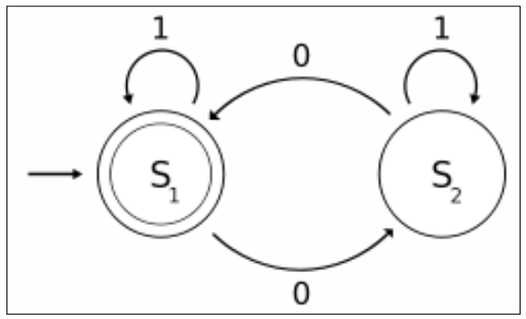
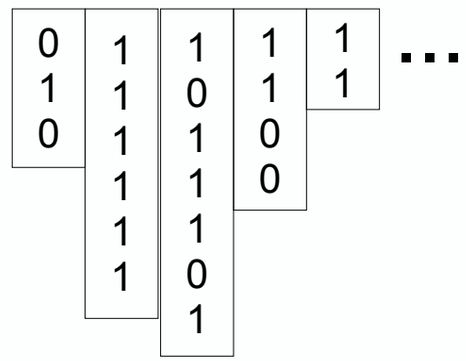
Leveraging Existing Instrumentation to Automatically Infer Invariant-Constrained Models

University of Washington

Ivan Beschastnikh
Yuriy Brun
Sigurd Schneider
Michael Sloan
Michael D. Ernst

Saarland University

Synoptic:

Mining  * from 

University of Washington

Ivan Beschastnikh
Yuriy Brun
Sigurd Schneider
Michael Sloan
Michael D. Ernst

Saarland University

* http://en.wikipedia.org/wiki/File:DFAexample.svg

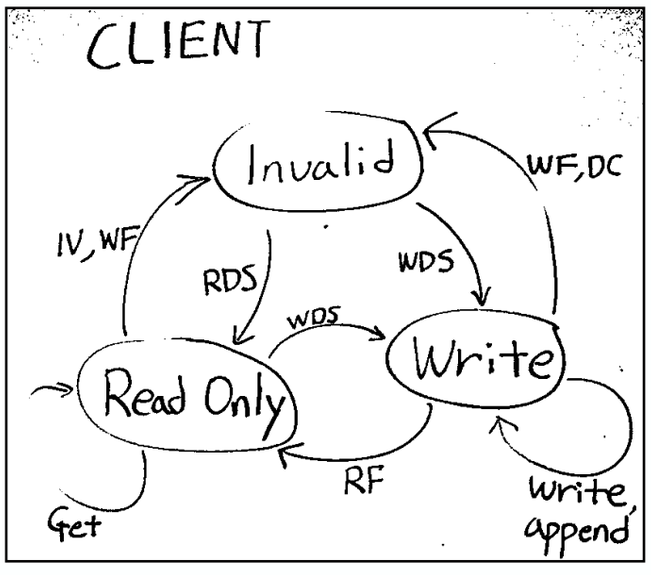
Motivating question

I am a developer.

Why does my system behave in a certain manner?

One answer is to...

Study the relevant artifacts




```
while (transIter.hasNext()) {
    // Create new tc map for node m.
    if (!tc.containsKey(m)) {
        tc.put(m, new LinkedHashSet<EventNode>());
    }

    EventNode child = transIter.next().getTarget();

    if (!tcParents.containsKey(child)) {
        tcParents.put(child, new HashSet<EventNode>()
    }

    // Link m to c
    tc.get(m).add(child);
    tcParents.get(child).add(m);

    // Link m to all nodes that c is linked to in tc
    if (tc.containsKey(child)) {
        // m can reach nodes the child can reach tran
        tc.get(m).addAll(tc.get(child));
        // nodes that child can reach have m as a tc
        for (EventNode n : tc.get(child)) {
            if (!tcParents.containsKey(n)) {
                tcParents.put(n, new HashSet<EventNod
            }
            tcParents.get(n).add(m);
        }
```

```
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
```

```
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
```

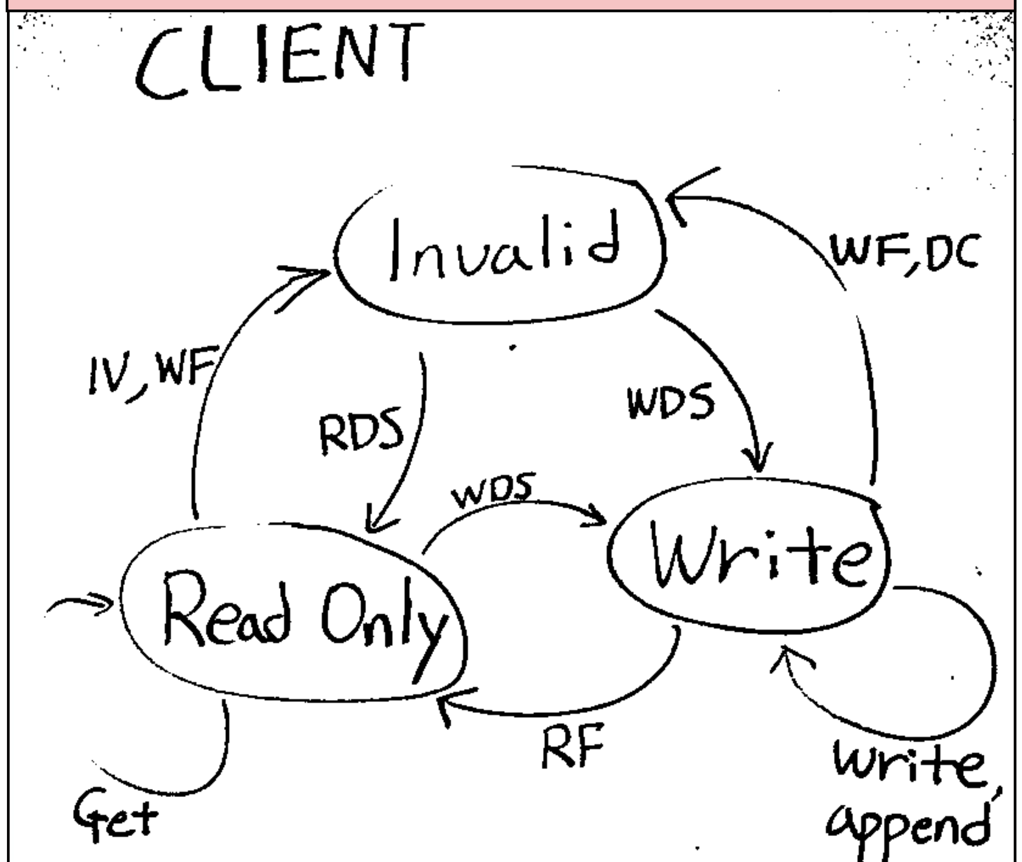
■ ■ ■

Specification | Code | Log

Models

Manually defined

Hard to Create



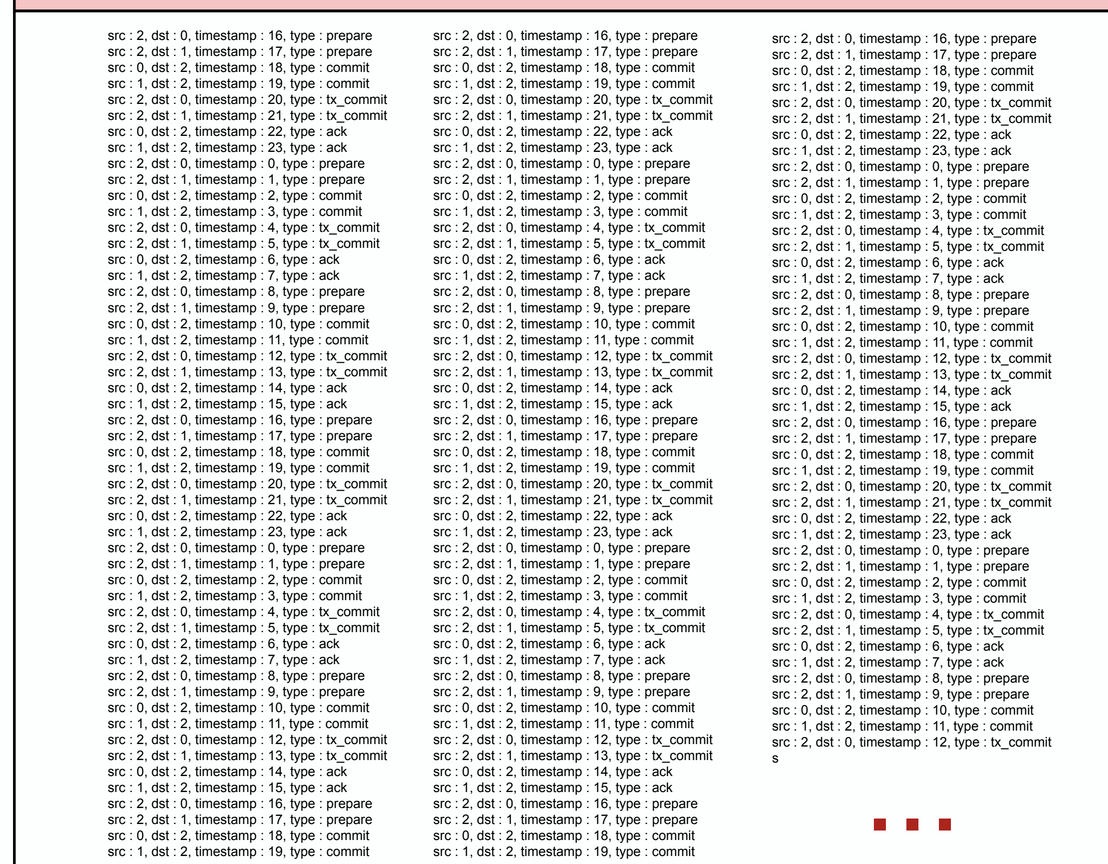
Easy to Use

Concise and exact

Logs

Trivial instrumentation

Easy to Create



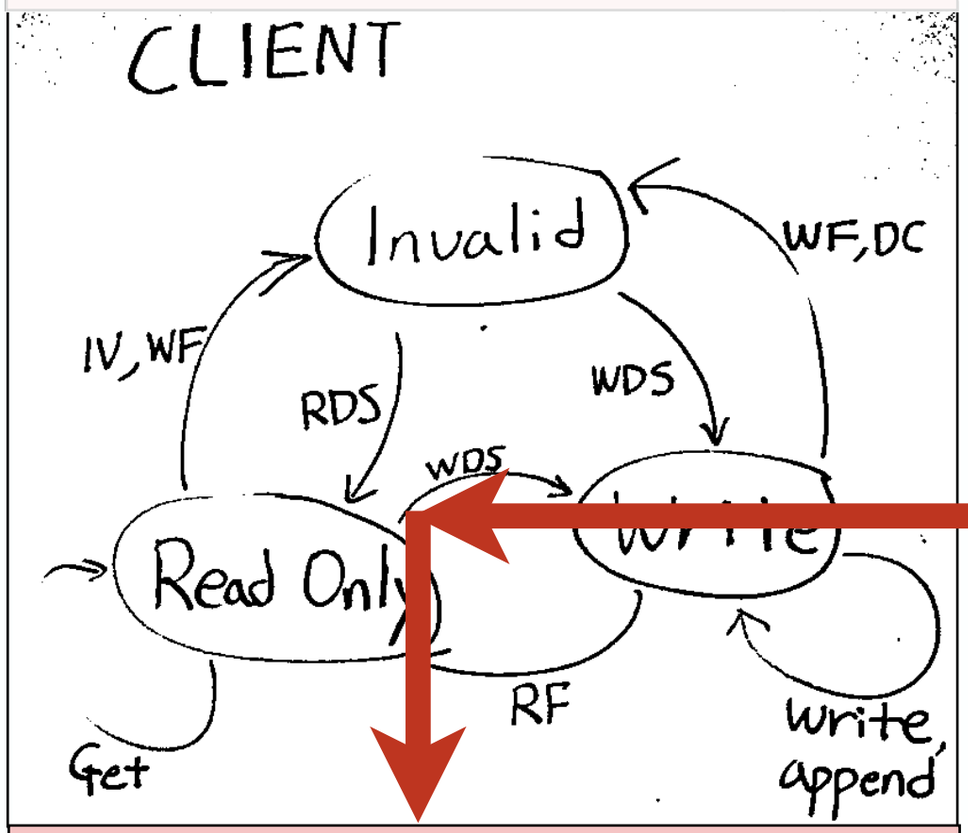
Hard to Use

A low-level view

Models

Manually defined

Hard to Create



Easy to Use

Concise and exact

Logs

Trivial instrumentation

Easy to Create

```
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
```

...

Hard to Use

A low-level view

Bridging logging and modeling

```
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 1, dst : 2, timestamp : 19, type : commit
...
```

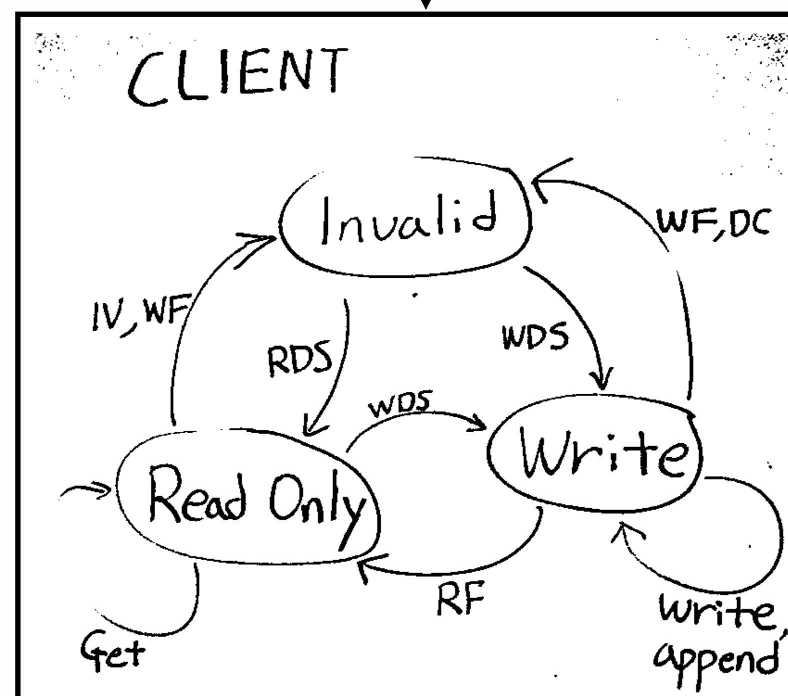
Input

Tool

- Specification mining
- Process discovery
- Model inference

Prior work:

Cook et al. TSE 1998
Dallmeier et al. WODA 2006
Lo et al. ASE 2010



Output

How Synoptic works

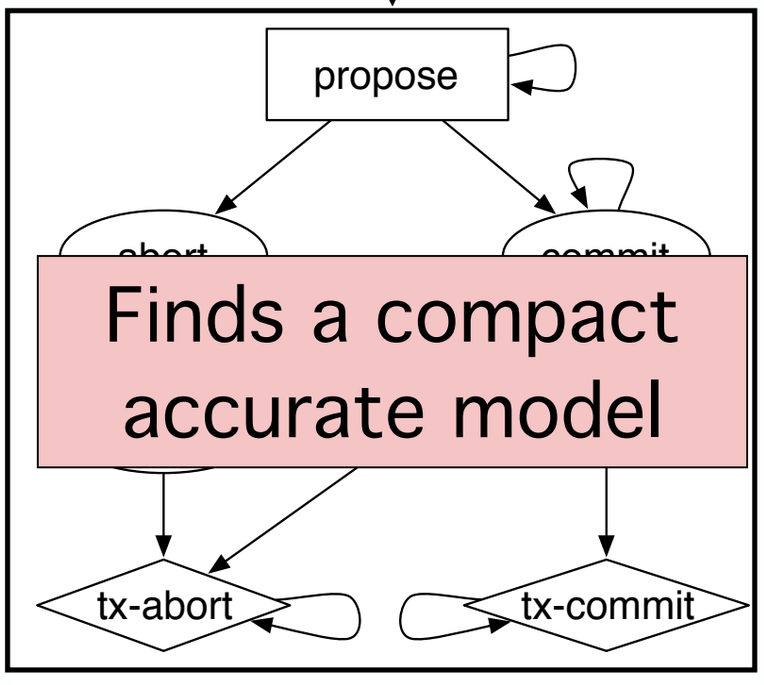
Does not require access to code

Input

Invariants

Mines and preserves temporal log invariants

Uses refinement instead of coarsening



Finds a compact accurate model

Output

10

Use cases for Synoptic models

- Log summarization
- Mental model validation
- Test case generation
- Comparison to formal specifications
- Evaluate test suite against actual usage
- Verifying a code fix

Talk outline

- Motivation
- Synoptic's design
- Evaluation results
- Future work
- Conclusion

Synoptic inputs

1. Log file
2. Regular expressions to parse the log file

13

Two phase commit protocol log

1. Log file

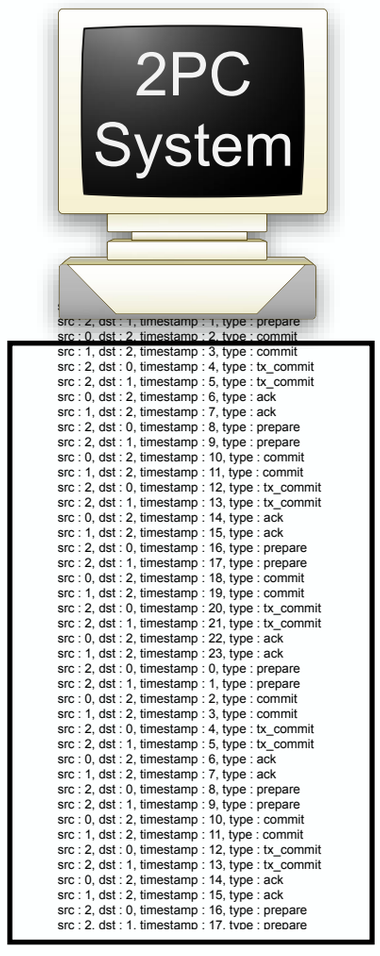
   Two phase commit protocol log

   I. Manager proposes a transaction (TX), each replica replies with an abort or a commit.

   II. Manager sends

   i) TX commit if all replicas commit

   ii) TX abort otherwise



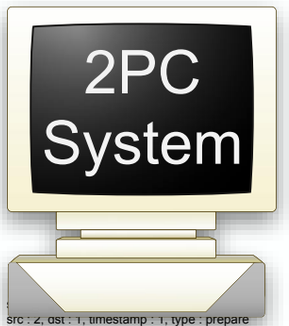
log.txt

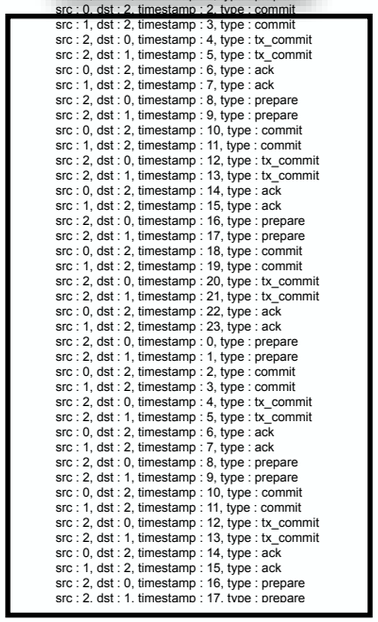
Manager maintains a totally ordered log of events for all transactions in the system

14

Regular expressions

1. Log file (e.g., two phase commit log)
2. Regular expressions to parse the log file
   - Split log into executions
   - From each log line extract:
     - Total event relation (e.g., time)
     - Event type





log.txt

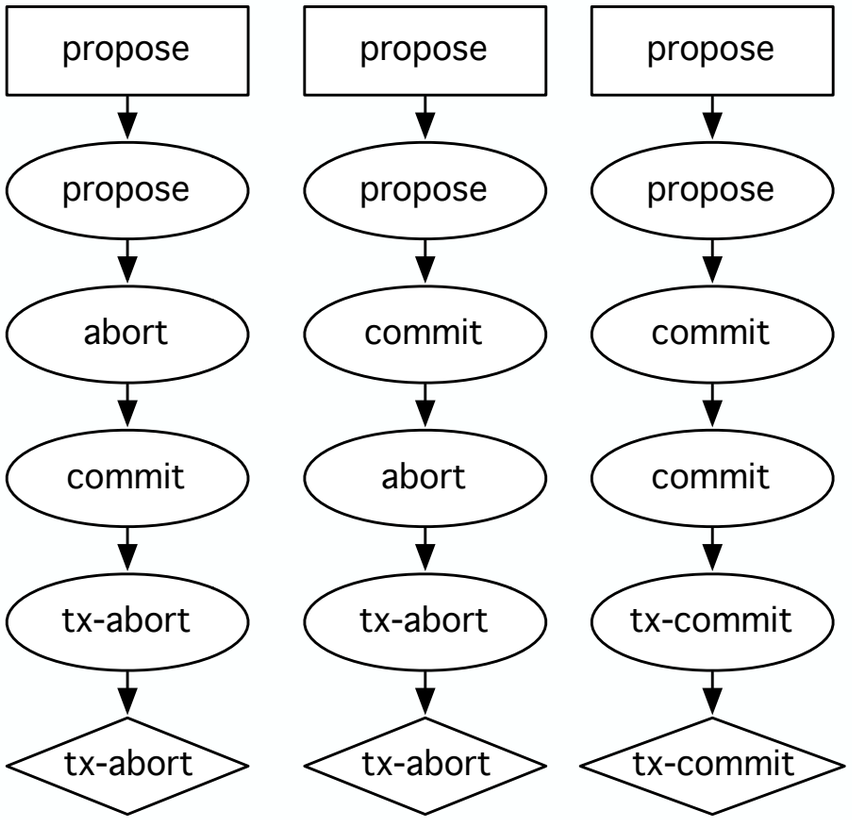
15

Synoptic overview

1/5. Parse log into a trace graph

2/5. Construct the initial model

3/5. Mine temporal invariants



X : initial event

Y : intermediate event

Z : terminal event

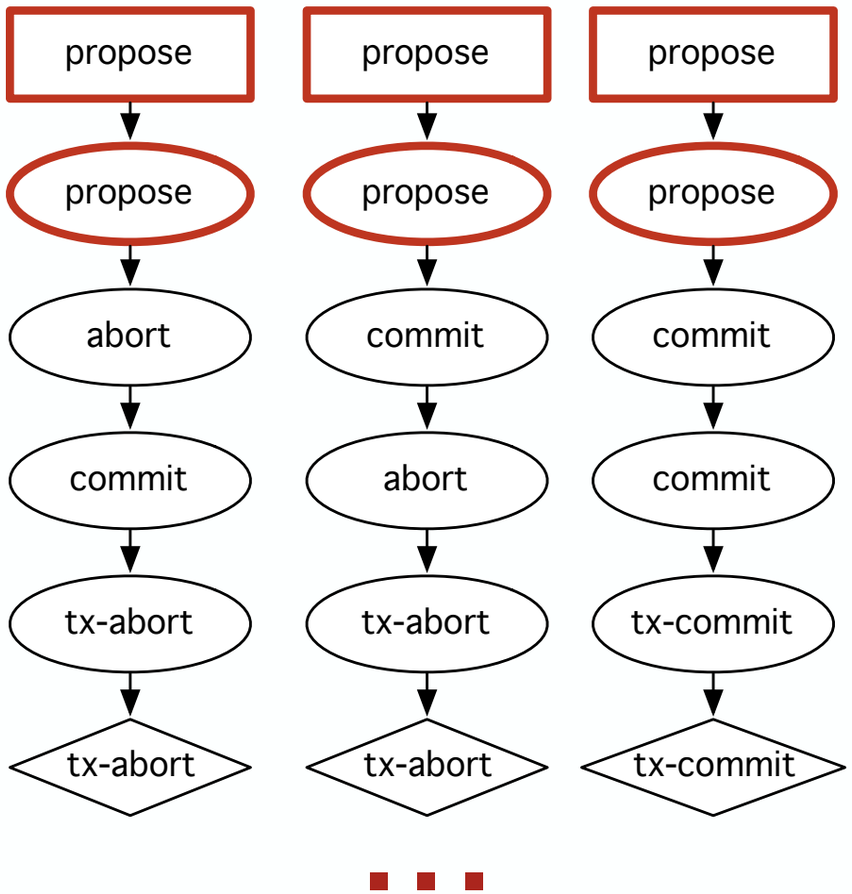
16

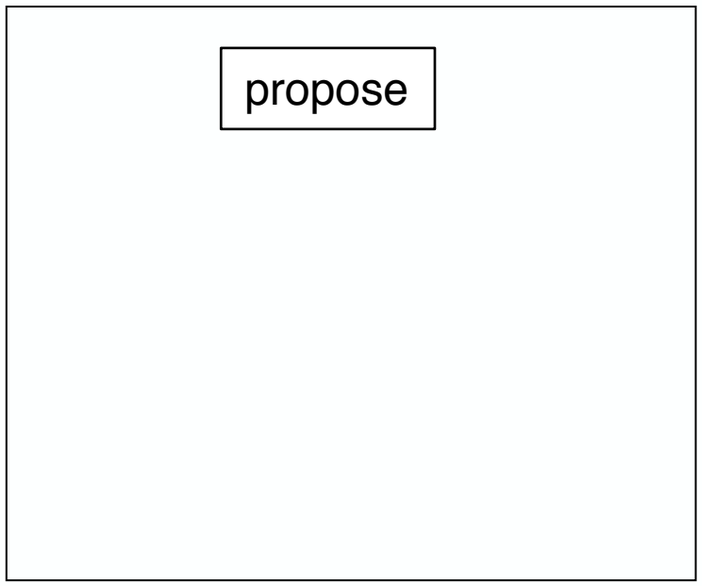
Synoptic overview

1/5. Parse log into a trace graph

2/5. Construct the initial model

3/5. Mine temporal invariants





A compact model with one node per event type

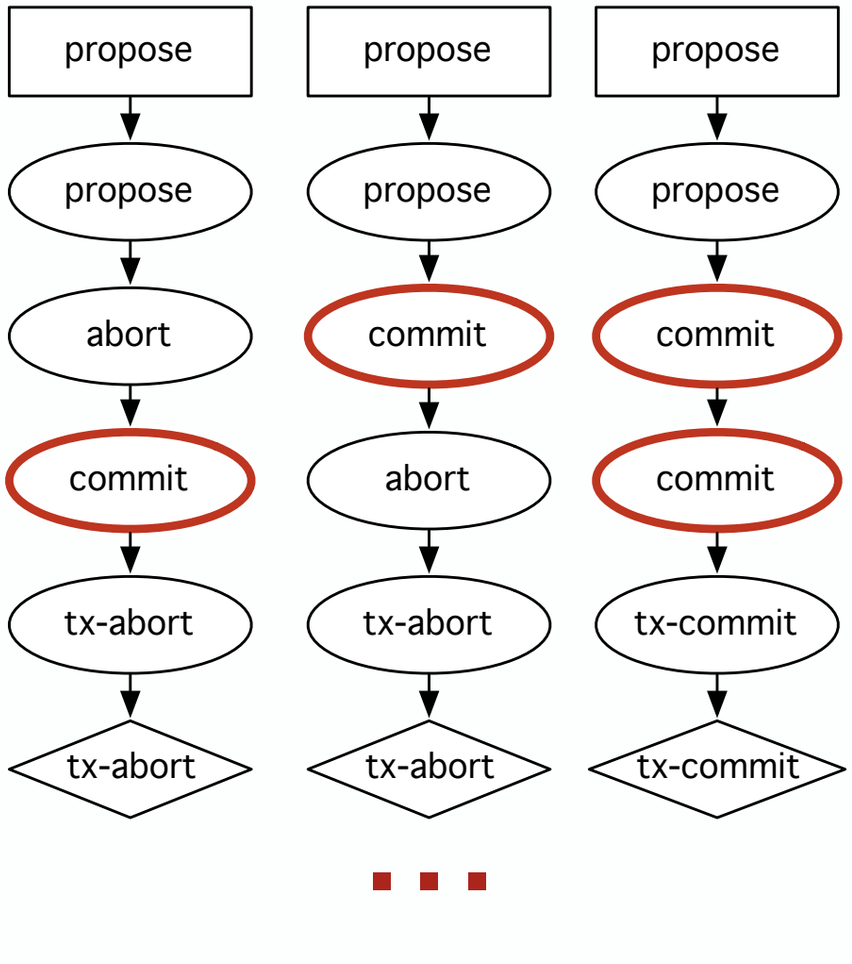
16

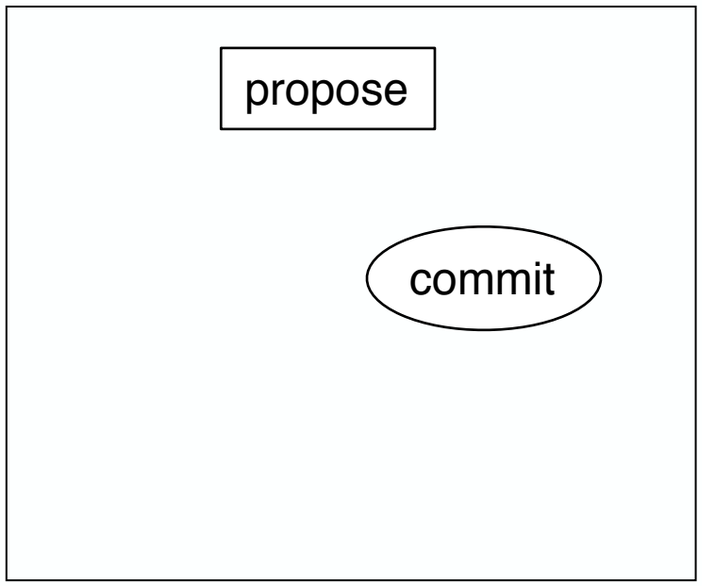
Synoptic overview

1/5. Parse log into a trace graph

2/5. Construct the initial model

3/5. Mine temporal invariants





A compact model with one node per event type

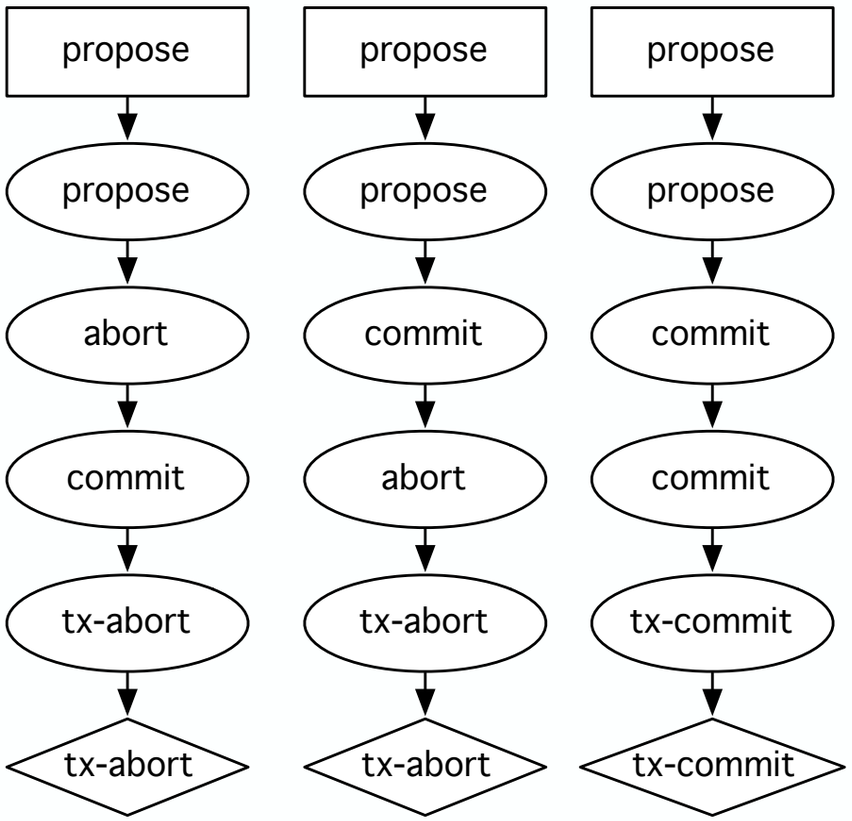
16

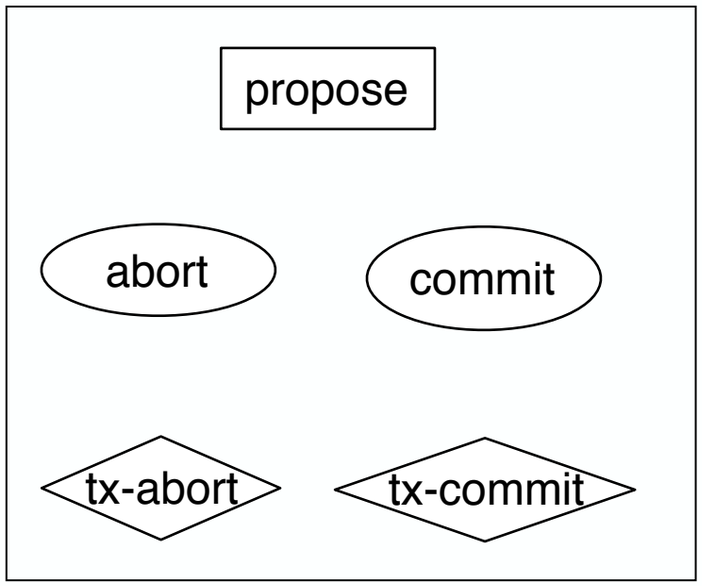
Synoptic overview

1/5. Parse log into a trace graph

2/5. Construct the initial model

3/5. Mine temporal invariants





A compact model with one node per event type

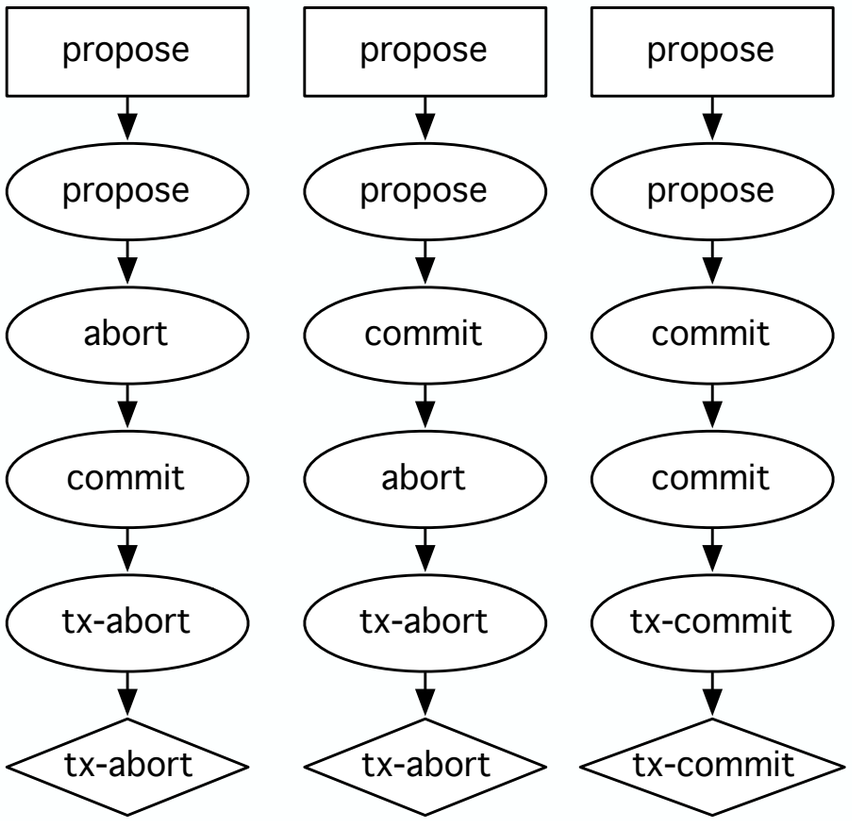
16

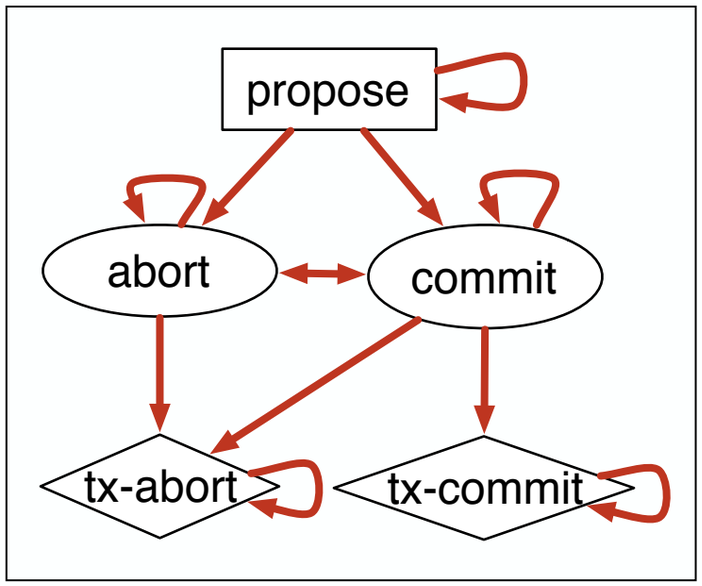
Synoptic overview

1/5. Parse log into a trace graph

2/5. Construct the initial model

3/5. Mine temporal invariants





A compact model with one node per event type

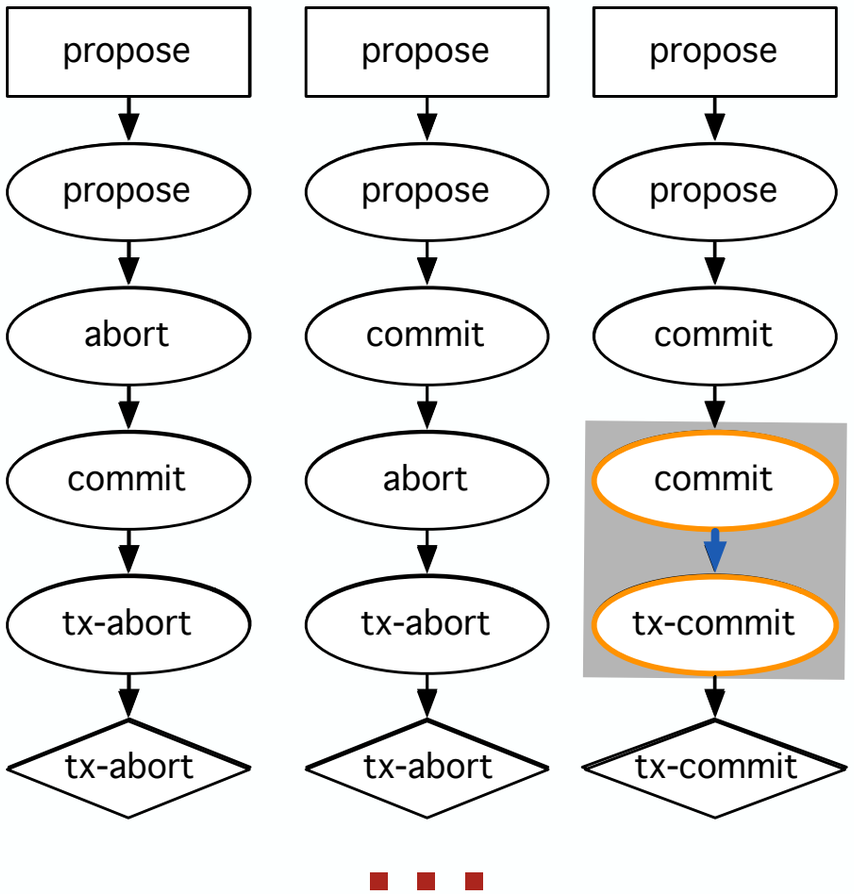
16

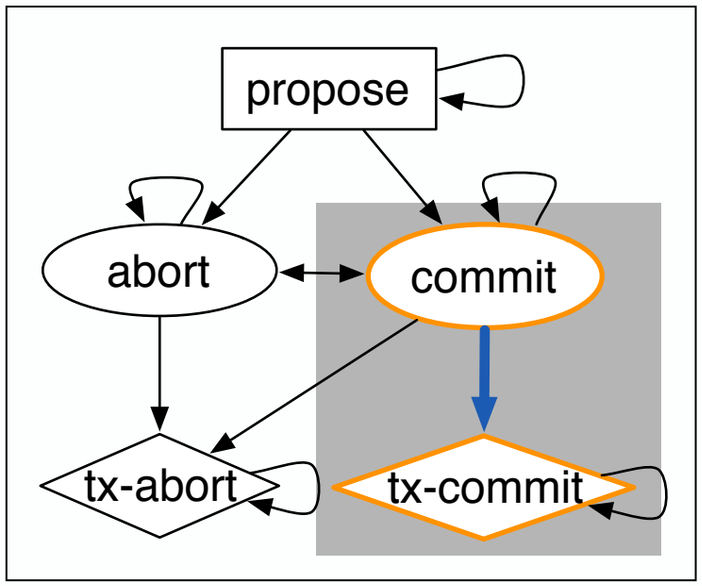
Synoptic overview

1/5. Parse log into a trace graph

2/5. Construct the initial model

3/5. Mine temporal invariants





A compact model with one node per event type

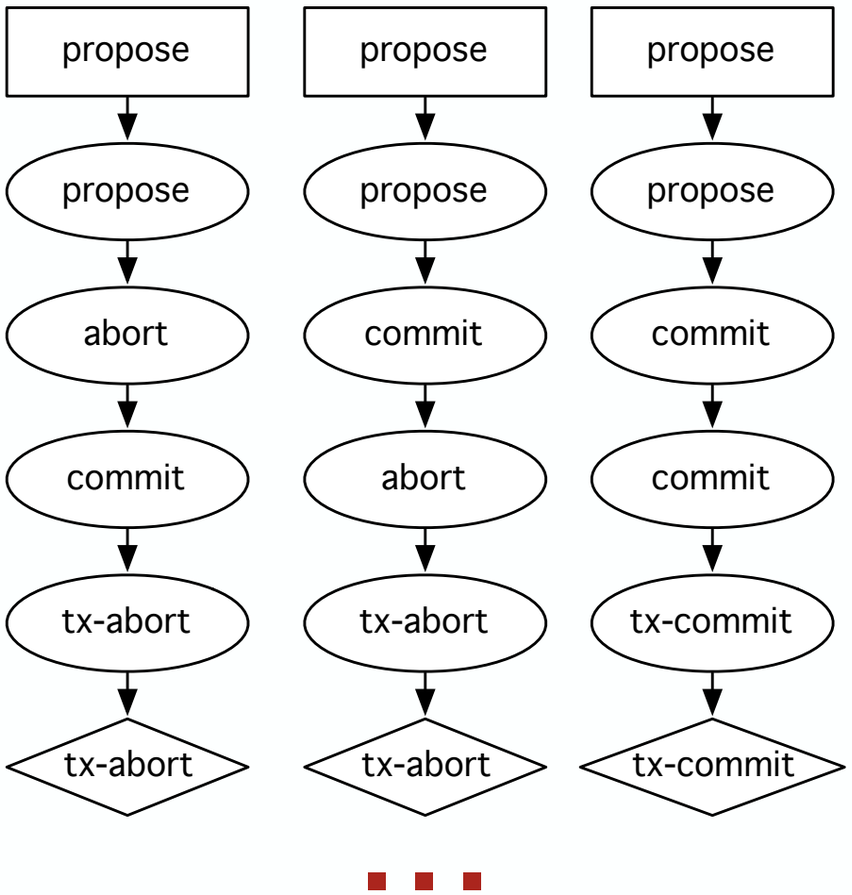
16

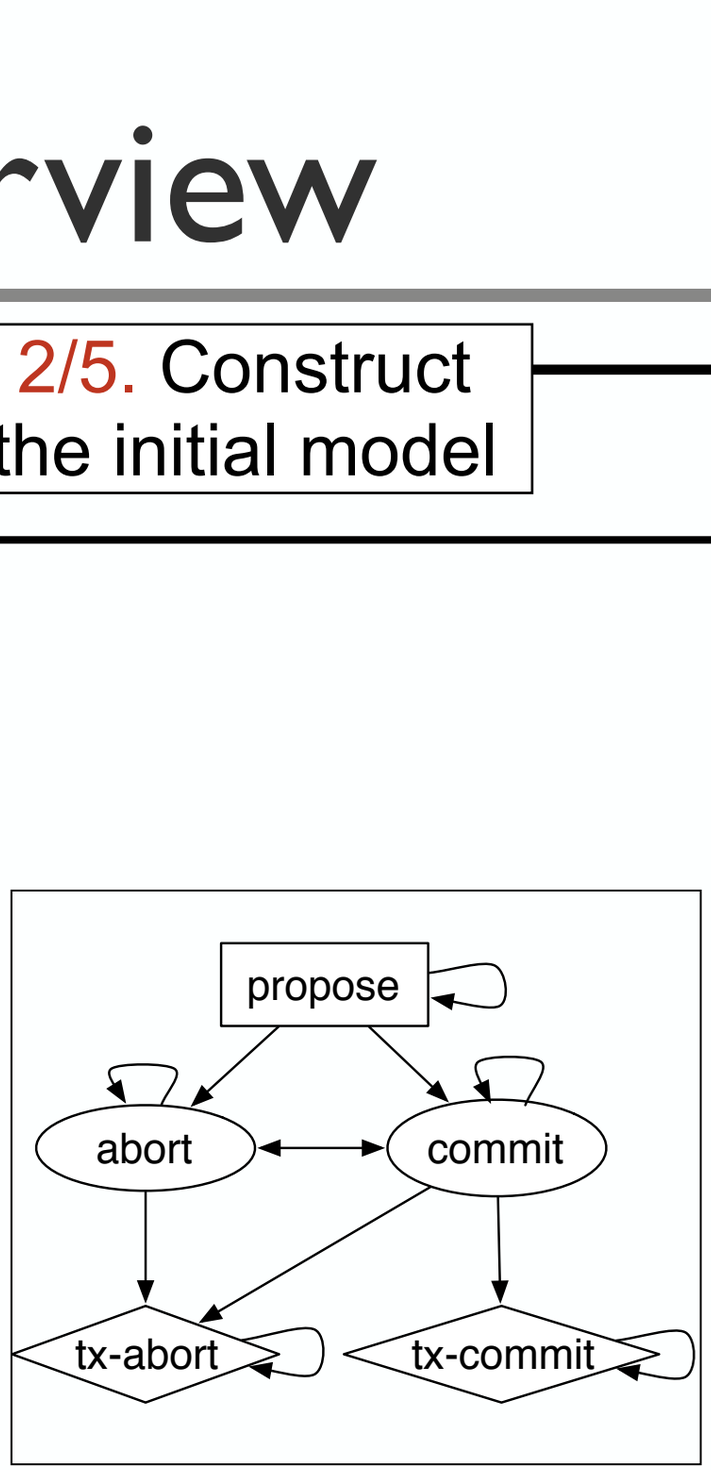
Synoptic overview

1/5. Parse log into a trace graph

2/5. Construct the initial model

3/5. Mine temporal invariants





A compact model with one node per event type

abort → tx-abort

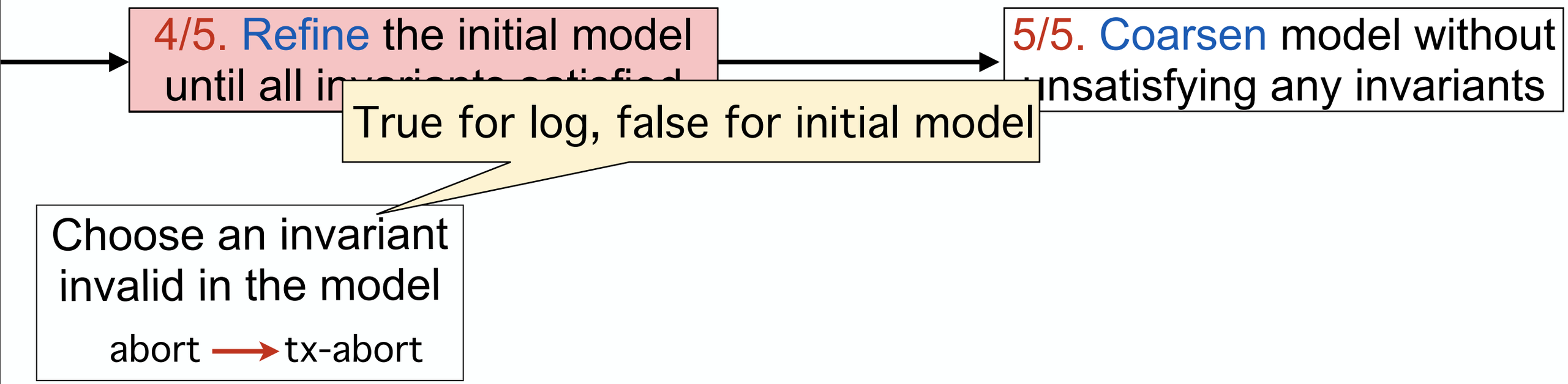
abort ↛ tx-commit

abort ← tx-abort

...

16

Synoptic overview



17

Synoptic overview

4/5. Refine the initial model until all invariants satisfied

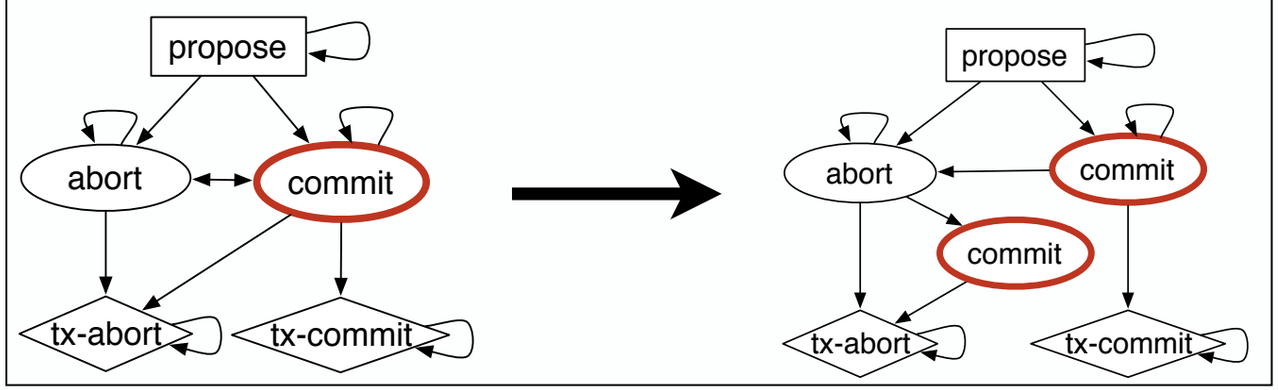
5/5. Coarsen model without unsatisfying any invariants

Choose an invariant invalid in the model

abort → tx-abort

Unsatisfied invariants exist

Refine a model node



17

Synoptic overview

4/5. Refine the initial model until all invariants satisfied

5/5. Coarsen model without unsatisfying any invariants

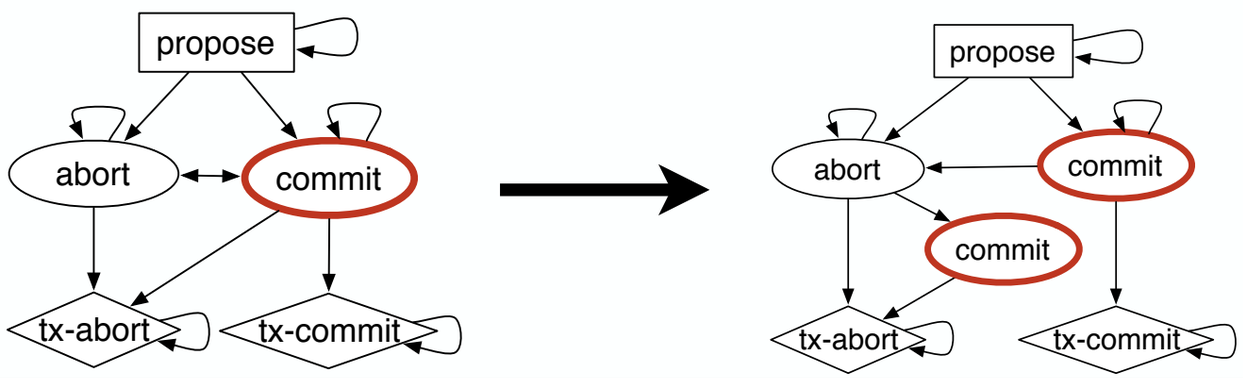
Choose an invariant invalid in the model

abort → tx-abort

Unsatisfied invariants exist

All invariants satisfied

Refine a model node

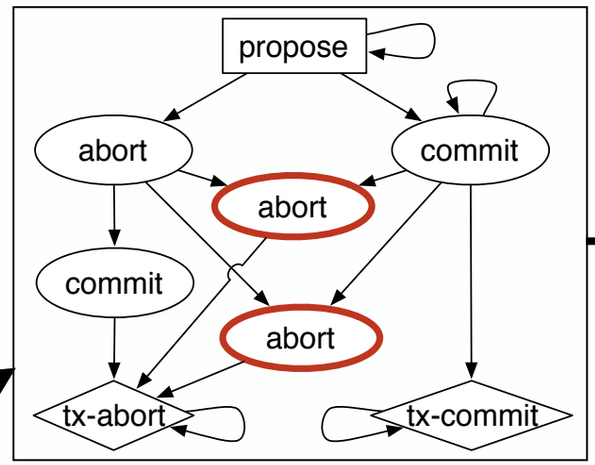


17

Synoptic overview

4/5. Refine the initial model until all invariants satisfied

5/5. Coarsen model without unsatisfying any invariants



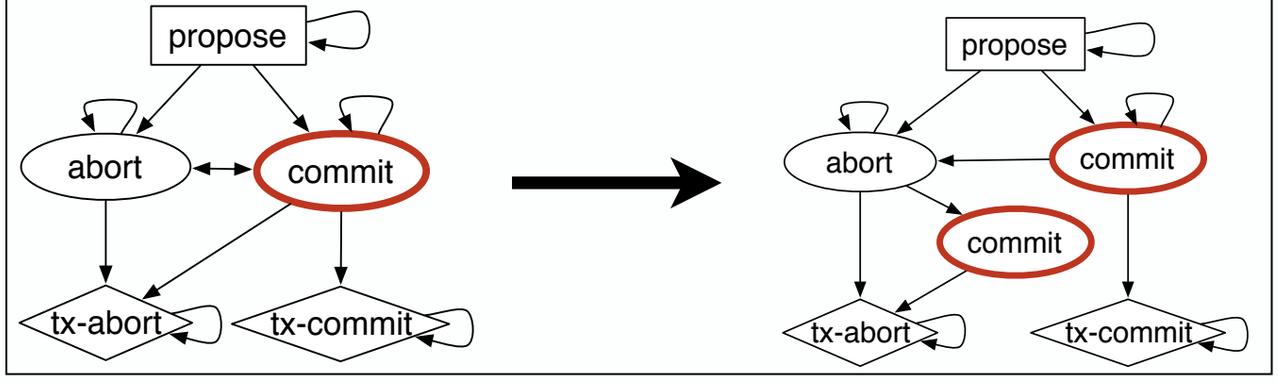
Choose an invariant invalid in the model

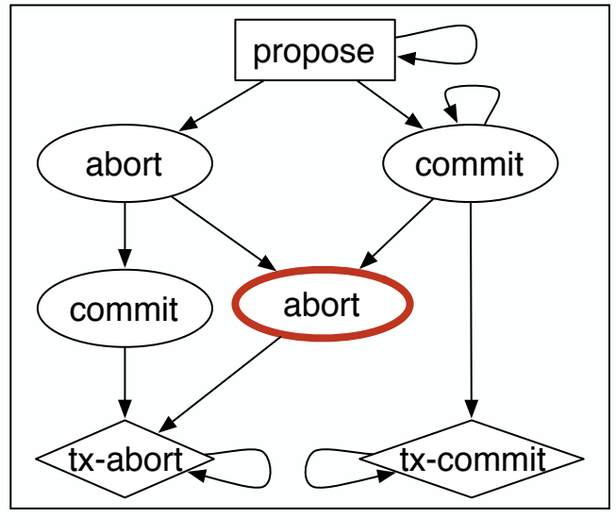
abort → tx-abort

Unsatisfied invariants exist

All invariants satisfied

Refine a model node





Merge nodes that exhibit the same behaviors

17

Assumptions

Logs

- Multiple executions
- Events in an execution are totally ordered

System

- Process generating the log can be modeled as an FSM
- Temporal system properties are evident in the orderings among events

18

Talk outline

- Motivation
- Synoptic's design
  - Overview
  - Assumptions
  - Temporal invariants
  - Model space exploration
- Evaluation results
- Future work
- Conclusion

19

Temporal log invariants

- Synoptic mines three kinds of temporal log invariants that are true for all the logged executions

Invariant	Example	Type
x → y always followed by	X Y y X X Y y X Y y y	liveness
x ← y always precedes	x y y x x y x z	safety
x ↛ y never followed by	x x y y y z	safety

Dwyer et al. ICSE 1999	Yang et al. PASTE 2004

20

Temporal log invariants

- Synoptic mines three kinds of temporal log invariants that are true for all the logged executions

Invariant	Example	Type
x → y always followed by	X Y y X X Y y X Y y y	liveness
x ← y always precedes	X Y Y x X Y x z	safety
x ↛ y never followed by	x x y y y z	safety

Dwyer et al. ICSE 1999	Yang et al. PASTE 2004

20

Temporal log invariants

- Synoptic mines three kinds of temporal log invariants that are true for all the logged executions

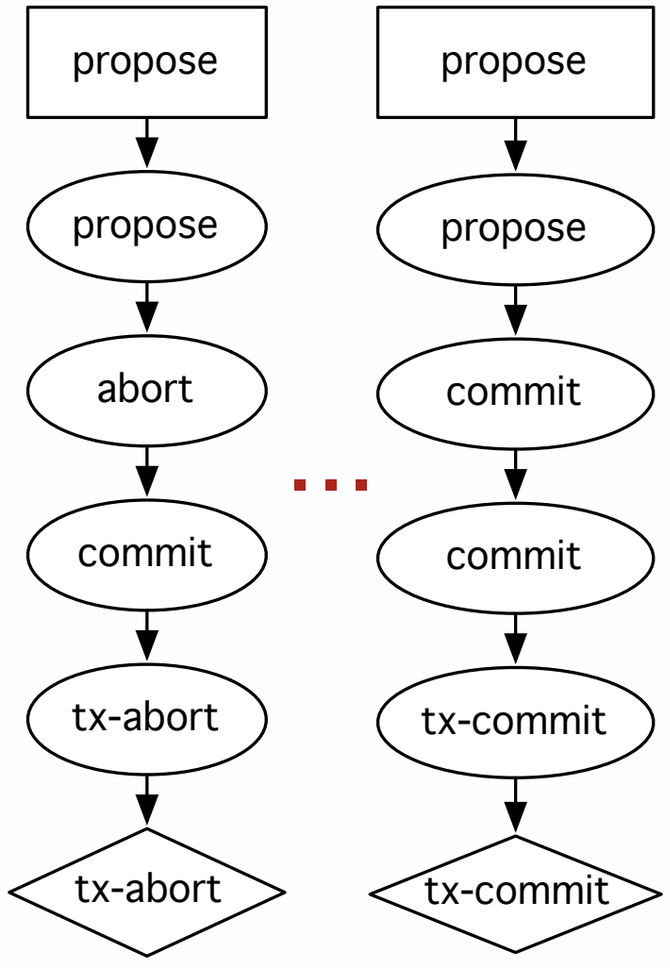
Invariant	Example	Type
x → y always followed by	X Y y X X Y y X Y y y	liveness
x ← y always precedes	X Y Y x X Y x z	safety
x ↛ y never followed by	X X y y y z	safety

Dwyer et al. ICSE 1999	Yang et al. PASTE 2004

20

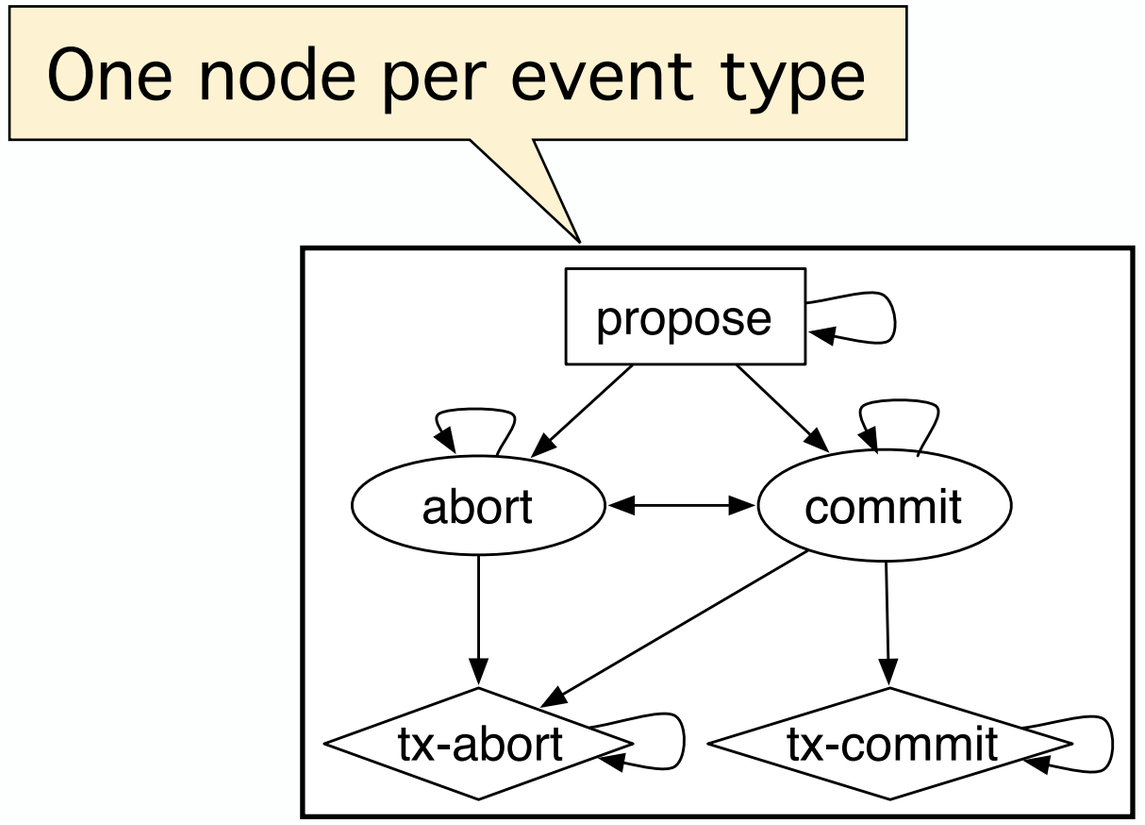
Two phase commit invariants

- Mined invariants:
  - abort ⟶ tx-abort
  - abort ↛ tx-commit
  - abort ⟵ tx-abort
  - commit ⟵ tx-commit
  - {abort, commit, tx-abort, tx-commit} ↛ propose
  - propose ⟵ {abort, commit, tx-abort, tx-commit}



Satisfying invariants with refinement

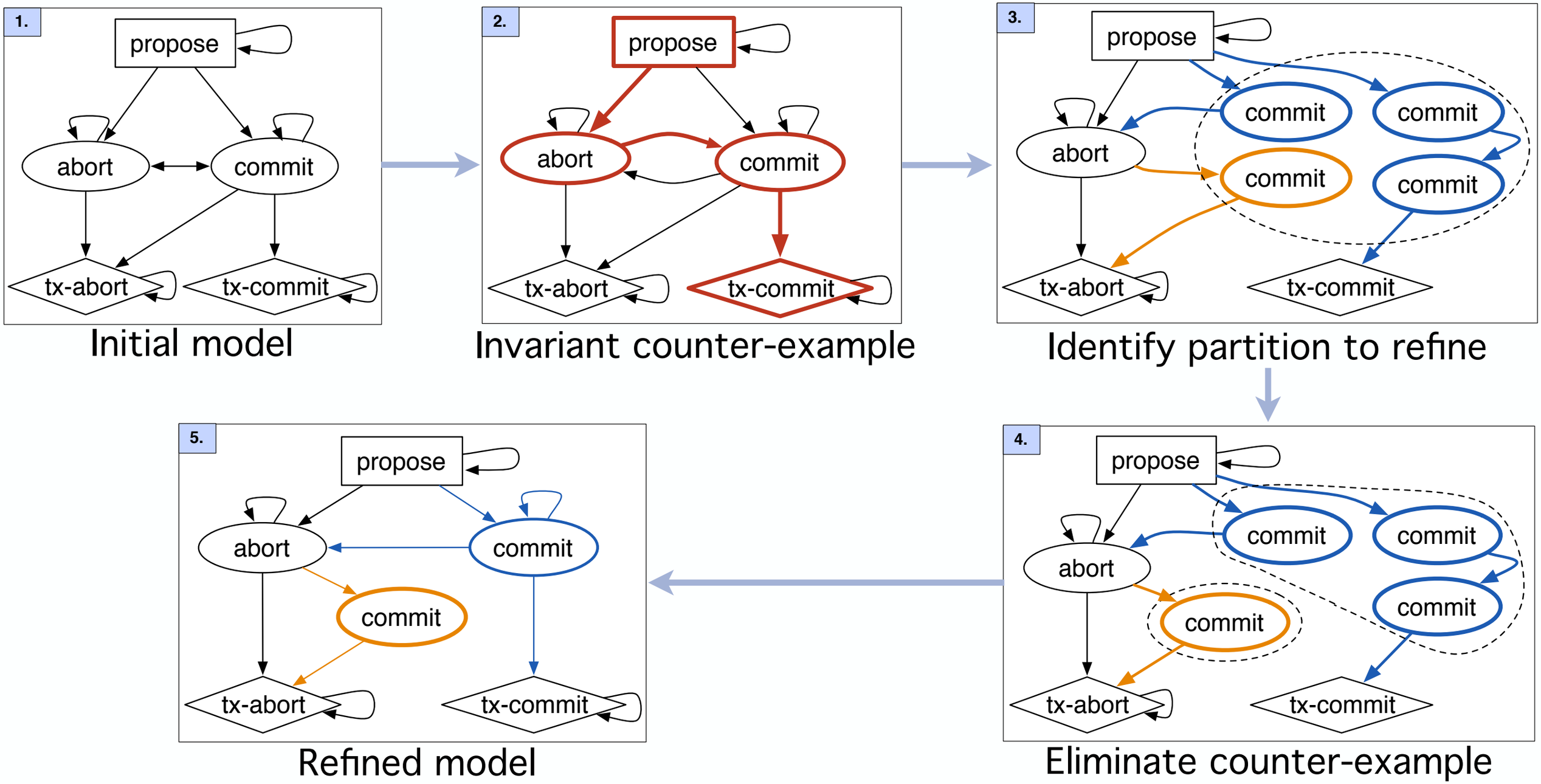
- The initial graph violates 3 mined invariants:
  - abort ⟶ tx-abort
  - abort ⟶/ tx-commit
  - abort ⟵ tx-abort
  - commit ⟵ tx-commit
  - {abort, commit, tx-abort, tx-commit} ⟶/ propose
  - propose ⟵ {abort, commit, tx-abort, tx-commit}



- For each violation find a corresponding path
- Use refinement to eliminate such path counter-examples

Counter-example based refinement

Example invariant: abort → tx-abort

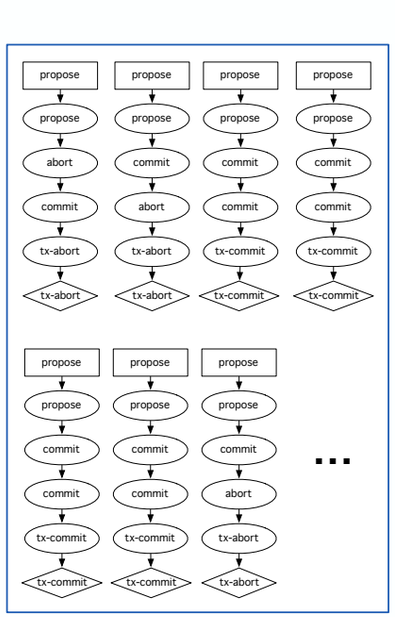


23

Model space

Larger models: fewer behaviors

Smaller models: more behaviors



Trace graph

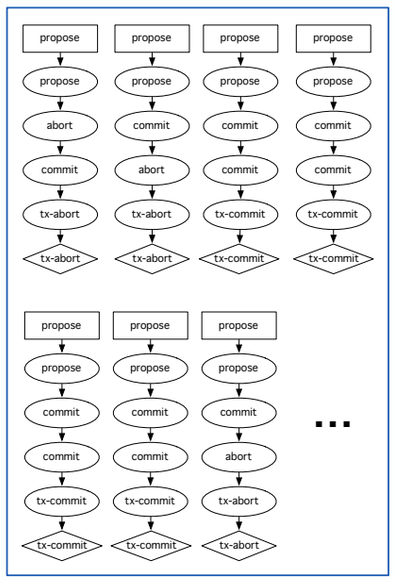
Initial model

24

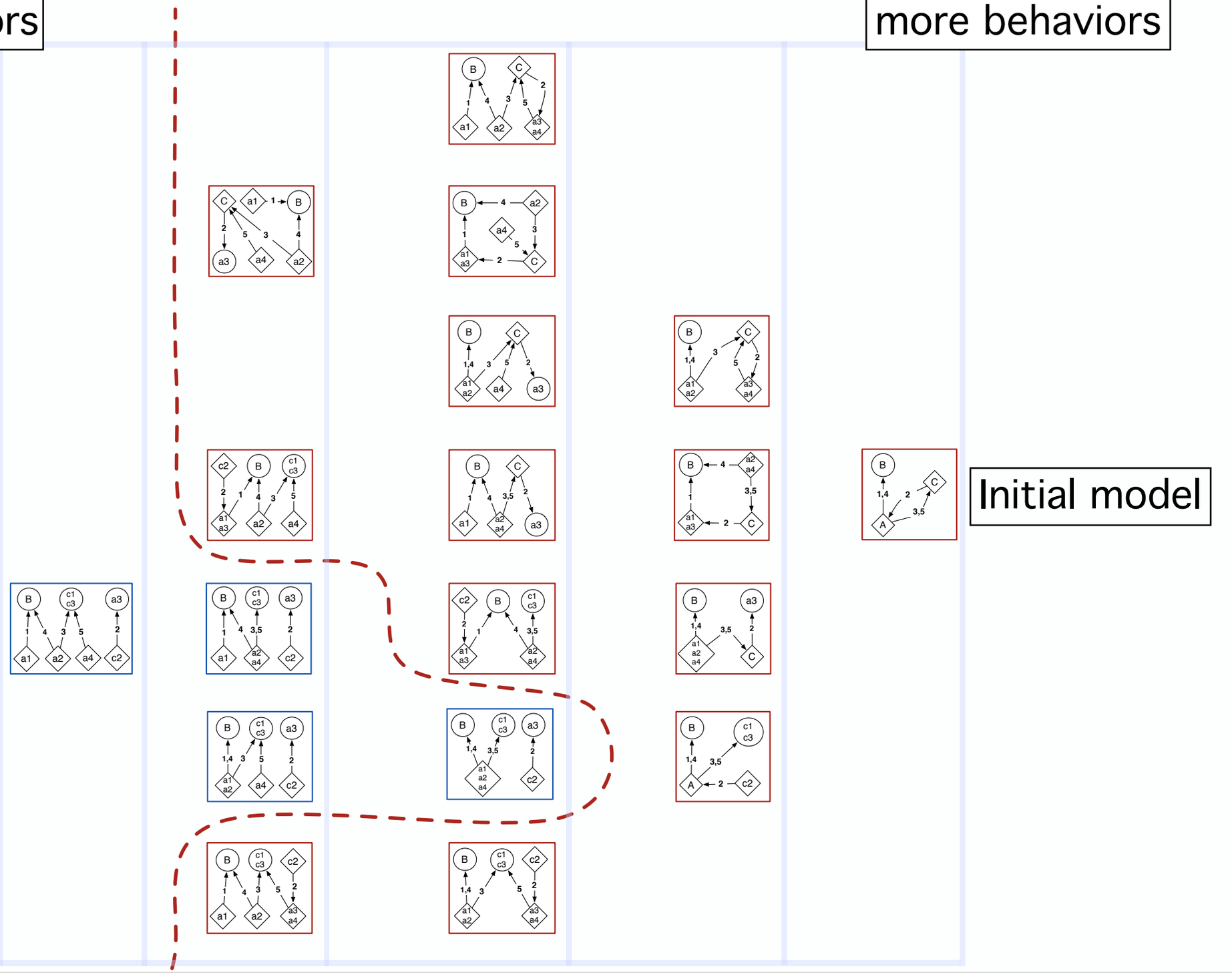
Model space

Larger models:
fewer behaviors

Smaller models:
more behaviors



Trace graph



Initial model

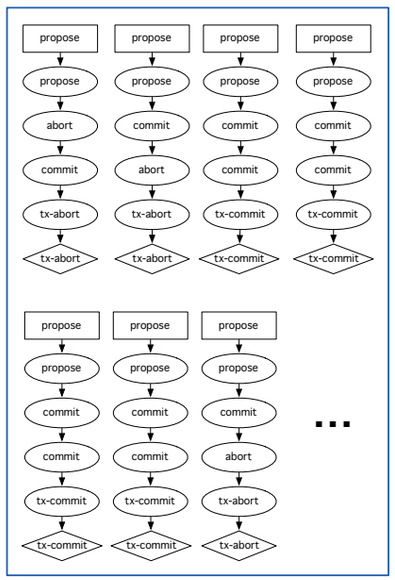
24

Model space

Larger models:
fewer behaviors

Smaller models:
more behaviors

Some invariants
falsified



Trace graph

Initial model

24

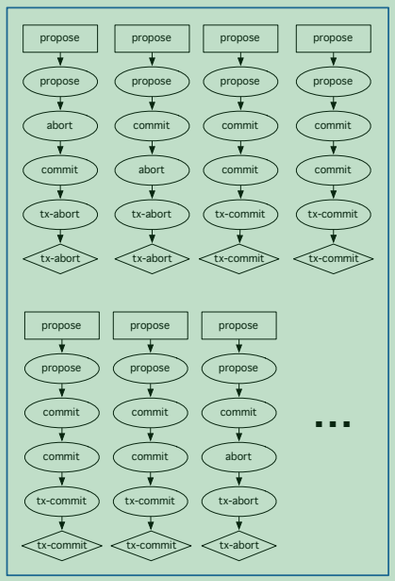
Model space

Larger models:
fewer behaviors

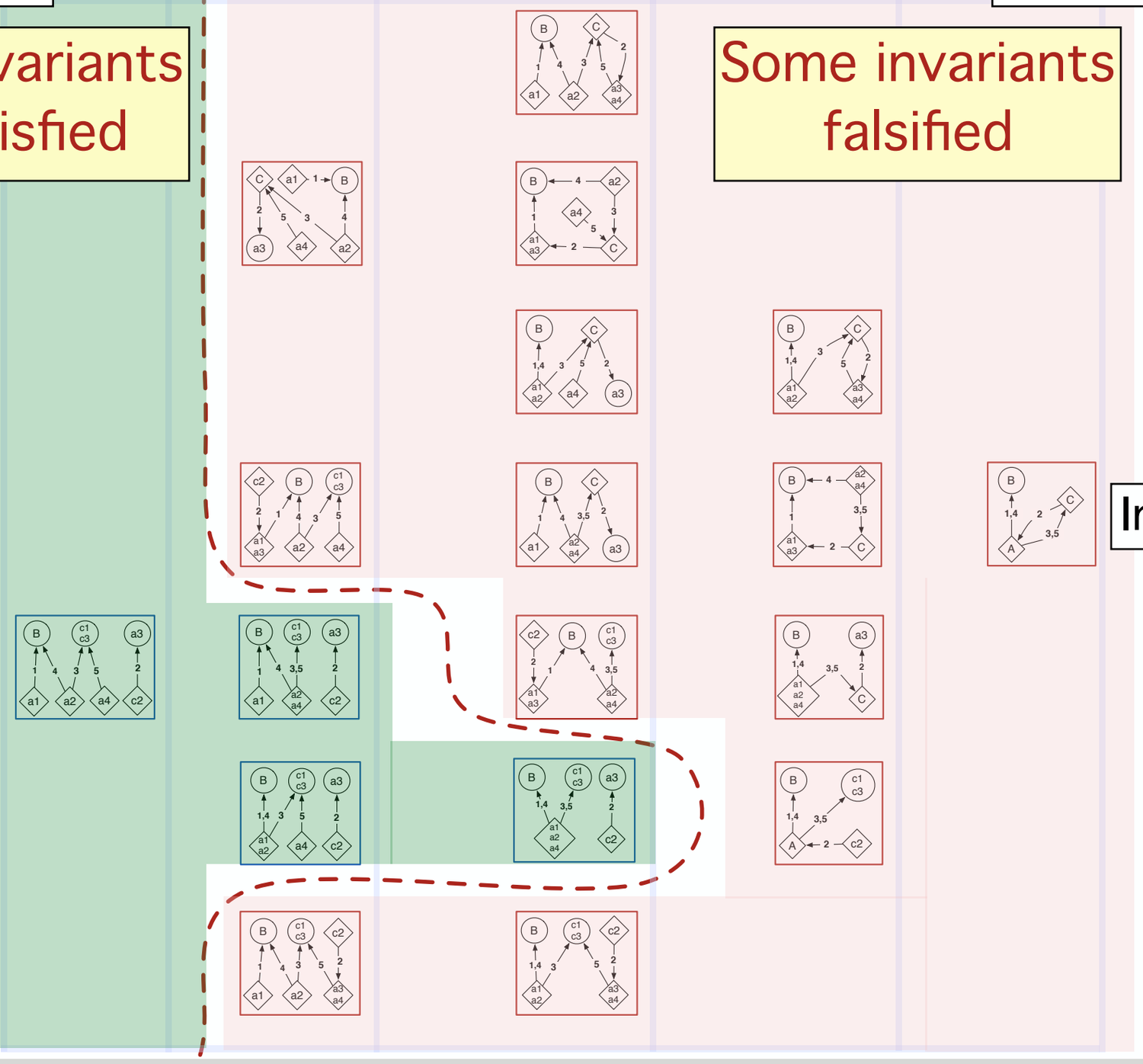
Smaller models:
more behaviors

All invariants satisfied

Some invariants falsified



Trace graph



Initial model

24

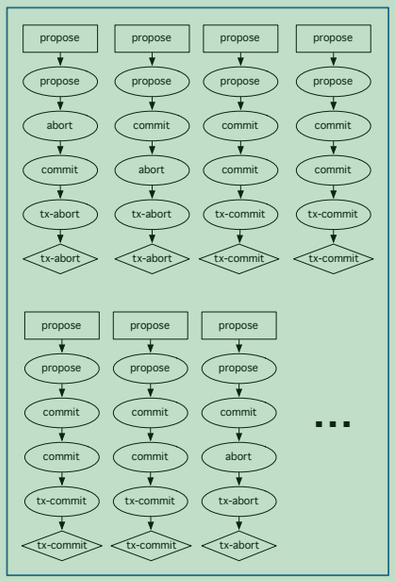
Model space

Larger models:
fewer behaviors

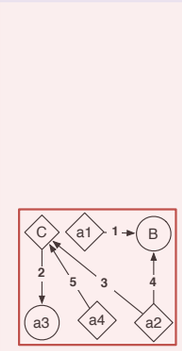
Smaller models:
more behaviors

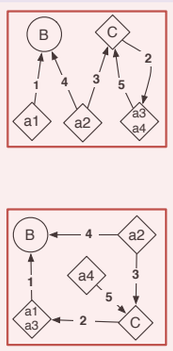
All invariants satisfied

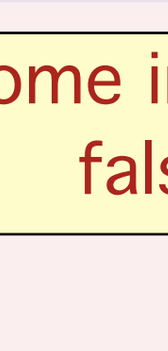
Some invariants falsified

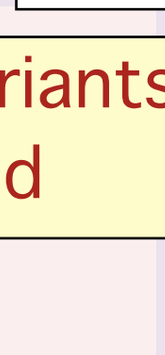


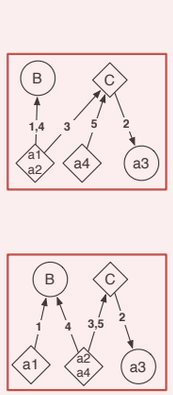
Trace graph

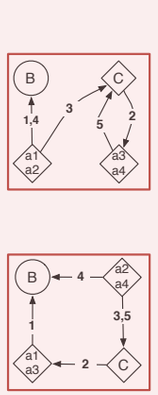


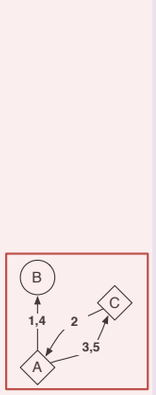


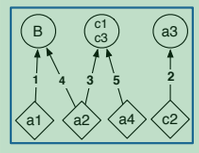


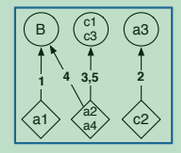


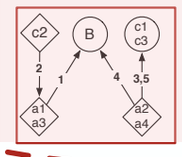


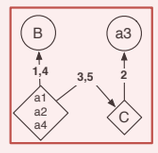


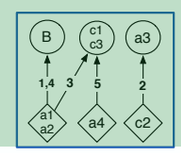


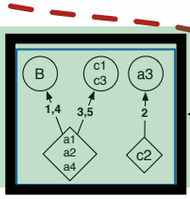


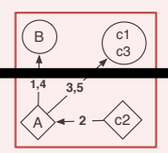


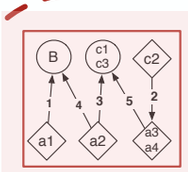


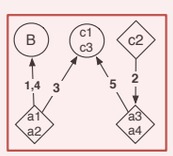












Initial model

Synoptic's goal:

Find the smallest model satisfying all invariants

24

Model space

Larger models:
fewer behaviors

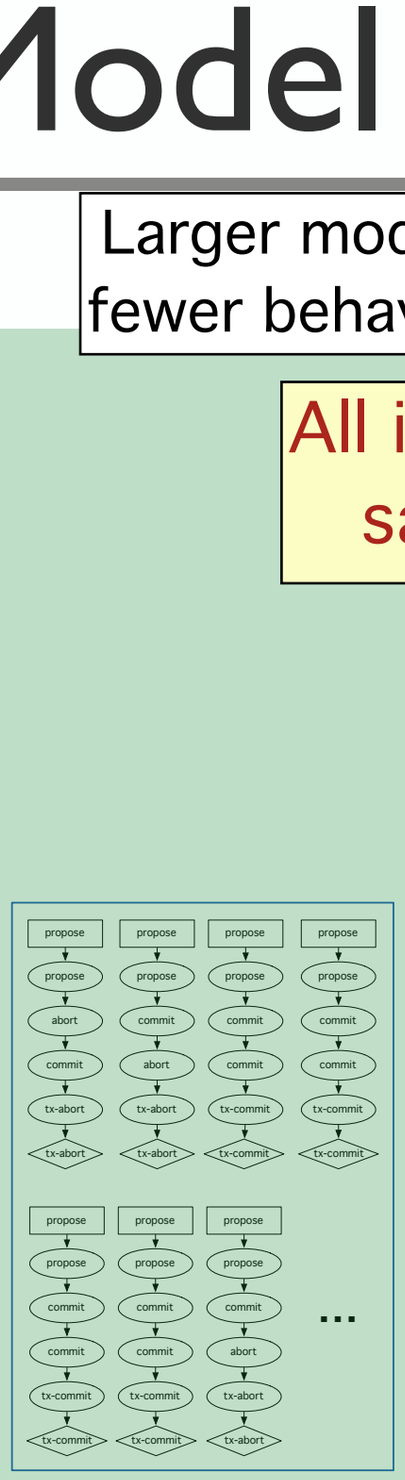
Smaller models:
more behaviors

All invariants satisfied

Some invariants falsified

Refinement

Coarsening



Trace graph

Initial model

Synoptic's goal:

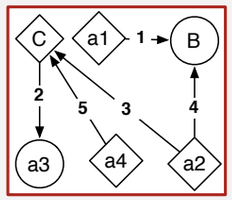
Find the smallest model satisfying all invariants

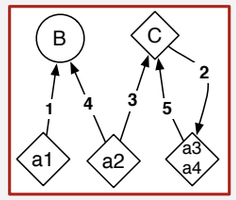
24

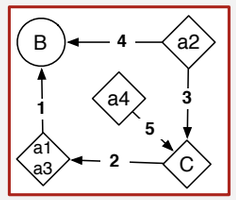
All invariants satisfied

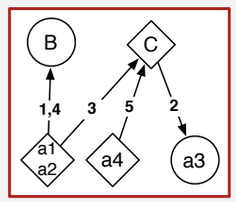
Some invariants satisfied

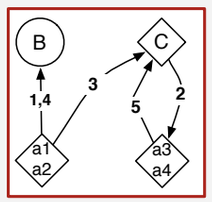
Synoptic finds a local optimum, not a global one

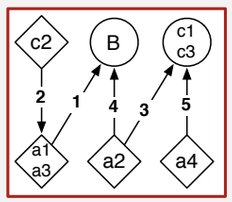


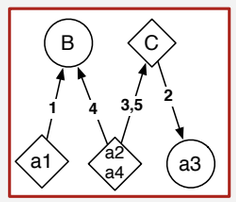


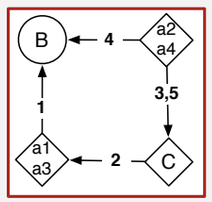


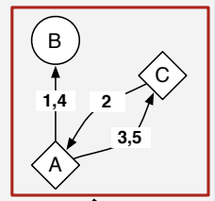






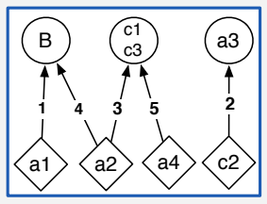


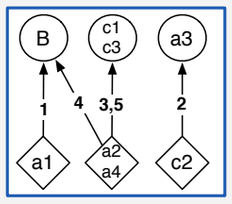


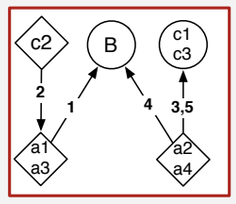


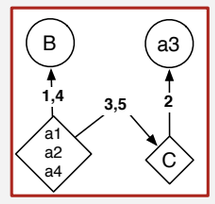
Initial model

Refine

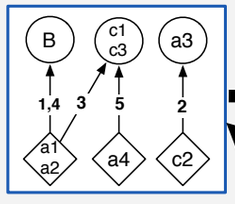




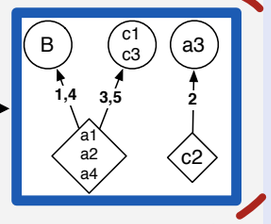


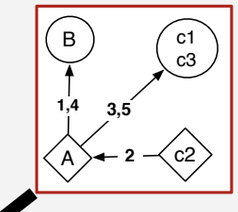


Select a coarsening that maintains all the invariants



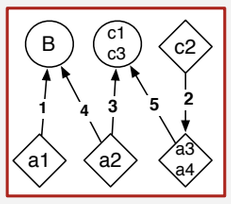
Coarsen

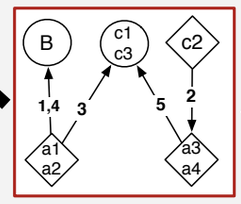




Refine

Refine





Select a refinement that satisfies an unsatisfied invariant

All invariants satisfied

Some invariants satisfied

Synoptic finds a local optimum, not a global one

- Coarsening from initial traces is inefficient
  - k-Tail Biermann et al. 1972 and GK-Tail Lorenzoli et al. ICSE 2008
- Refinement is well known Paige et al. 1987
  - Using mined invariant constraints is a new idea

Initial model

Refine

Select a coarsening that maintains all the invariants

Coarsen

Refine

Refine

Select a refinement that satisfies an unsatisfied invariant

Talk outline

- Motivation
- Synoptic's design
- Evaluation results
  - Formal evaluation
  - Reverse traceroute
  - Distributed systems course
- Future work
- Conclusion

26

Formal evaluation

- Termination
  - Always finds a model
- Correctness
  - Satisfies all the mined invariants, and no others
- Conciseness
  - Local minimum
- Efficiency
  - Never violate a mined invariant
  - Make progress towards satisfying some mined invariant

See the paper for more

27

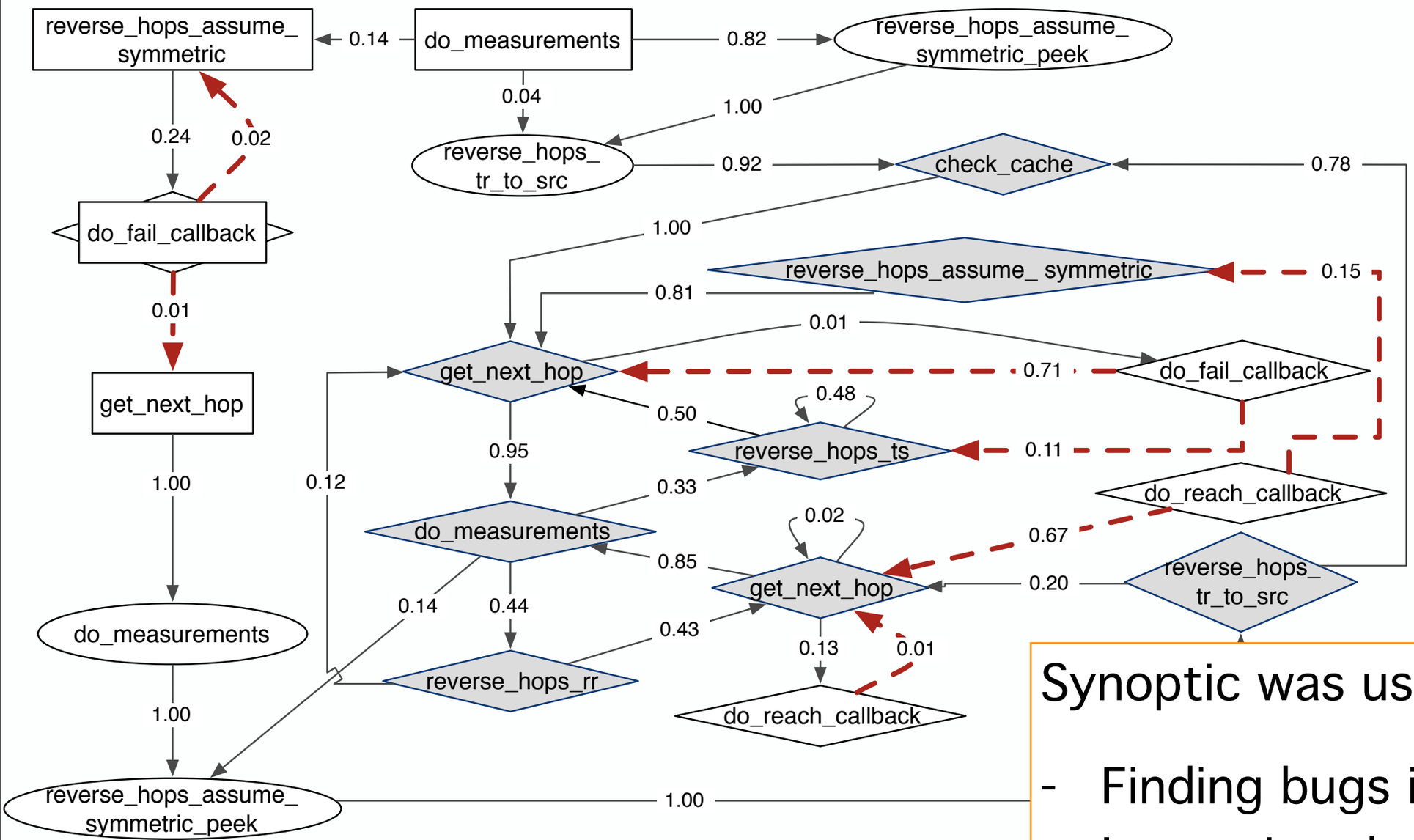
Reverse traceroute case-study

- Reverse traceroute KATZ-BASSETT et al. NSDI 2010
  - Finds the likely reverse traceroute from an arbitrary destination to a source host
  - In live deployment for over a year
  - Deployed internally by a large Internet company

- Ran Synoptic on reverse traceroute server logs
  - Developer added 16 lines of logging code
  - Synoptic processed the 900,000 events in 12 minutes

28

Reverse traceroute case-study



2 bugs found:

- Shaded nodes should not be terminal
- Red edges should not exist

Synoptic was useful for:

- Finding bugs in code
- Increasing developer confidence
- Building system understanding

29

Synoptic limitations

- Quality and usefulness of the model depends on granularity of logging statements and user input
- Only considers three kinds of temporal invariants
- For some applications, the small model size may be irrelevant
- Online log analysis is not supported
- The input log must be totally ordered

30

Future work

- Handle logs generated by distributed systems
- Generalize invariants
  - Relax always and never constraints — Lou et al. KDD 2010
  - Explore composition of temporal invariants — Gabel et al. FSE 2008
  - Incorporating structural/data invariants — Ernst et al. TSE 2001
- Tool improvements
  - Explicating model paths in the GUI
  - Eclipse plug-in for ease of use
- Automatically infer log format — Vaarandi et al. IFIP ICICS 2004; Zhu et al. OSR 2010

31

Contributions

Synoptic: a tool that extracts models from logs

- Specification mining algorithm
  - Uses refinement for efficiency
  - Mines and satisfies temporal log invariants for accuracy
- Proved important properties of refinement
- Showed that Synoptic helps developers to study large and complex logs, and can help find bugs

Try it!

http://synoptic.googlecode.com

32