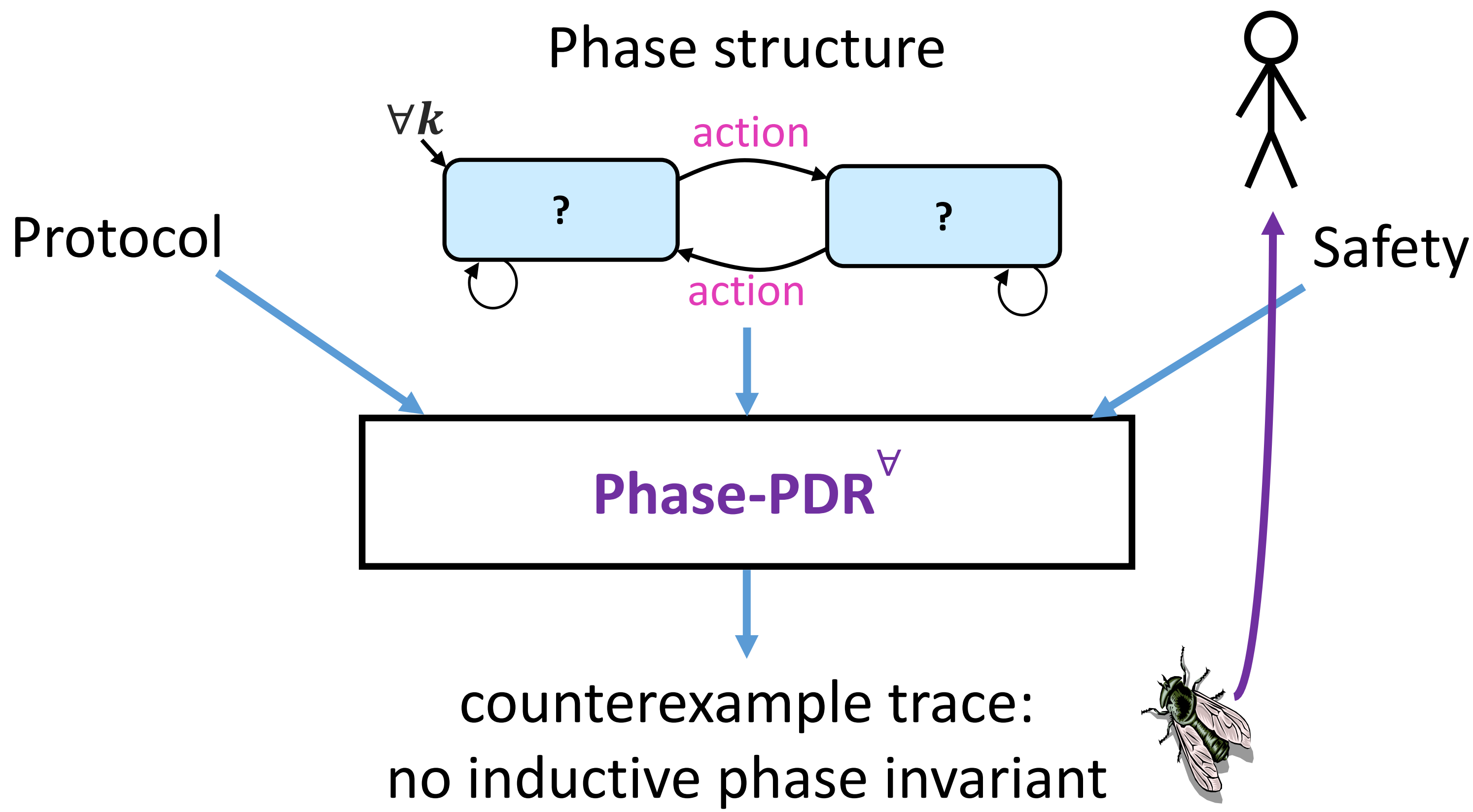
$\varphi_1 \wedge \text{TR} \Longrightarrow \text{TR}_{\text{reshard}} \vee \text{TR}_{\text{drop\_msg}} \vee \ldots, \ldots$



Inference Using Phase Structures



Inference Using Phase Structures



Inferring Inductive Phase Invariants

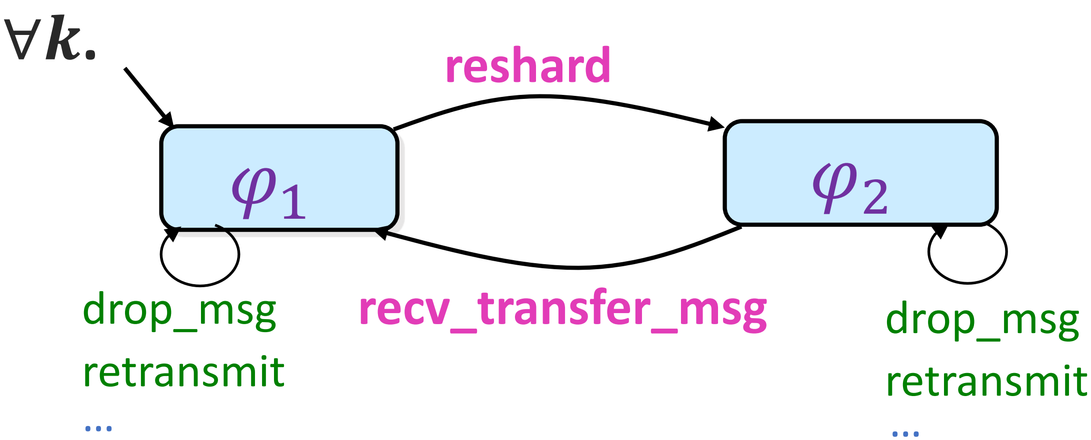
$\text{Init} \Longrightarrow \varphi_1$

$\varphi_1 \Longrightarrow \text{Safety}, \ldots$

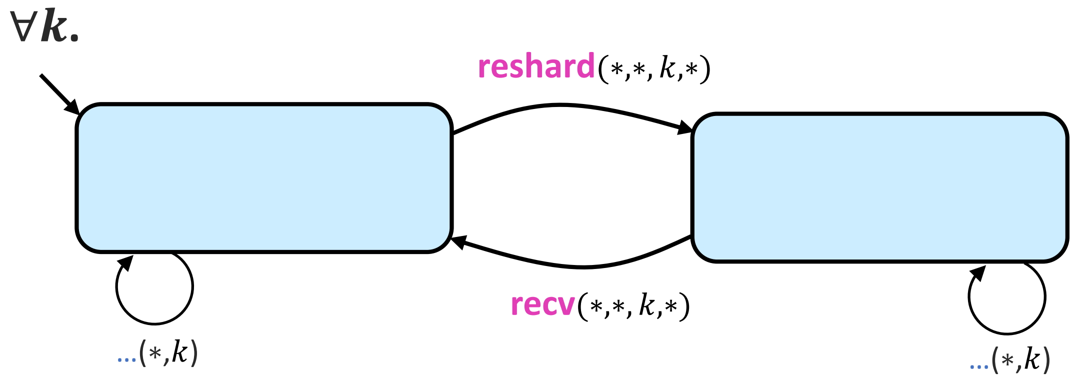
$\varphi_1 \wedge \text{TR}_{\text{reshard}} \Longrightarrow \varphi_2{}', \ldots$

$\varphi_1 \wedge \text{TR} \Longrightarrow \text{TR}_{\text{reshard}} \vee \text{TR}_{\text{drop\_msg}} \vee \ldots, \ldots$

System of *linear* **Constrained Horn Clauses** (CHC) over unknown predicates φ_1, φ_2!

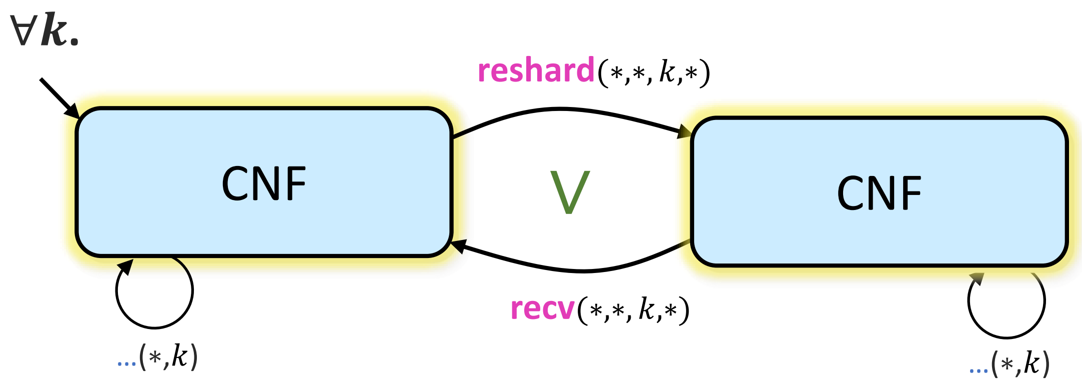


Phases Guide Inference



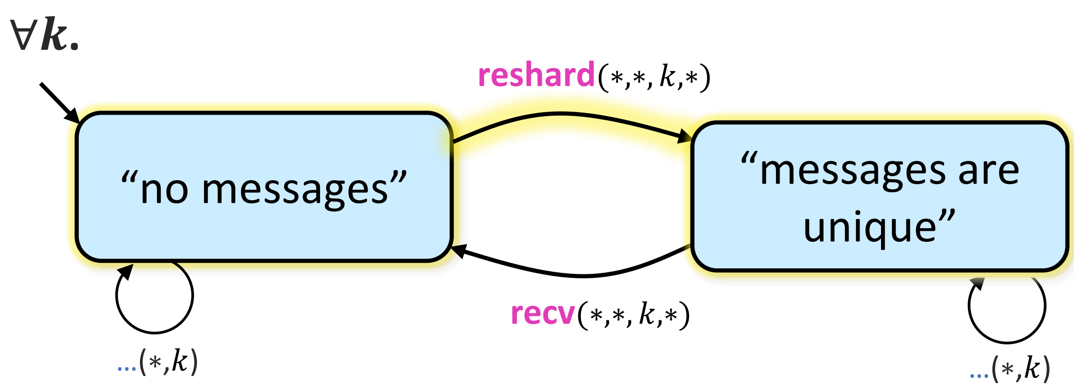
Phases Guide Inference

- **Semantic disjunctive template**



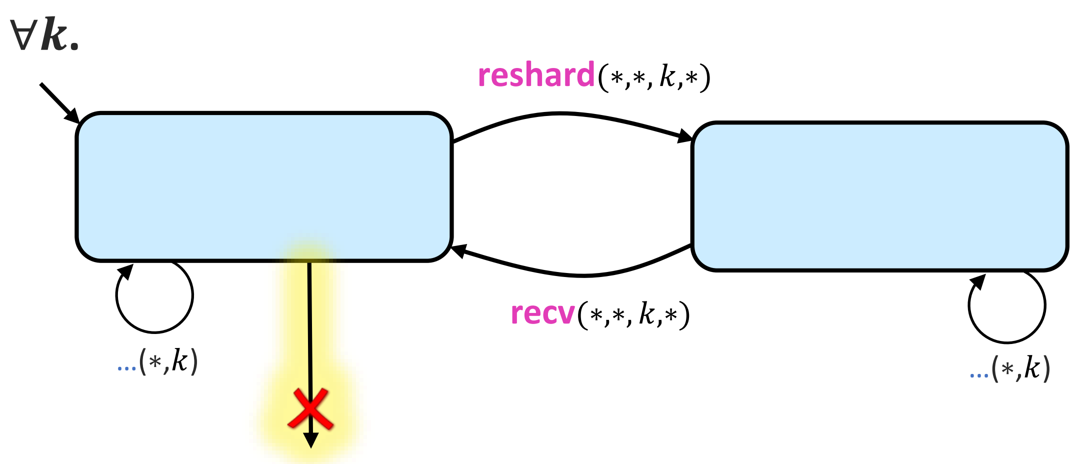
Phases Guide Inference

- Semantic disjunctive template
- **Phase decomposition and incremental construction**



Phases Guide Inference

- Semantic disjunctive template
- Phase decomposition and incremental construction
- **Impossible transitions**



Implementation

wilcoxjay/mypyvy

- **mypyvy**: a tool inspired by Ivy, over Z3
  - Statically-typed Python

Invariant inference:

- Standard adaptation
  - **PDR**$^{\forall}$ for standard inductive invariants
  - **Phase-PDR**$^{\forall}$ for phase invariants

[CAV'15, JACM'17] Property-Directed Inference of Universal Invariants or Proving Their Absence, A. Karbyshev, N. Bjorner, S. Itzhaky, N. Rinetzky and S. Shoham.

Evaluation

Protocol	Inductive Invariant [seconds]	Phase Structure [seconds]
Lock server (single lock)	2.21	**0.67**
Lock server (multiple locks)	2.73	**1.06**
Simple consensus	**60.54**	1355*
Ring leader election	152.44	**2.53**
Sharded KV (basic)	1.79	**1.59**
Sharded KV	2070*	**372.5**
MESI cache coherence	-	**90.1**

* not all runs terminated in 1 hour

\- no runs terminated in 1 hour

Evaluation

Protocol	Inductive Invariant [seconds]	Phase Structure [seconds]
Lock server (single lock)	2.21	**0.67**
Lock server (multiple locks)	2.73	**1.06**
Simple consensus	**60.54**	1355*
Ring leader election	152.44	**2.53**
Sharded KV (basic)	1.79	**1.59**
Sharded KV	2070*	**372.5**
MESI cache coherence	-	**90.1**

* not all runs terminated in 1 hour

\- no runs terminated in 1 hour

Evaluation

Protocol	Inductive Invariant [seconds]	Phase Structure [seconds]
Lock server (single lock)	2.21	**0.67**
Lock server (multiple locks)	2.73	**1.06**
Simple consensus	**60.54**	1355*
Ring leader election	152.44	**2.53**
Sharded KV (basic)	1.79	**1.59**
Sharded KV	2070*	**372.5**
MESI cache coherence	-	**90.1**

* not all runs terminated in 1 hour

\- no runs terminated in 1 hour

Evaluation

Protocol	**Inductive Invariant [seconds]**	**Phase Structure [seconds]**
Lock server (single lock)	2.21	**0.67**
Lock server (multiple locks)	2.73	**1.06**
Simple consensus	**60.54**	1355*
Ring leader election	152.44	**2.53**
Sharded KV (basic)	1.79	**1.59**
Sharded KV	2070*	**372.5**
MESI cache coherence	-	**90.1**

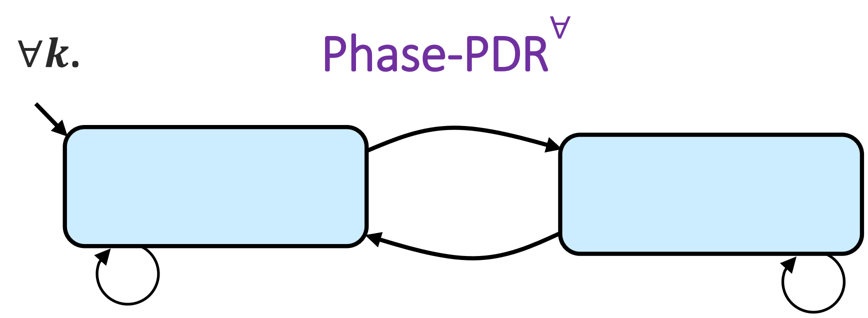
* not all runs terminated in 1 hour

\- no runs terminated in 1 hour

Summary

User-guided invariant inference by **phase structures**

- Convey high-level intuition
- Direct proof search effectively
  - Semantic disjunctive template
  - Incrementality between phases
  - Disabled transitions
- Facilitate inference beyond the state of the art
- Faster convergence



- **Sketching correctness** of infinite-state systems