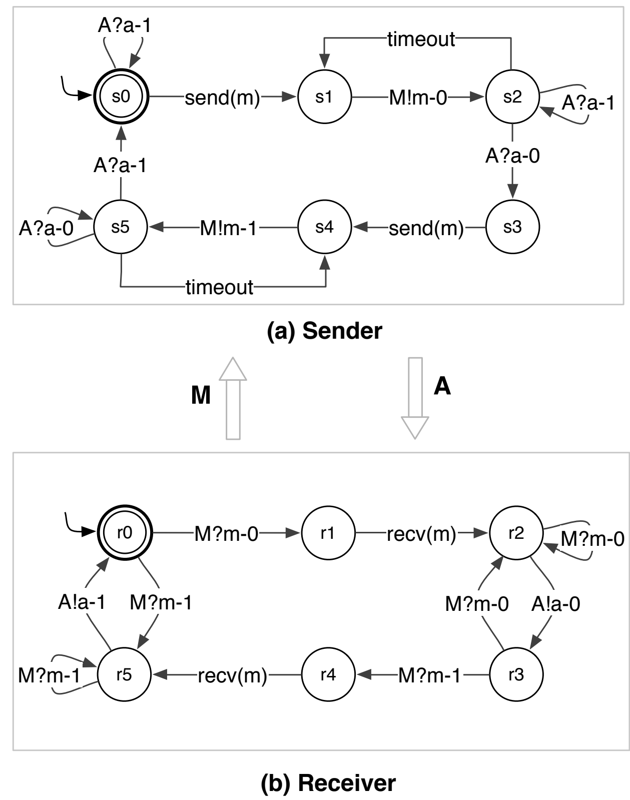
Inferring Models of Concurrent Systems from Logs of Their Behavior with CSight

Sender	Receiver
1,0 send(m)	2,1 M?m-0
2,0 M!m-0	2,2 recv(m)
3,3 A?a-0	2,3 A!a-0
4,3 send(m)	5,4 M?m-1
5,3 M!m-1	5,5 recv(m)
6,6 A?a-1	5,6 A!a-1
7,6 send(m)	8,7 M?m-0
8,6 M!m-0	8,8 recv(m)
9,9 A?a-0	8,9 A!a-0
10,9 send(m)	11,10 M?m-1
11,9 M!m-1	11,11 recv(m)
12,12 A?a-1	11,12 A!a-1

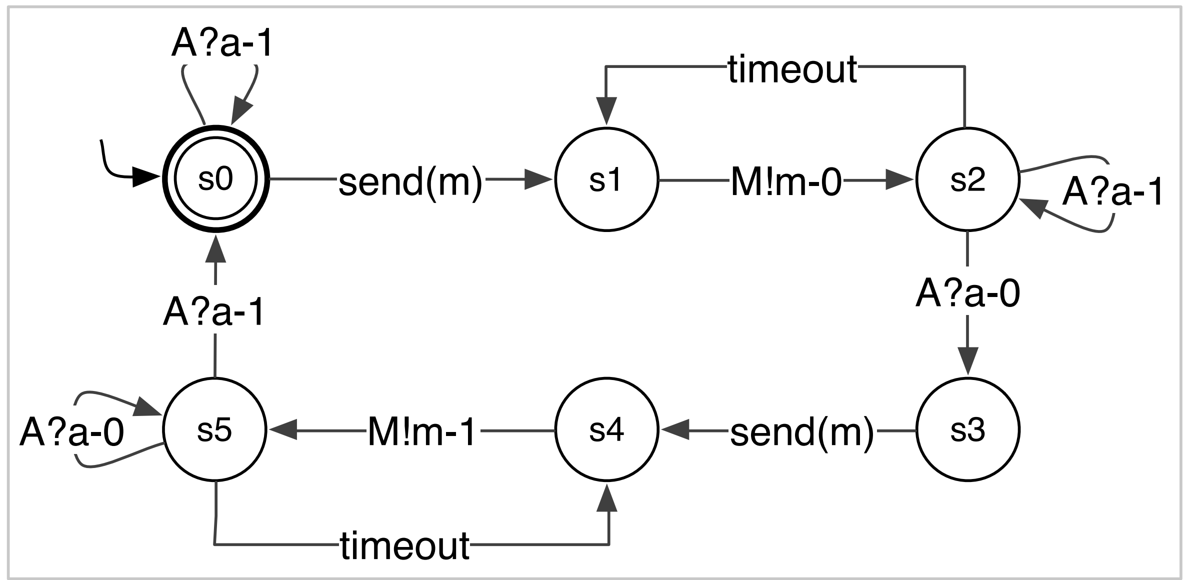


Ivan Beschastnikh
Yuriy Brun
Michael D. Ernst
Arvind Krishnamurthy

University of British Columbia
UMass Amherst
University of Washington

Sender	Receiver
1,0 send(m)	2,1 M?m-0
2,0 M!m-0	2,2 recv(m)
3,3 A?a-0	2,3 A!a-0
4,3 send(m)	5,4 M?m-1
5,3 M!m-1	5,5 recv(m)
6,6 A?a-1	5,6 A!a-1
7,6 send(m)	8,7 M?m-0
8,6 M!m-0	8,8 recv(m)
9,9 A?a-0	8,9 A!a-0
10,9 send(m)	11,10 M?m-1
11,9 M!m-1	11,11 recv(m)
12,12 A?a-1	11,12 A!a-1

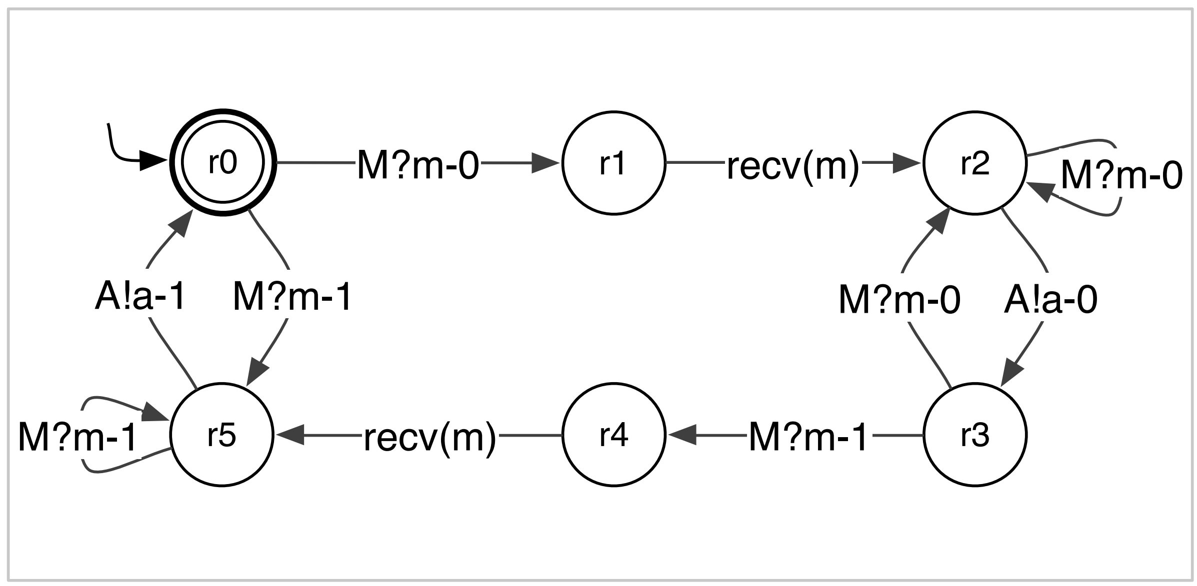
CSight



(a) Sender

M

A



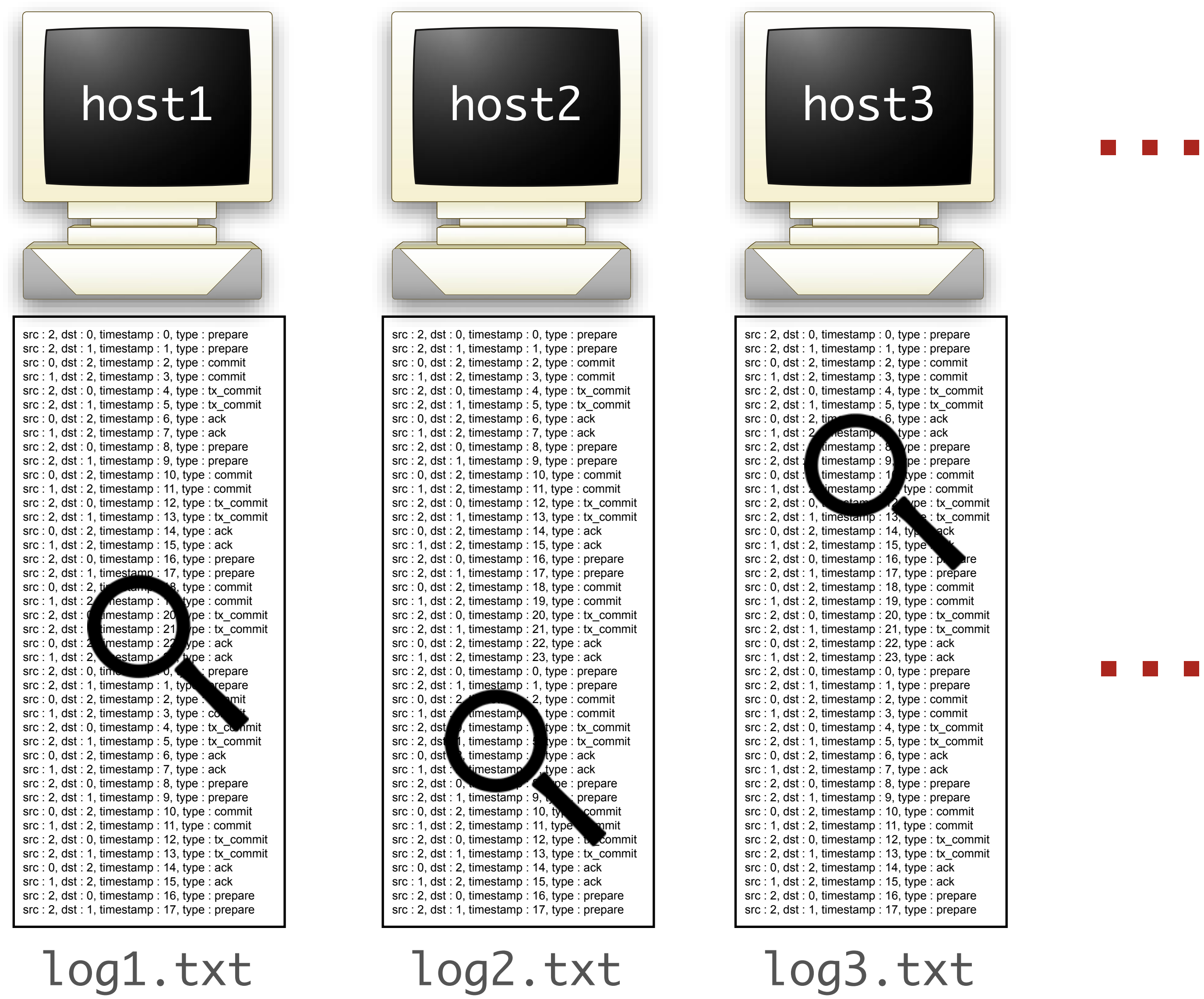
(b) Receiver

Input log

Output model

Challenges of logging for debugging

- Logs present a low-level view of the system
  - Developers are not sure what to look for, or where
- Reasoning about logged concurrency is a nightmare
  - Must connect clues across host and execution logs

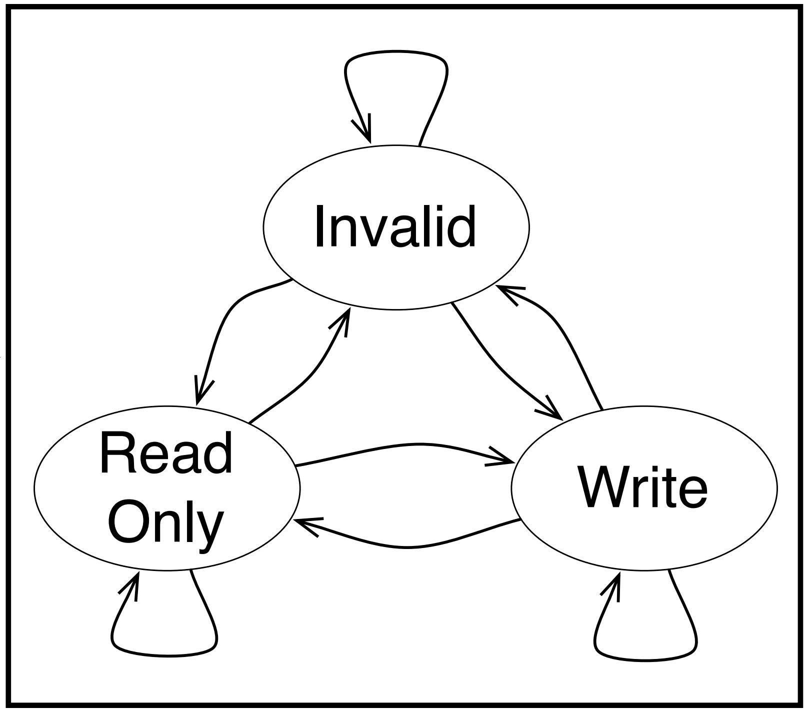


Models from observations

```
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

Model inference



Output

Models from observations

```
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
...
```

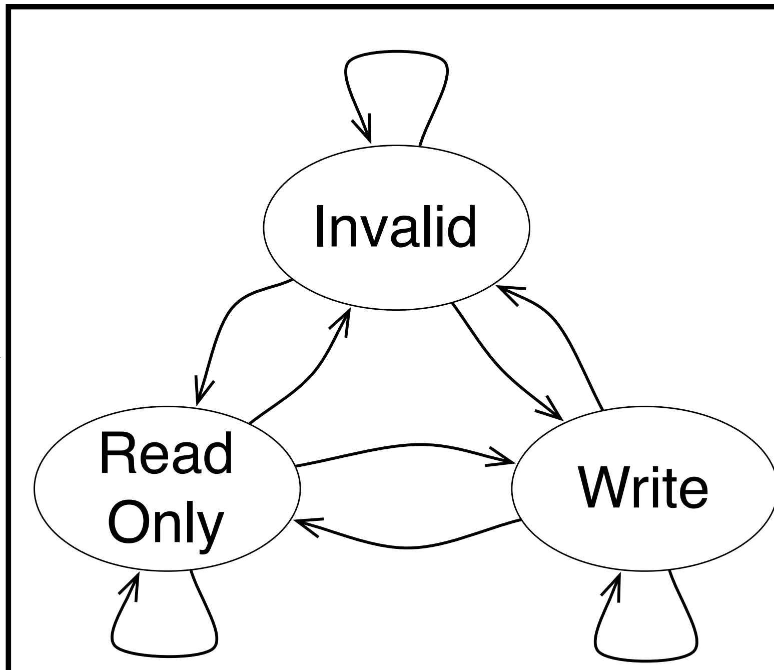
Input

Prior work:

Cook et al. TSE 1998
Lorenzoli et al. ICSE 2008
Lo et al. ASE 2010
Beschastnikh et al. FSE 2011
Ghezzi et al. ICSE 2014

Model inference

- Specification mining
- Process discovery



Output

Using inferred models

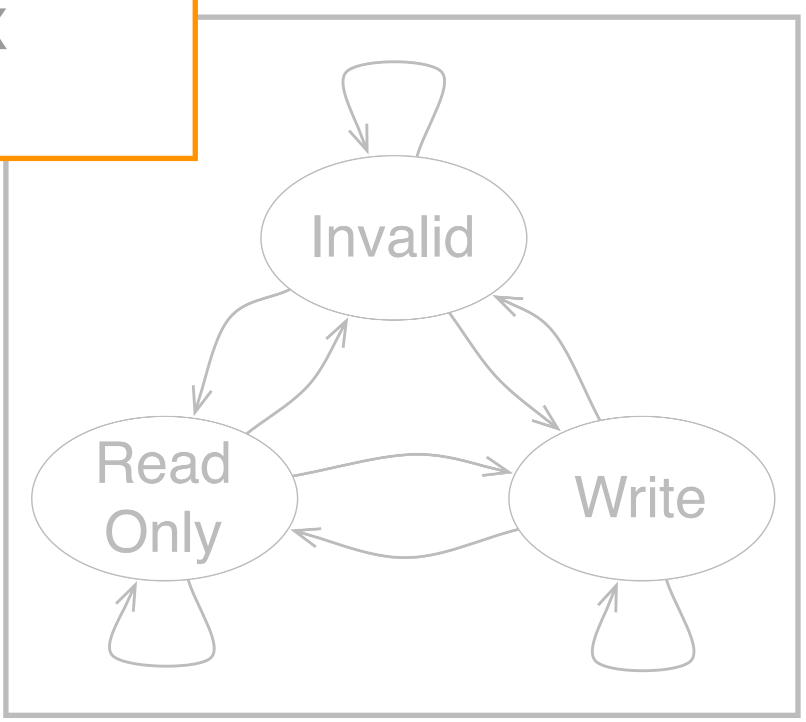
Input

Applications:

- Mental model validation
- Test case generation
- Evaluating test suites
- Anomaly detection
- Log summarization
- Verifying a code fix

...

Model inference



Output

Model inference and concurrency

Input

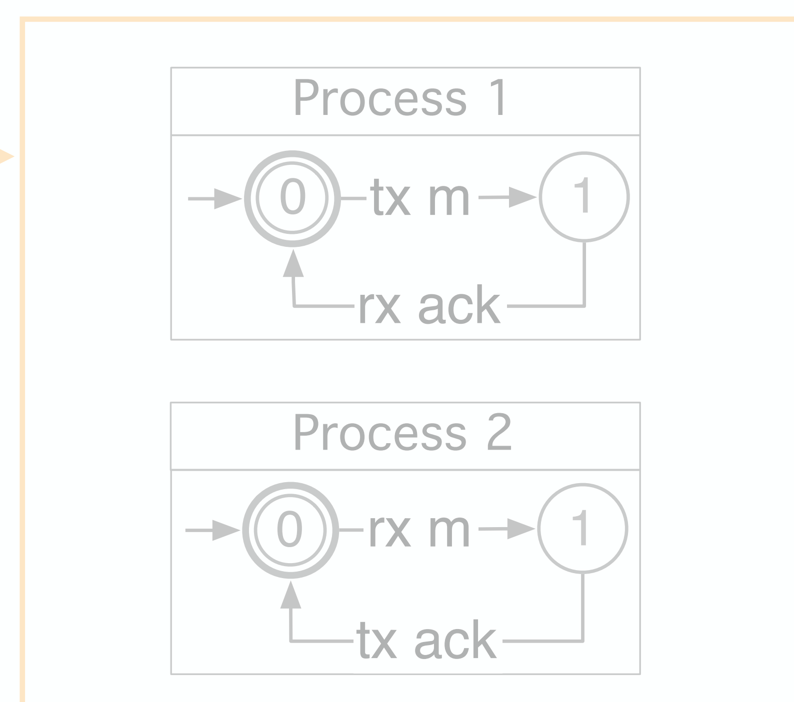
Challenges:

- Modeling concurrency
- Accuracy
- Efficiency

Contributions:

- Communicating FSM inference
- Refinement
- Leveraging mined invariants

CSight



Output

Cook et al. FSE 1998
Kumar et al. ICSE 2012

CSight overview

Challenges:

- Modeling concurrency
- Accuracy
- Efficiency

Contributions:

- Communicating FSM inference
- Refinement
- Leveraging mined invariants

Key results:

- CSight infers concise models of distributed systems: distributed hash table, and TCP
- User study with 39 students shows that CFSM models are helpful in findings bugs
- Proved termination/progress properties

CSight talk outline

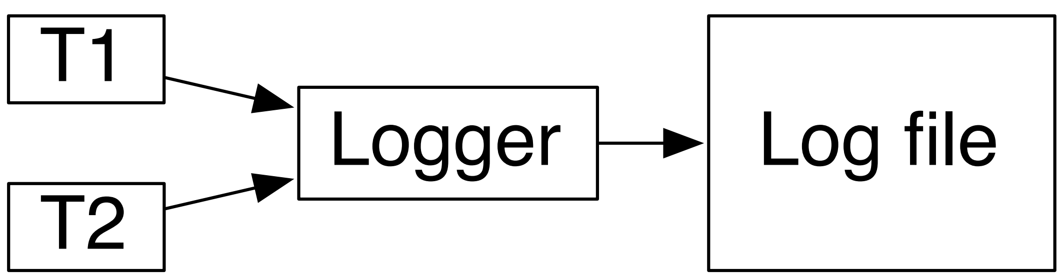
- Motivation
- Background
  - Logging: partial order and vector clocks
  - Modeling: communicating FSMs
- CSight approach
- Evaluation

http://synoptic.googlecode.com

Limitations of total order

- A system with two threads: T1, T2
  - T1 generates event a, T2 generates event b

- Logging pipeline:

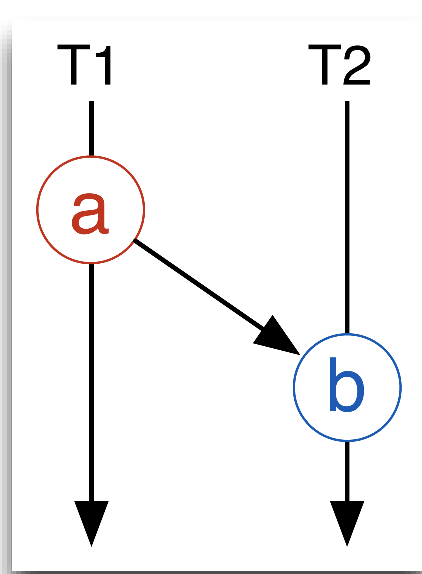


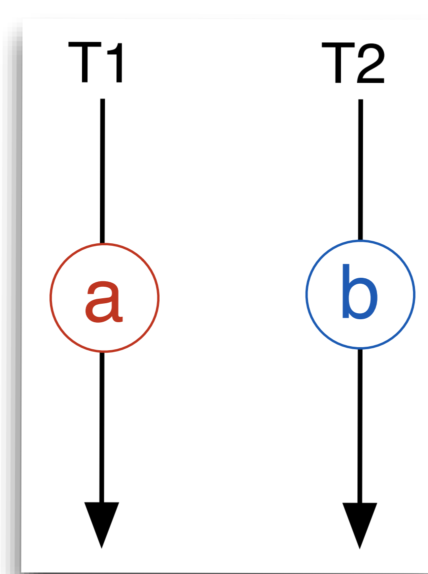
- Generated log file:

1	a
2	b

Which of these two systems generated the log?

Two possible systems:

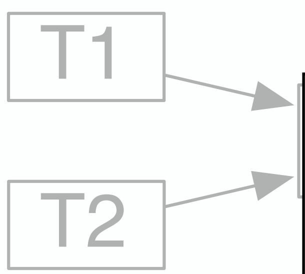




Limitations of total order

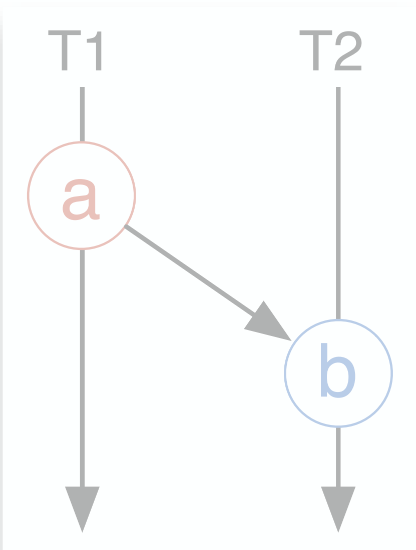
- A system with two threads: T1, T2
  - T1 generates event a, T2 generates event b

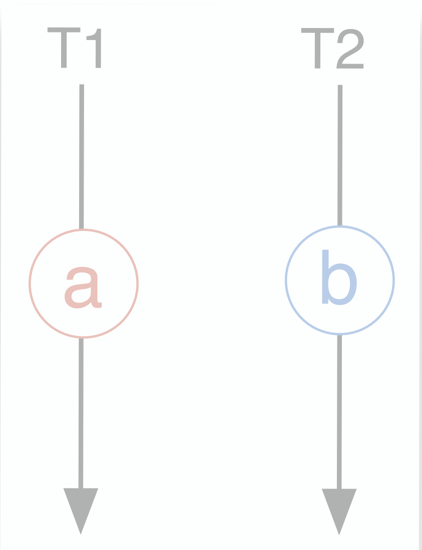
- Logging pipeline:



- Generated log file:

A totally ordered log is insufficient.





Logging the partial order

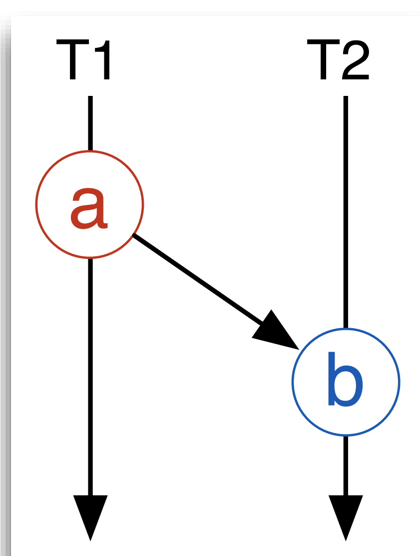
- We know how to do this
  - Lamport defined the happens-before relation in 1978
  - Operationalized with vector clocks in | Fidge, ACSS 1988 | Mattern, PDA 1989 |

- Vector clocks capture the partial order of an execution
  - Track dependencies between processes
  - Tell us if two events are concurrent, or if one occurs before the other.

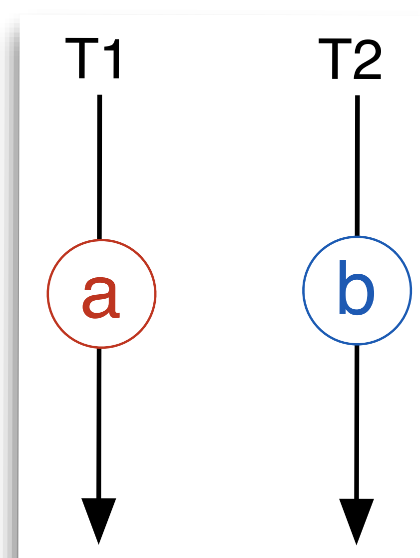
Logging the partial order

- We know how to do this
  - Lamport defined the happens-before relation in 1978
  - Operationalized with vector clocks in Fidge, ACSS 1988 | Mattern, PDA 1989

[1,0]	a
[1,1]	b

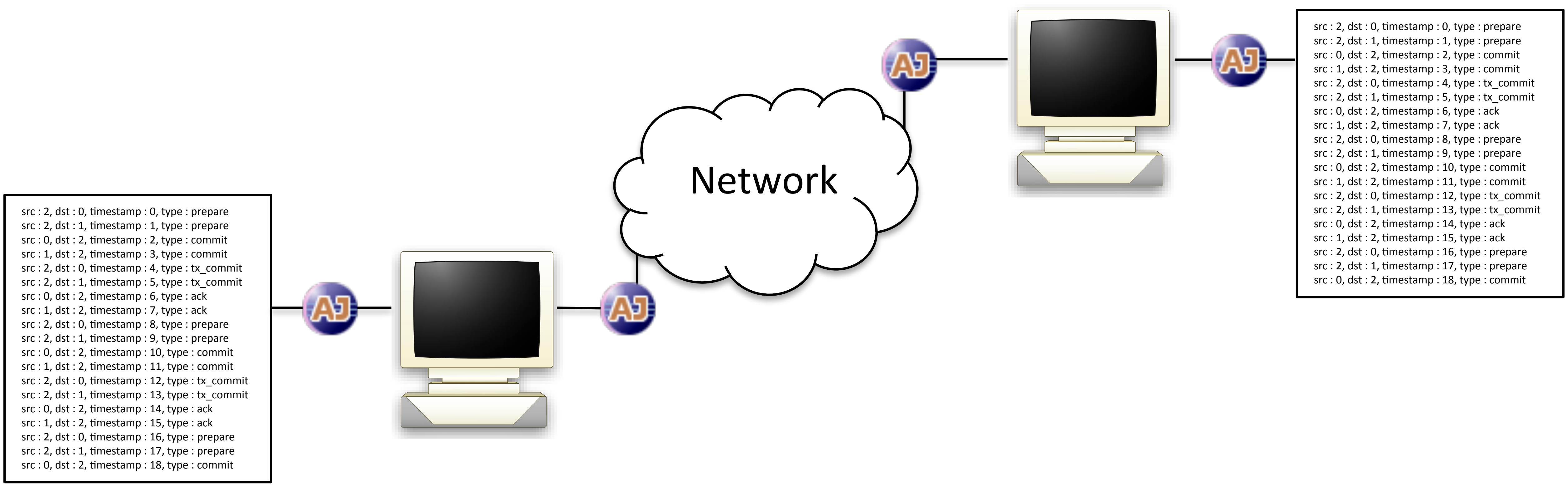


[1,0]	a
[0,1]	b



Vector clock instrumentation

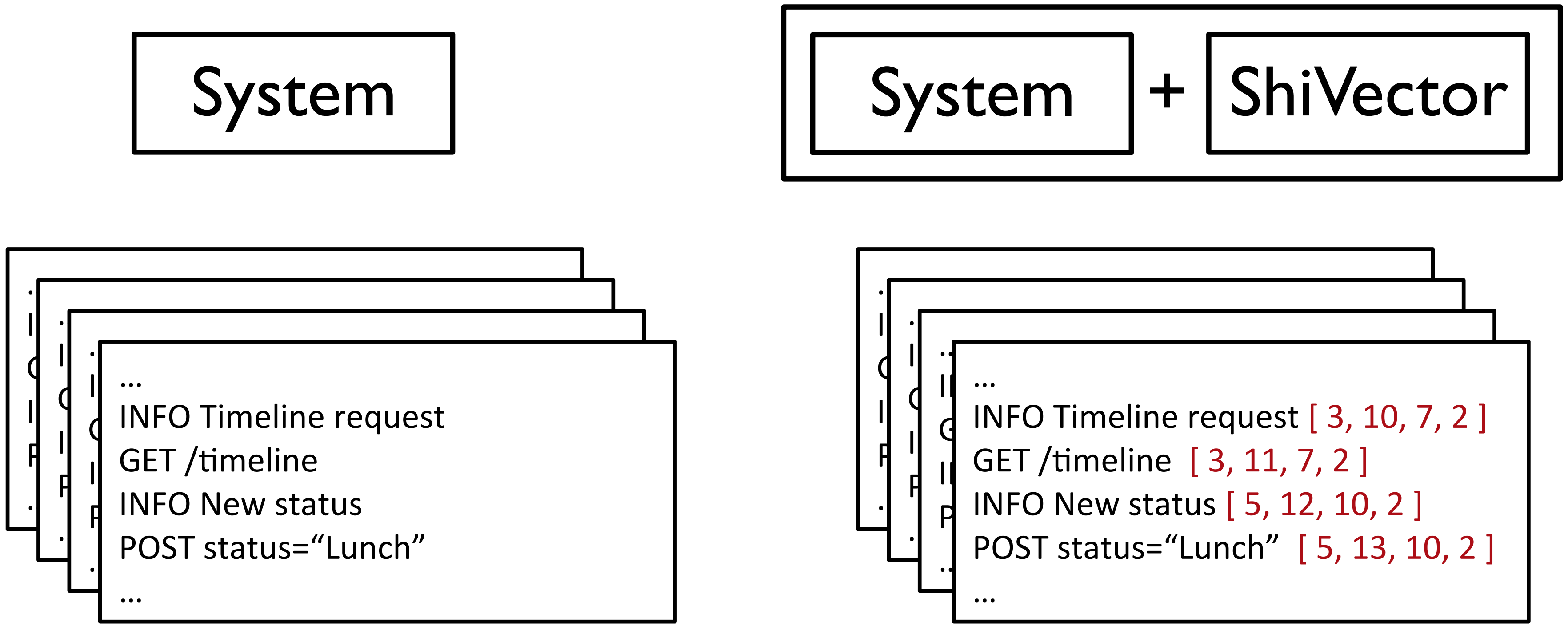
- Automatically instruments a distributed system logging with vector clock information
- Transparently interposes on all socket communication and logging (via AspectJ)



https://bitbucket.org/bestchai/shivector

Vector clock instrumentation

- Automatically instruments a distributed system logging with vector clock information
- Transparently interposes on all socket communication and logging (via AspectJ)



https://bitbucket.org/bestchai/shivector

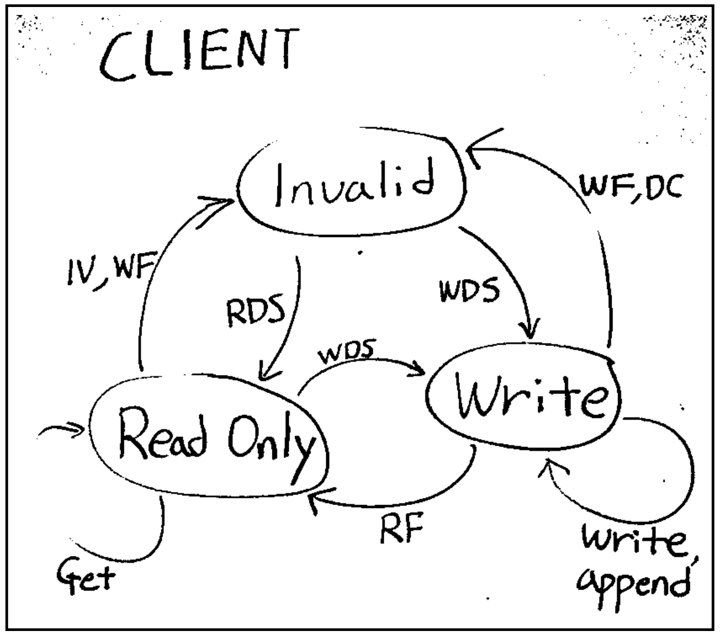
CSight talk outline

- Motivation
- Background
  - Logging: partial order and vector clocks
  - Modeling: communicating FSMs
- CSight approach
- Evaluation

http://synoptic.googlecode.com

Extending finite state machines

Sequential:

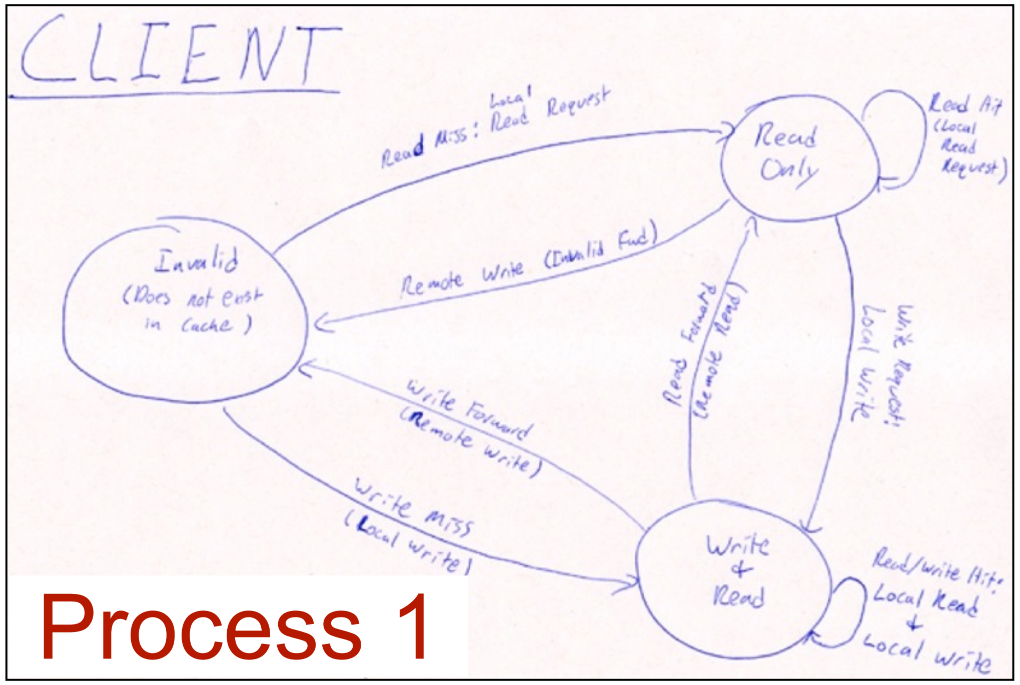


Concurrent:

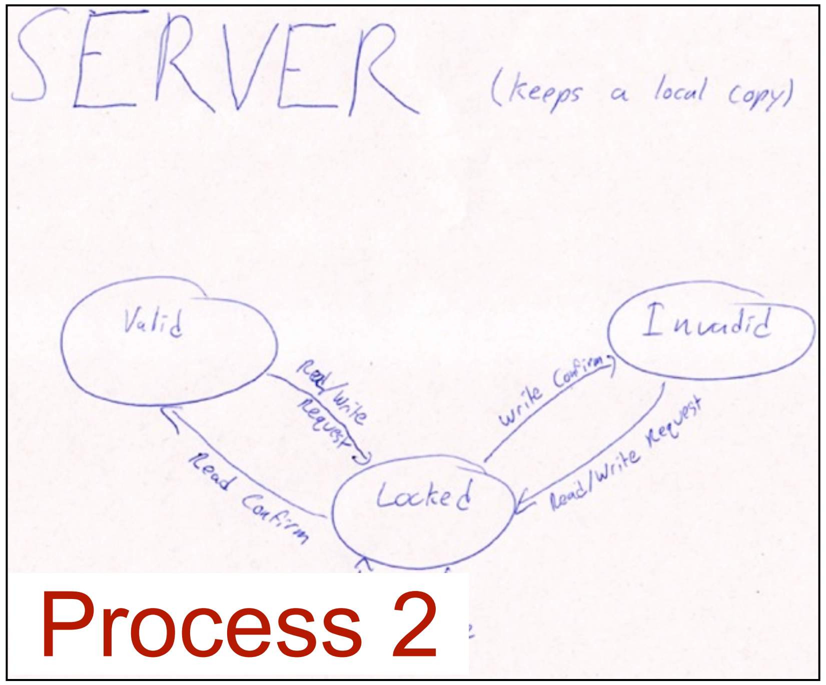
Extending finite state machines

Sequential:

Concurrent:

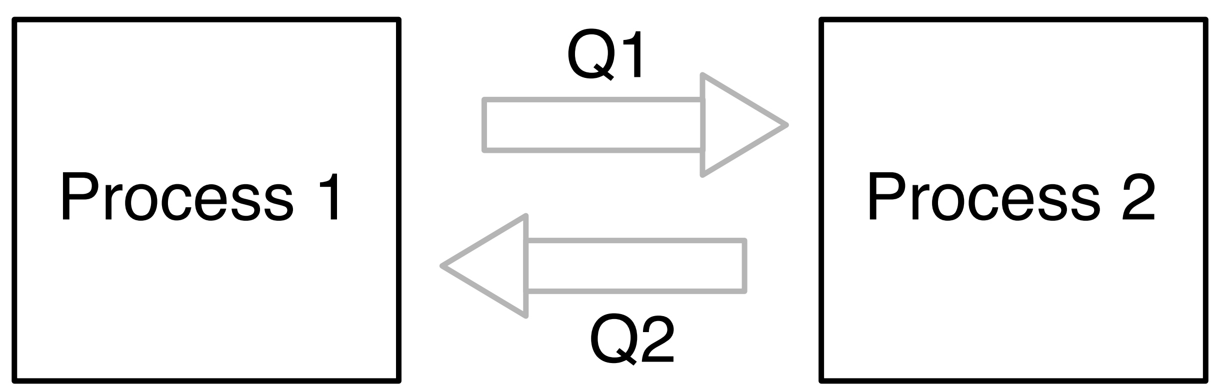


Process 1

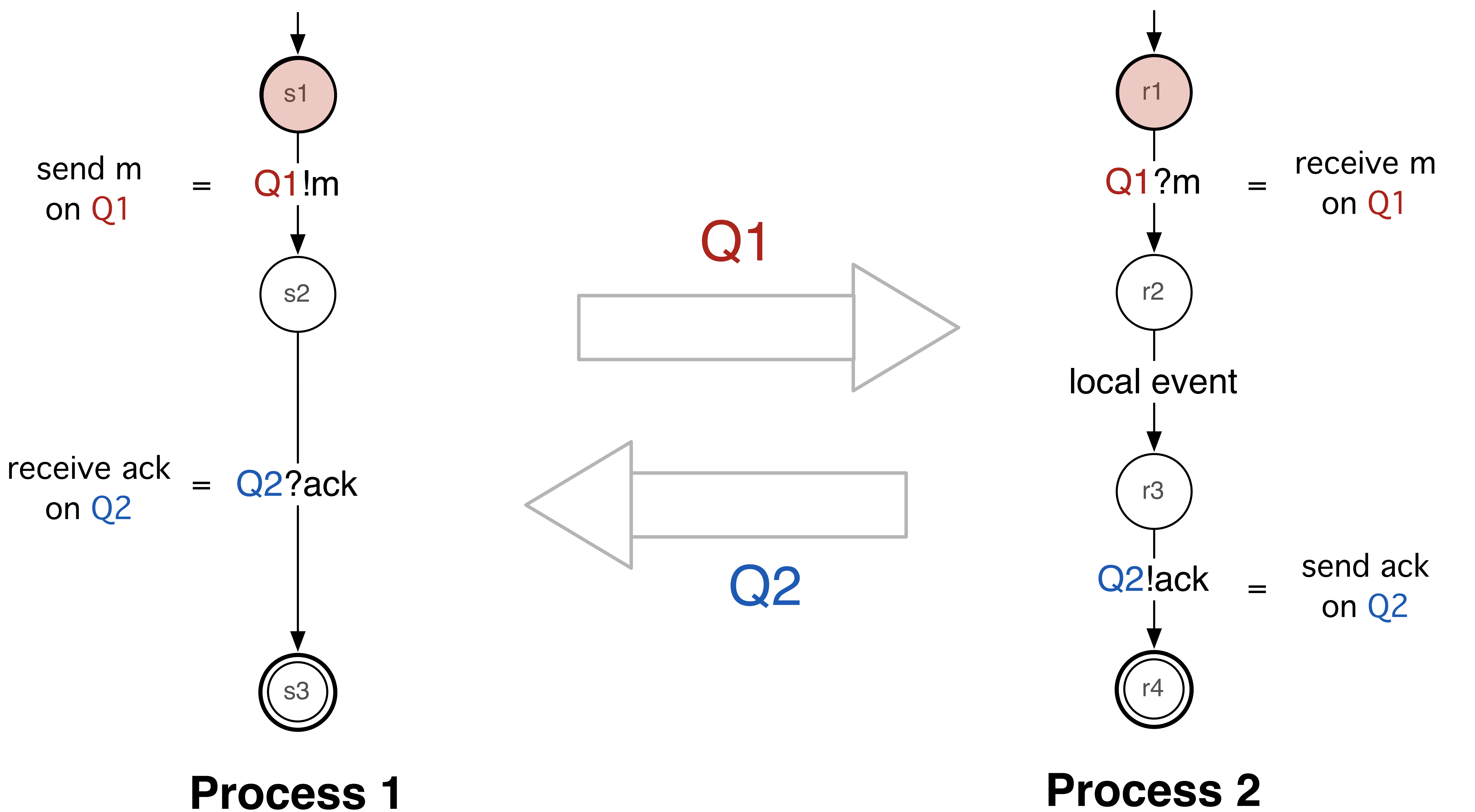


Process 2

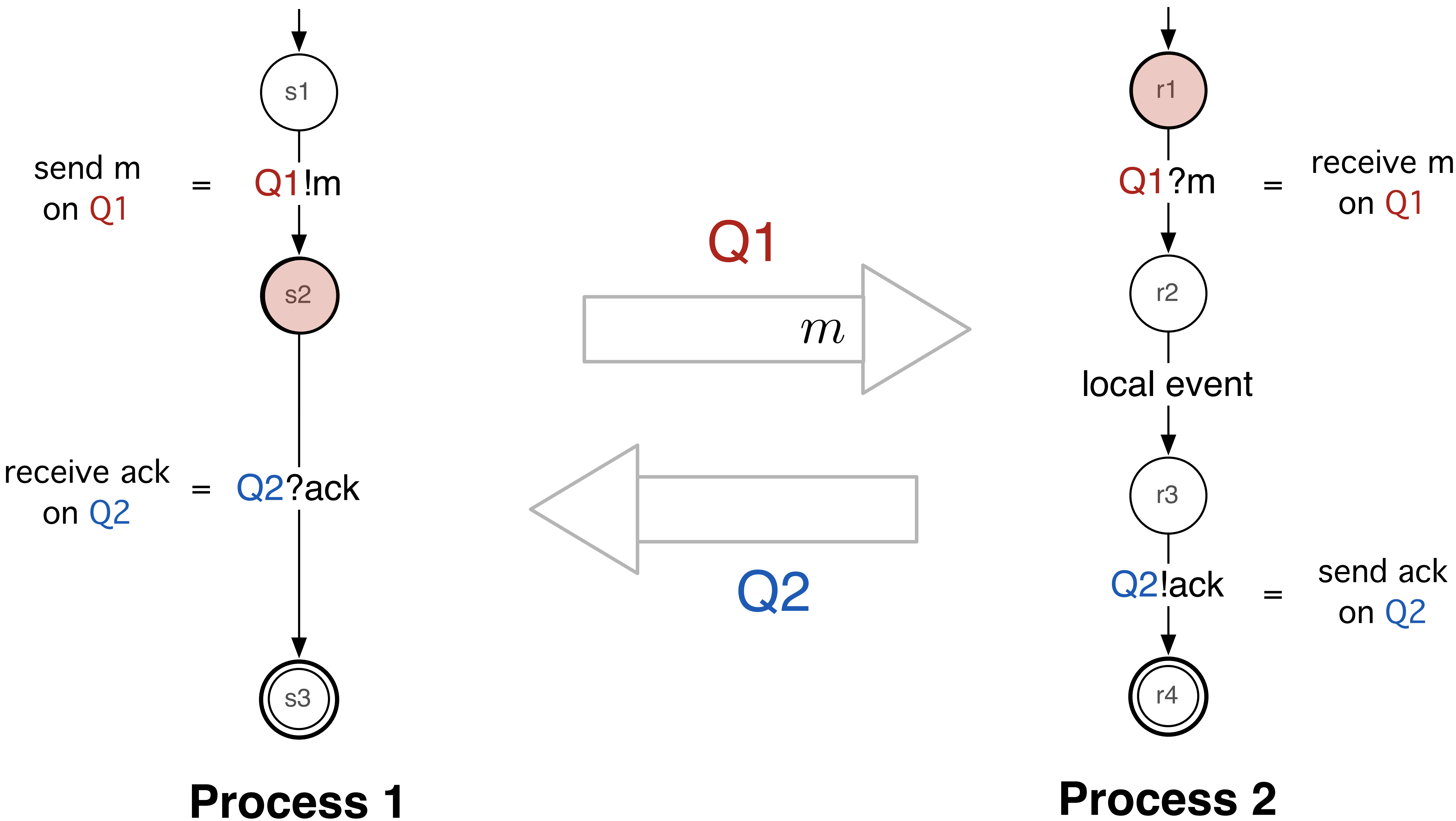
Communicating FSM formalism



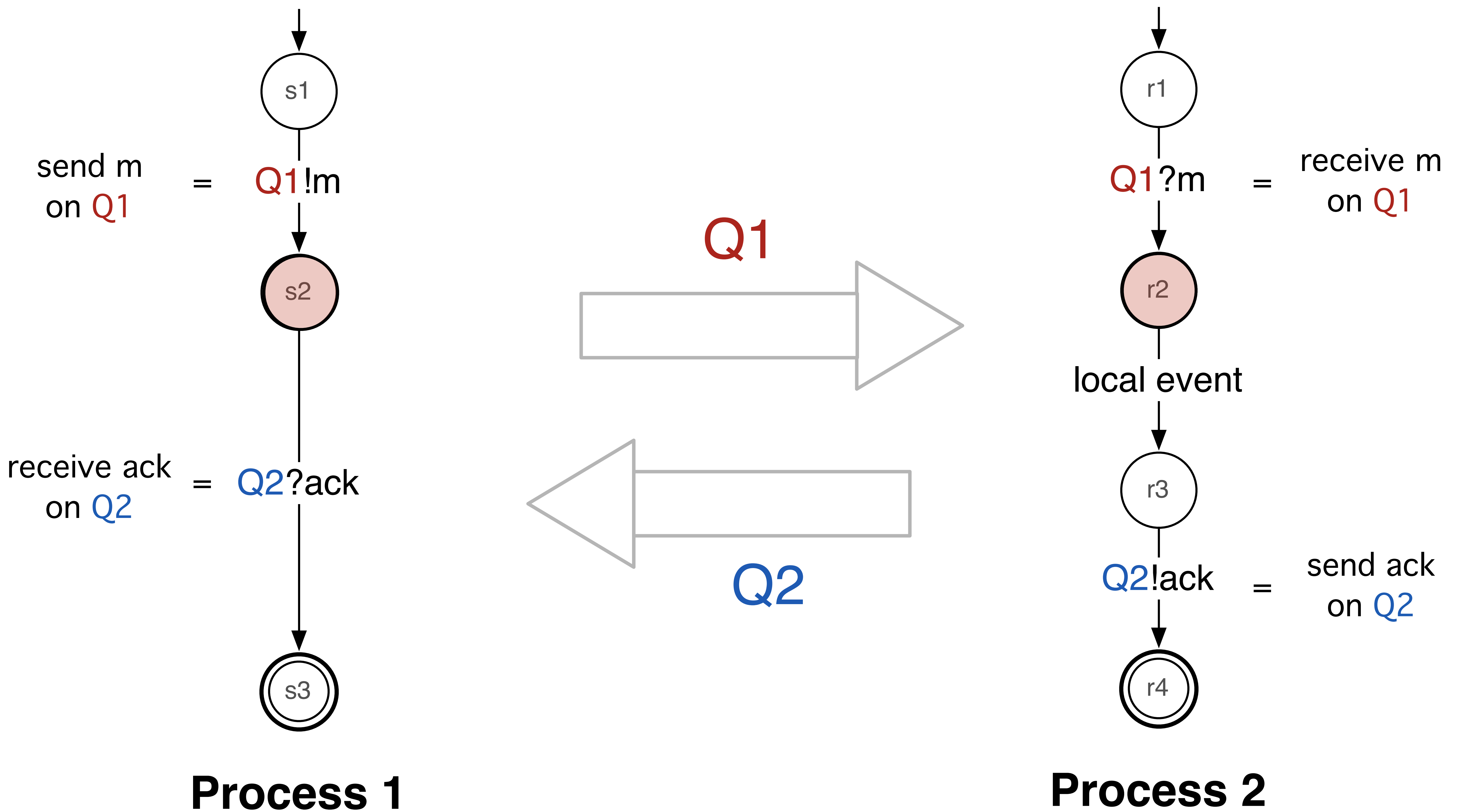
Communicating FSM formalism



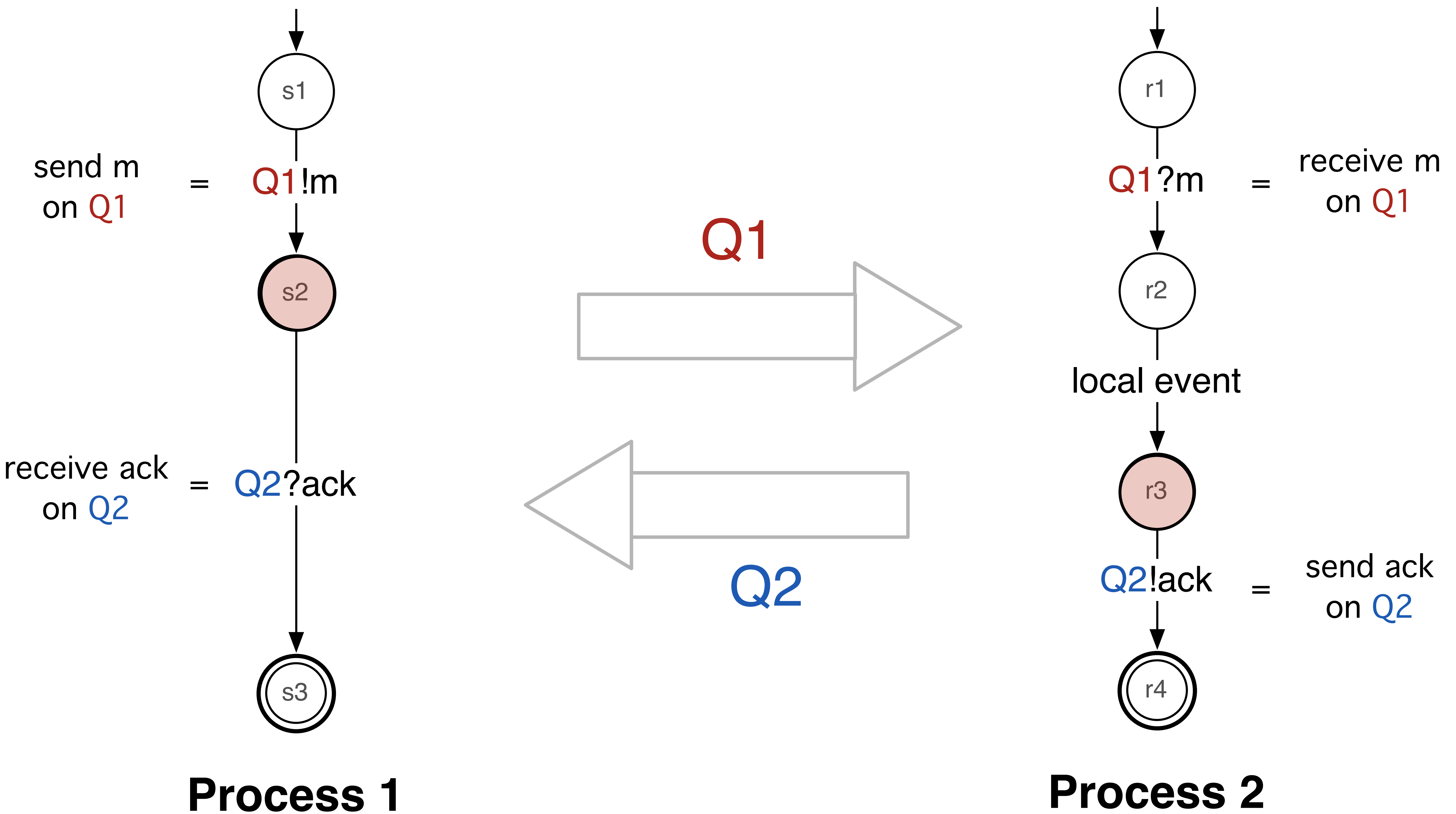
Communicating FSM formalism



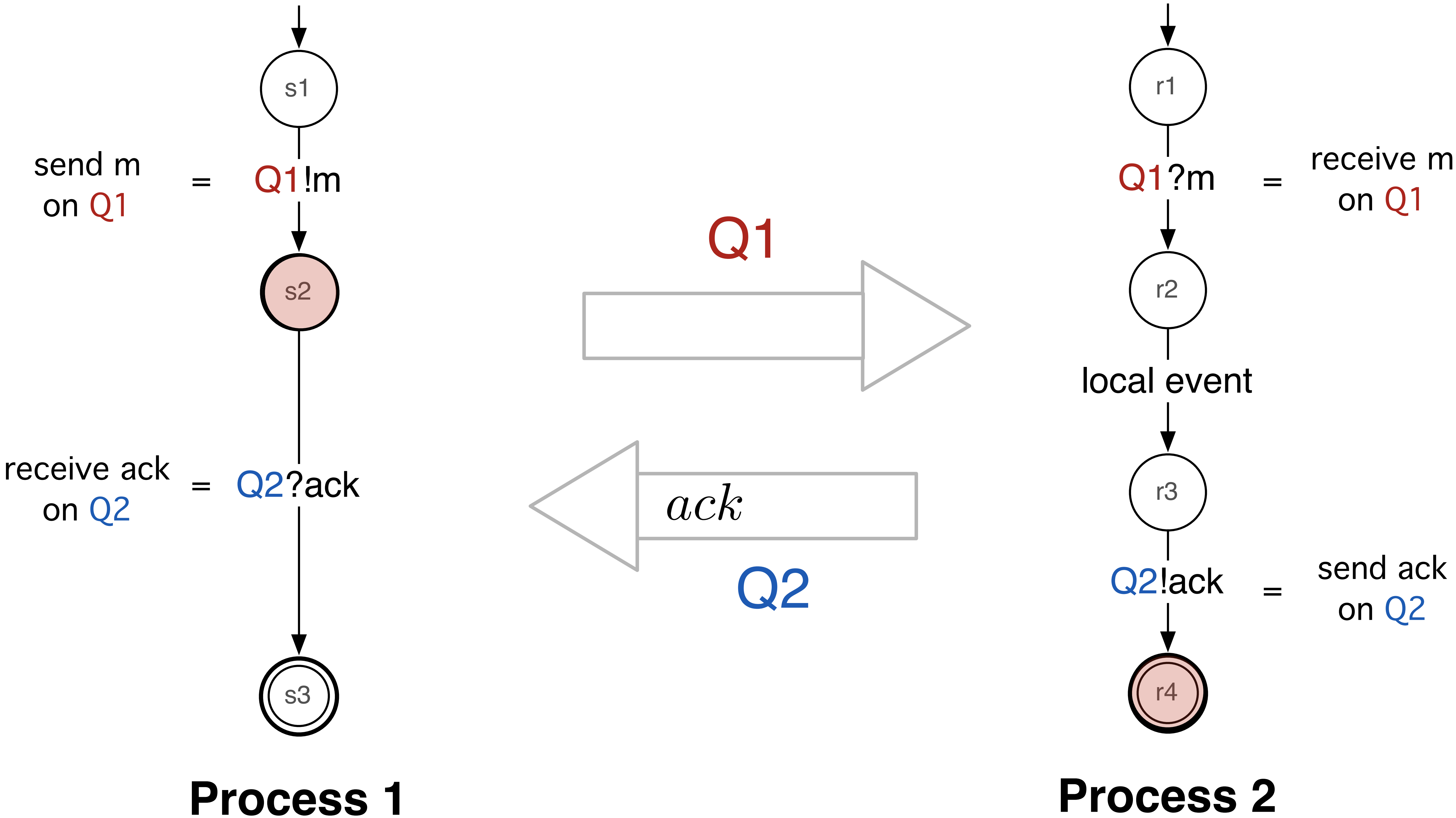
Communicating FSM formalism



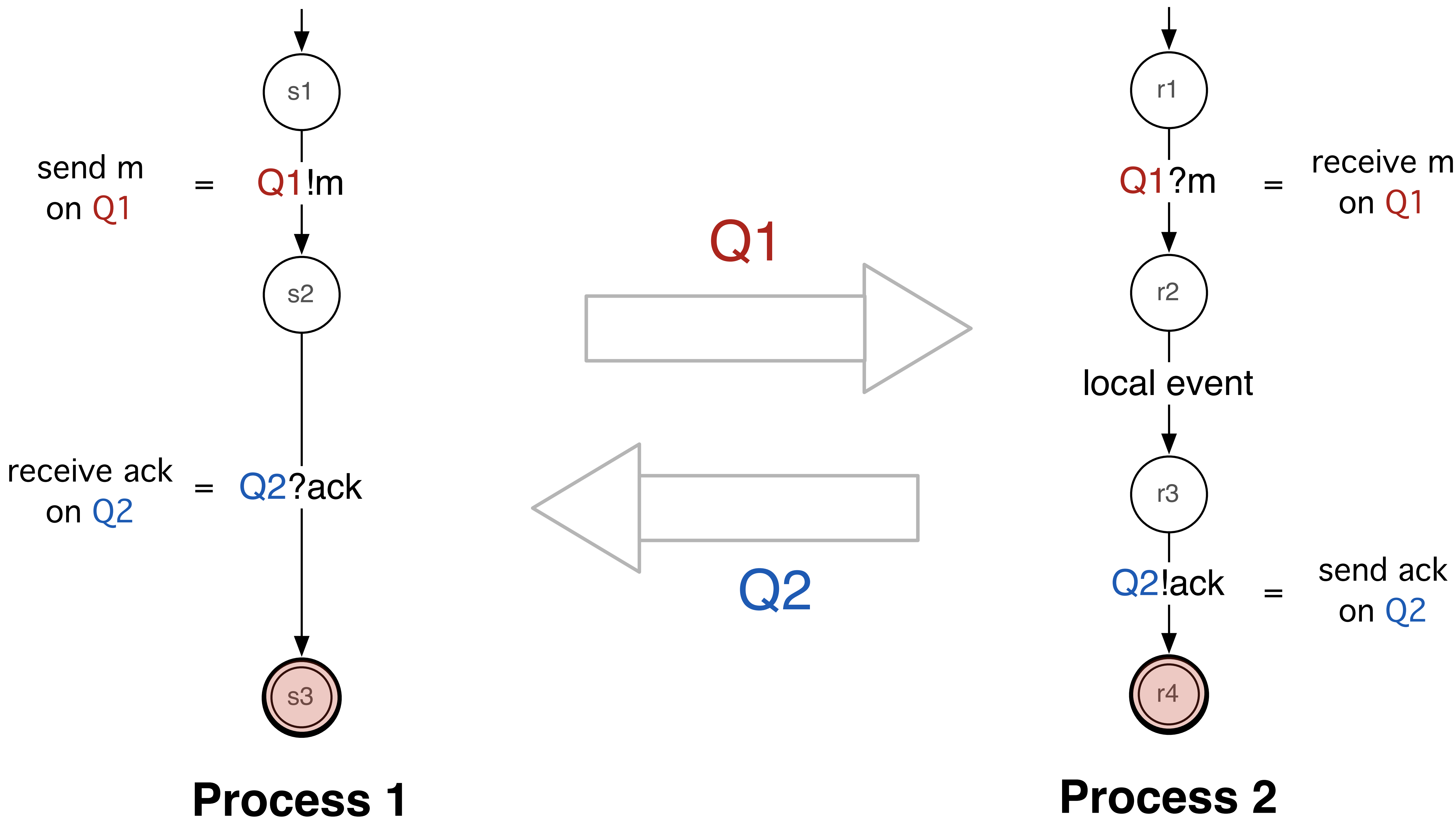
Communicating FSM formalism



Communicating FSM formalism



Communicating FSM formalism



CSight talk outline

- Motivation
- Background
  - Logging: partial order and vector clocks
  - Modeling: communicating FSMs
- CSight approach
- Evaluation

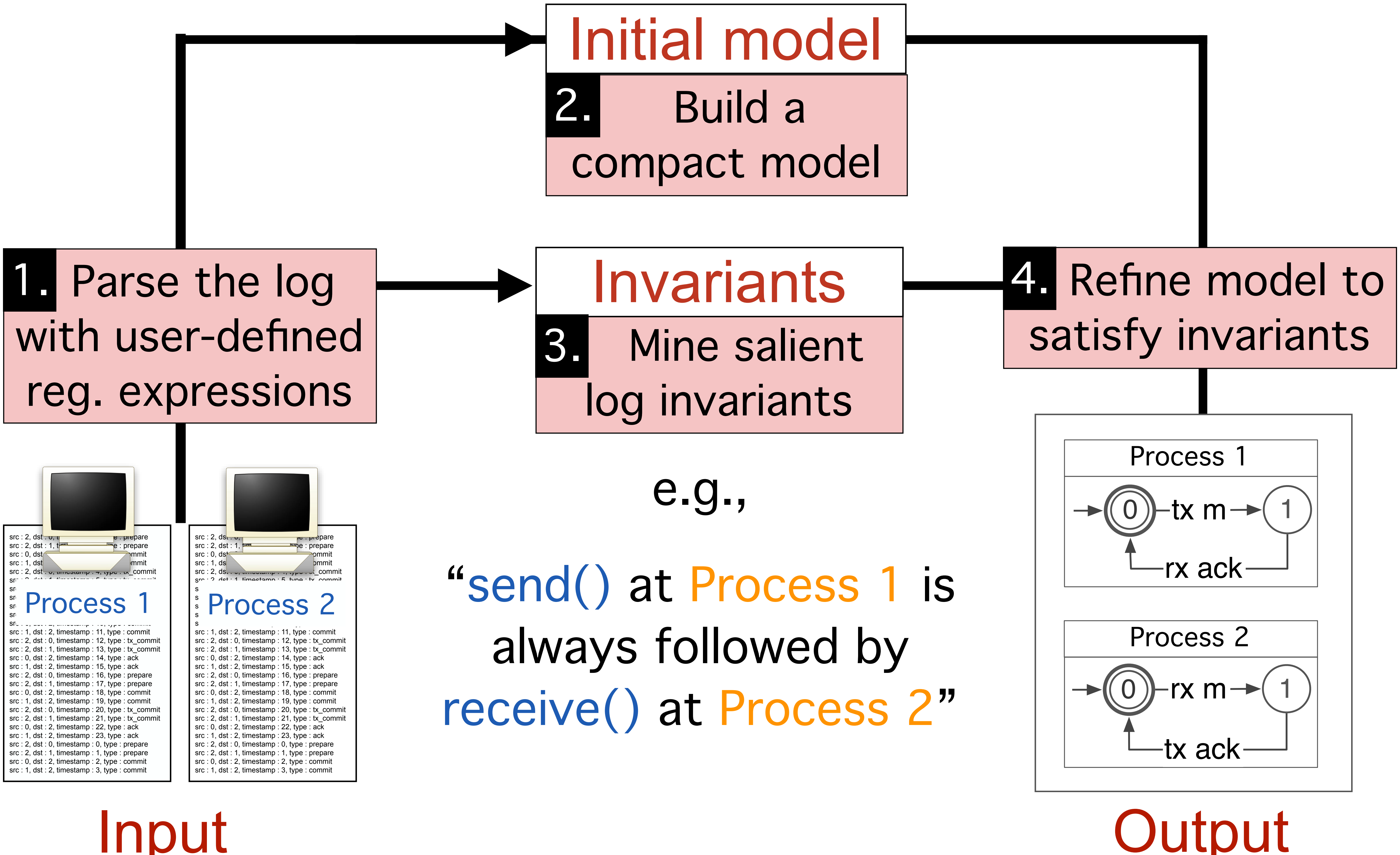
http://synoptic.googlecode.com

Goal

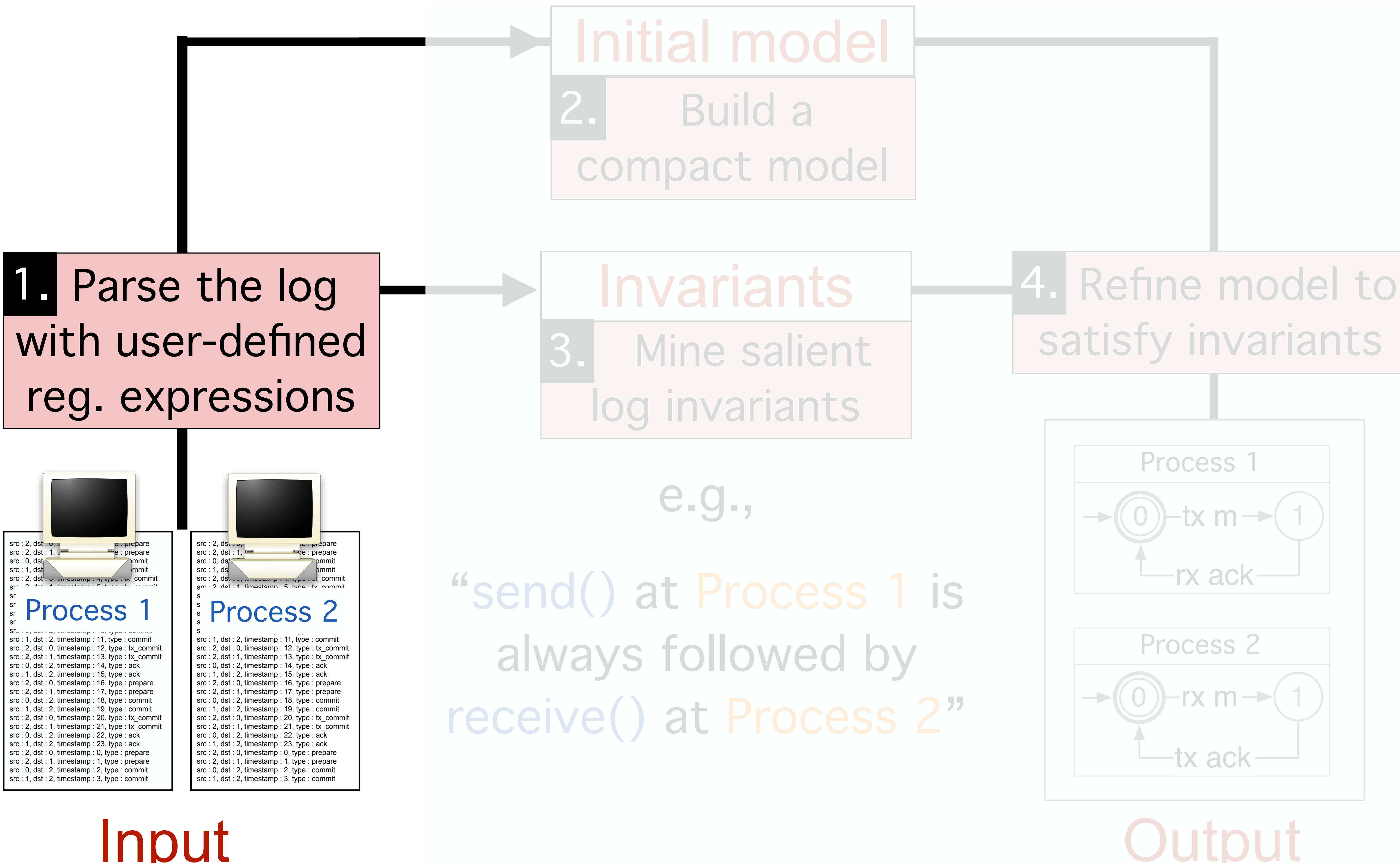
Input

Output

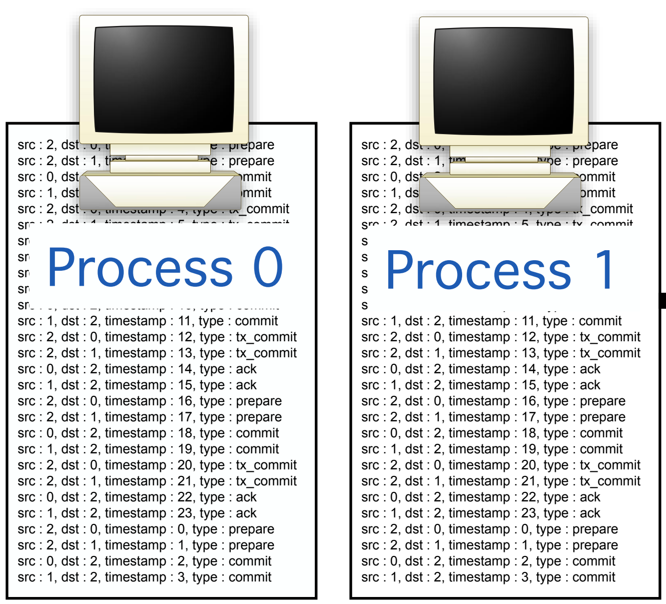
CSight approach



CSight approach



From logs to execution DAGs



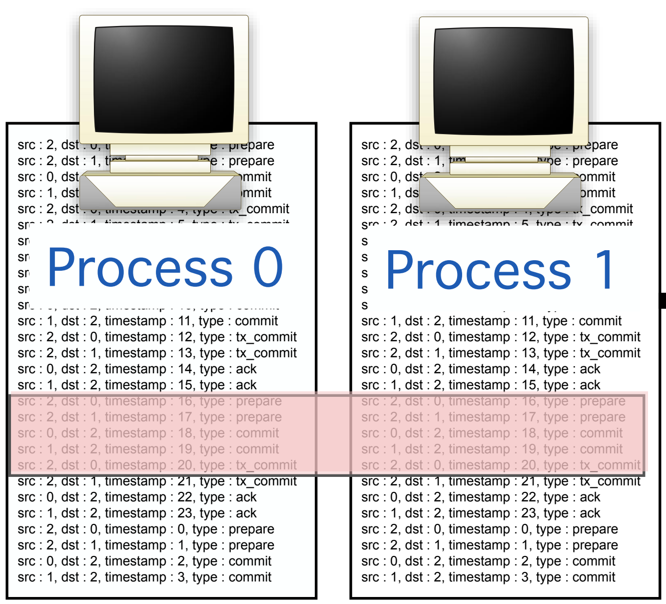
regular
expressions

Input

Time-space
DAGs

Fidge, ACSS 1988	Mattern, PDA 1989

From logs to execution DAGs



regular

expressions

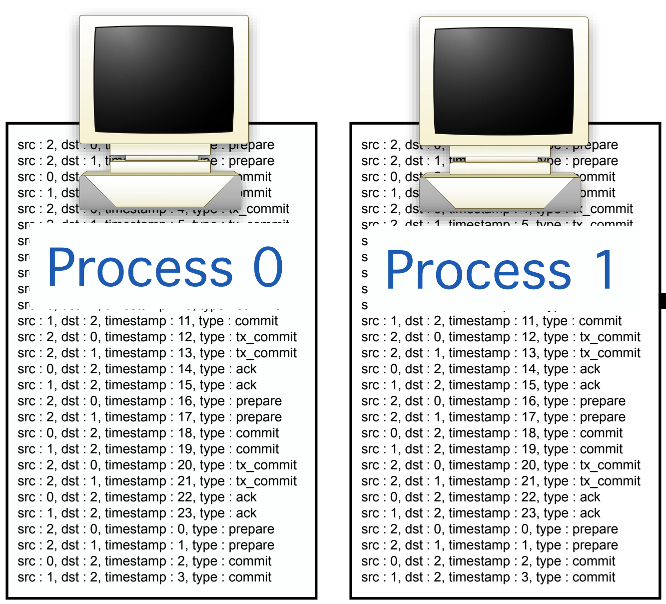
Input

Time-space DAGs

Fidge, ACSS 1988	Mattern, PDA 1989

From logs to execution DAGs

[1,0] Q1!m
[1,1] Q1?m
[1,2] local event
[1,3] Q2!ack
[2,3] Q2?ack



Input

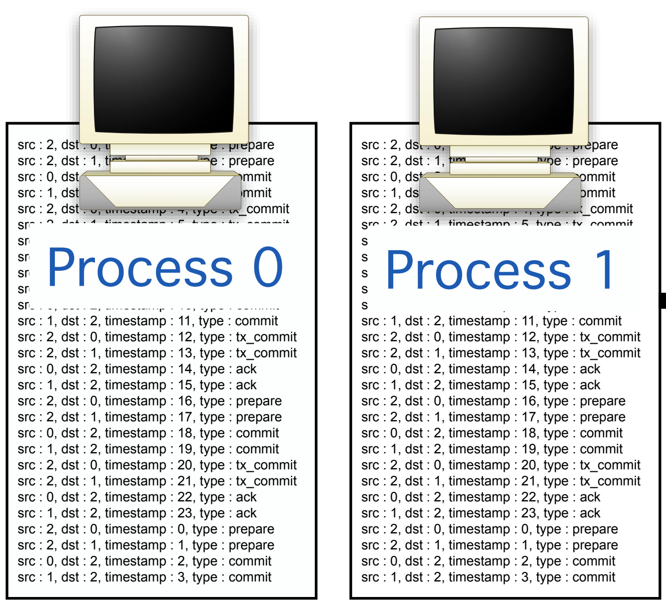
regular expressions

Time-space DAGs

Fidge, ACSS 1988	Mattern, PDA 1989

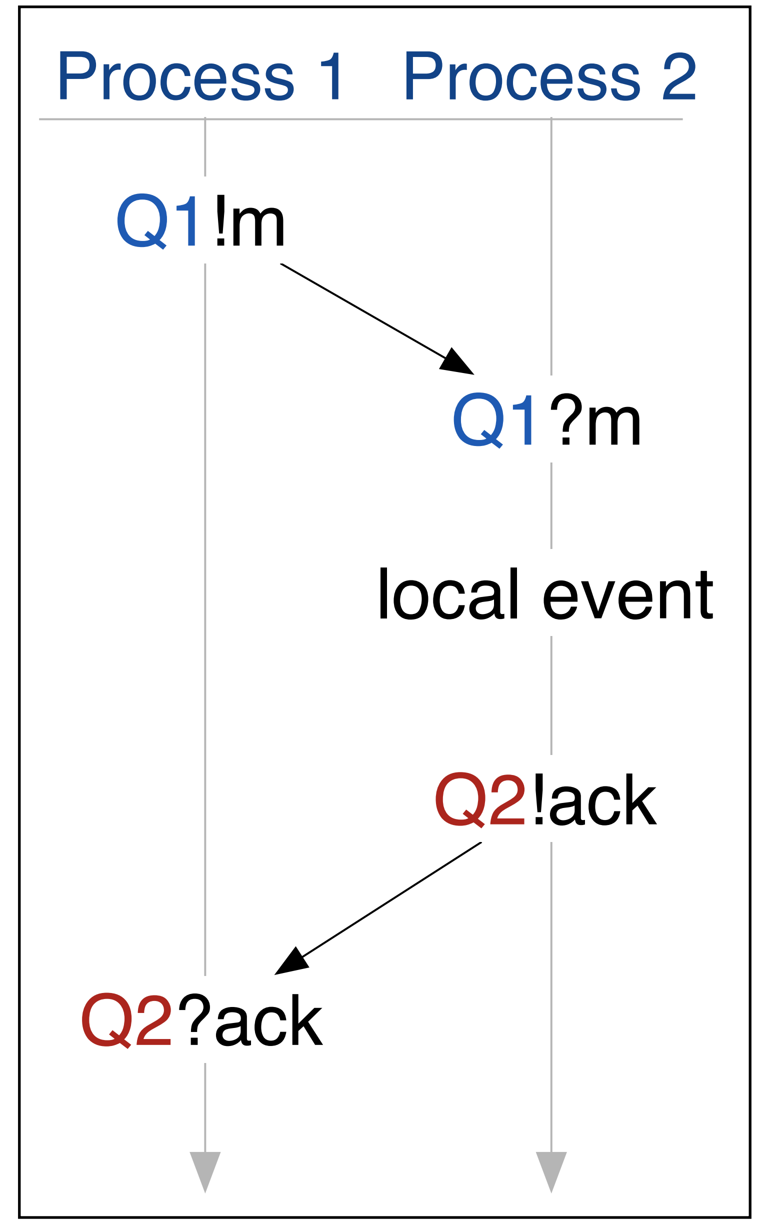
From logs to execution DAGs

[1,0] Q1!m
[1,1] Q1?m
[1,2] local event
[1,3] Q2!ack
[2,3] Q2?ack



Input

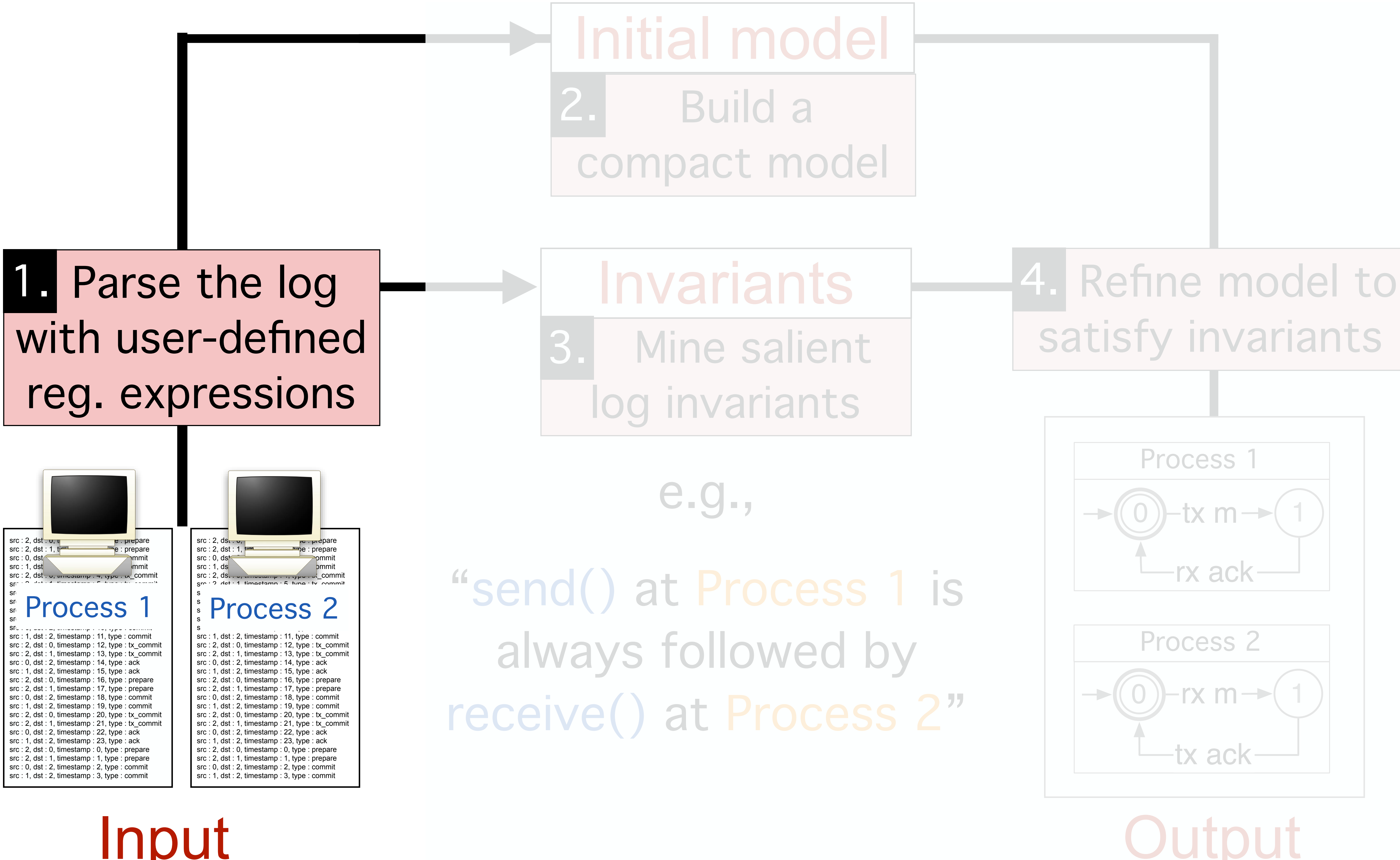
regular expressions



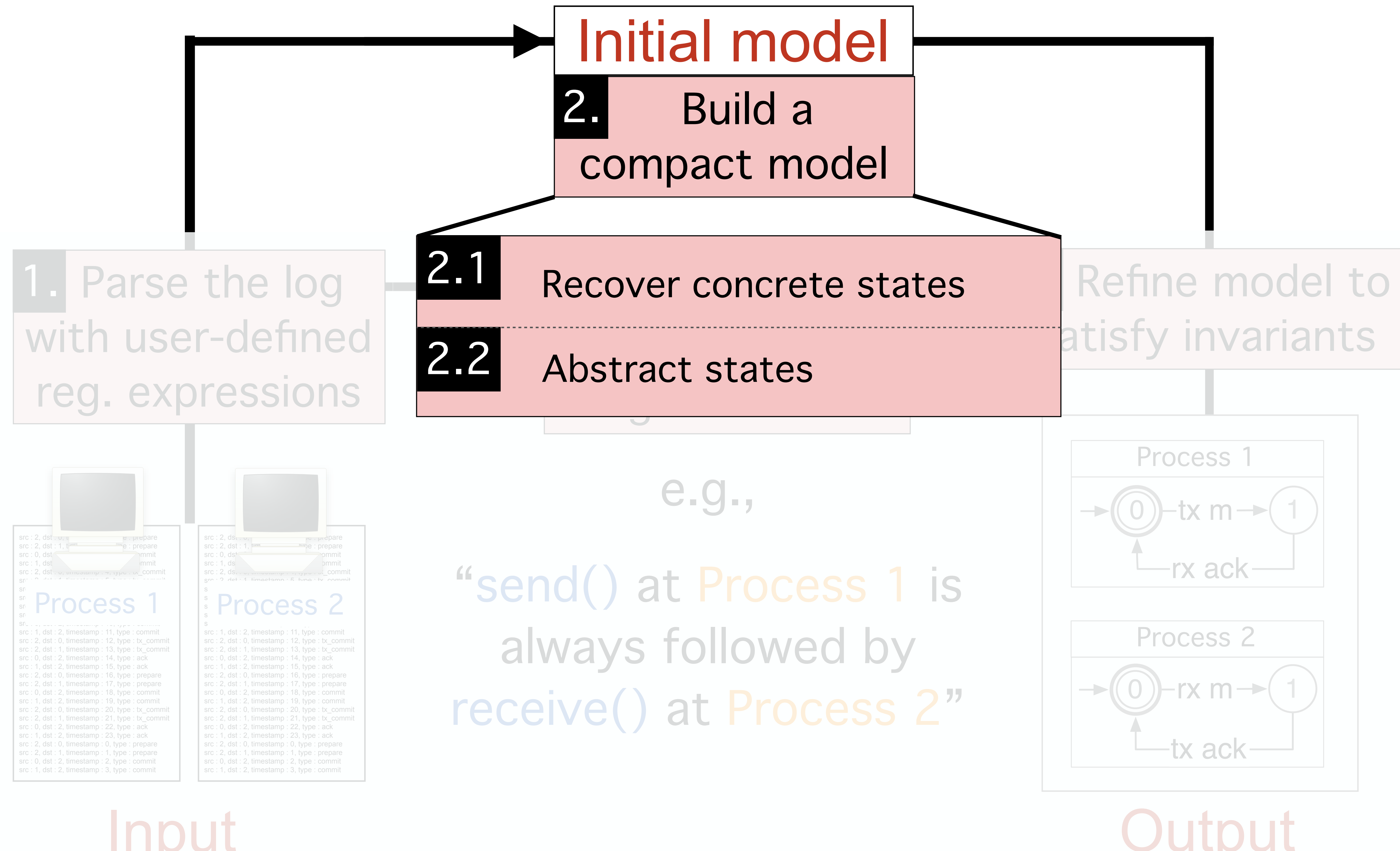
Time-space DAGs

Fidge, ACSS 1988	Mattern, PDA 1989

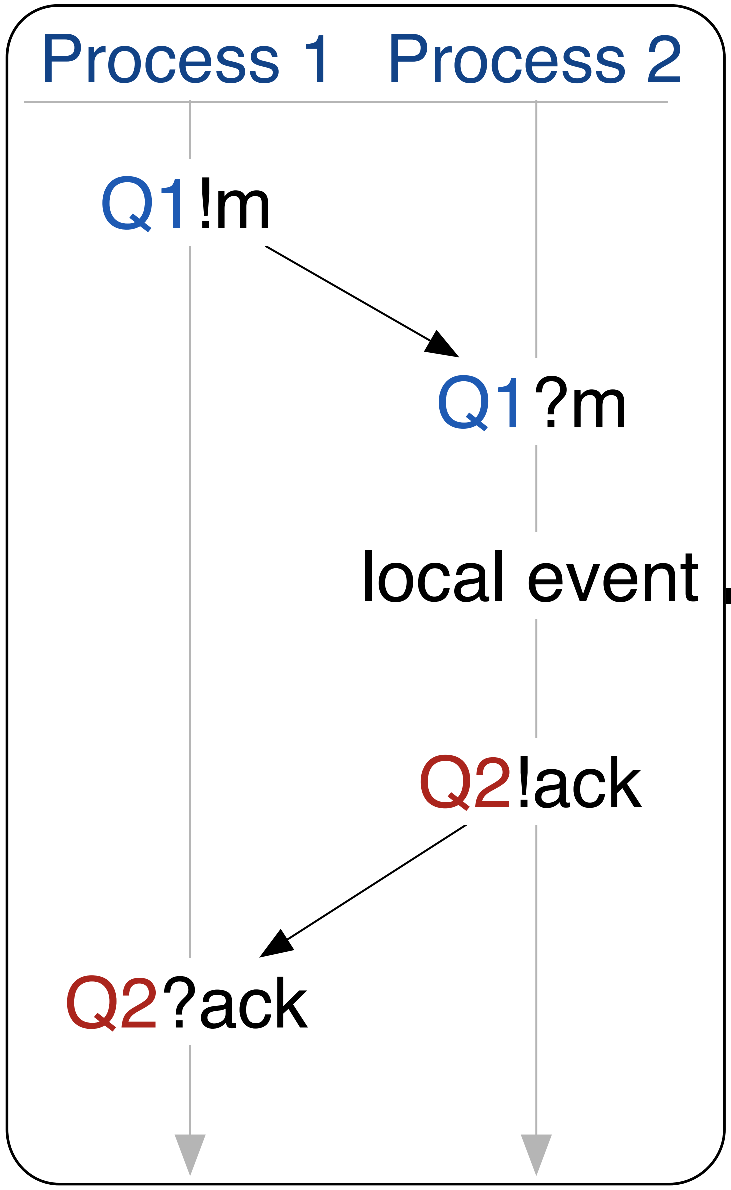
CSight approach



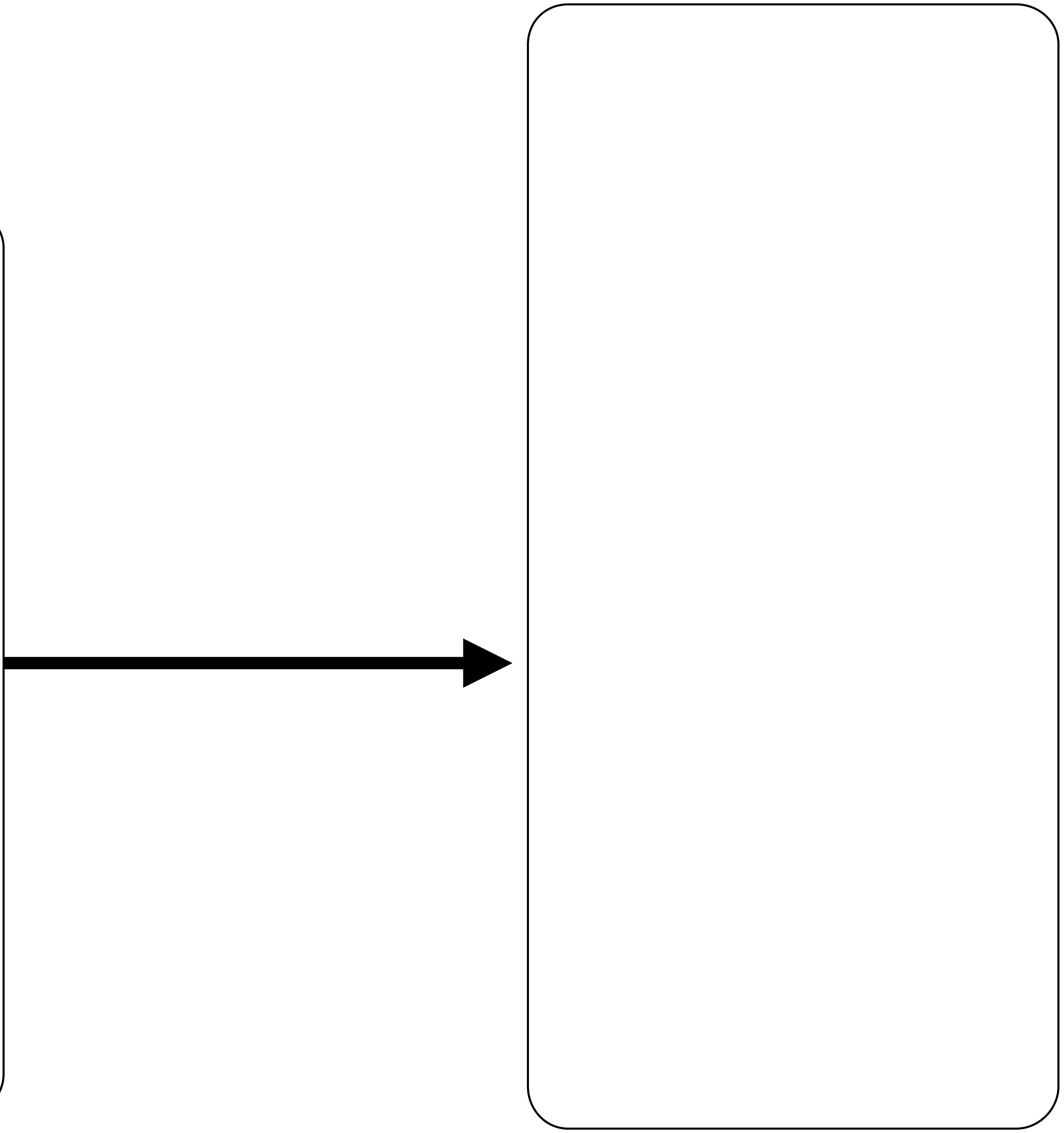
CSight approach



Recover concrete queue states

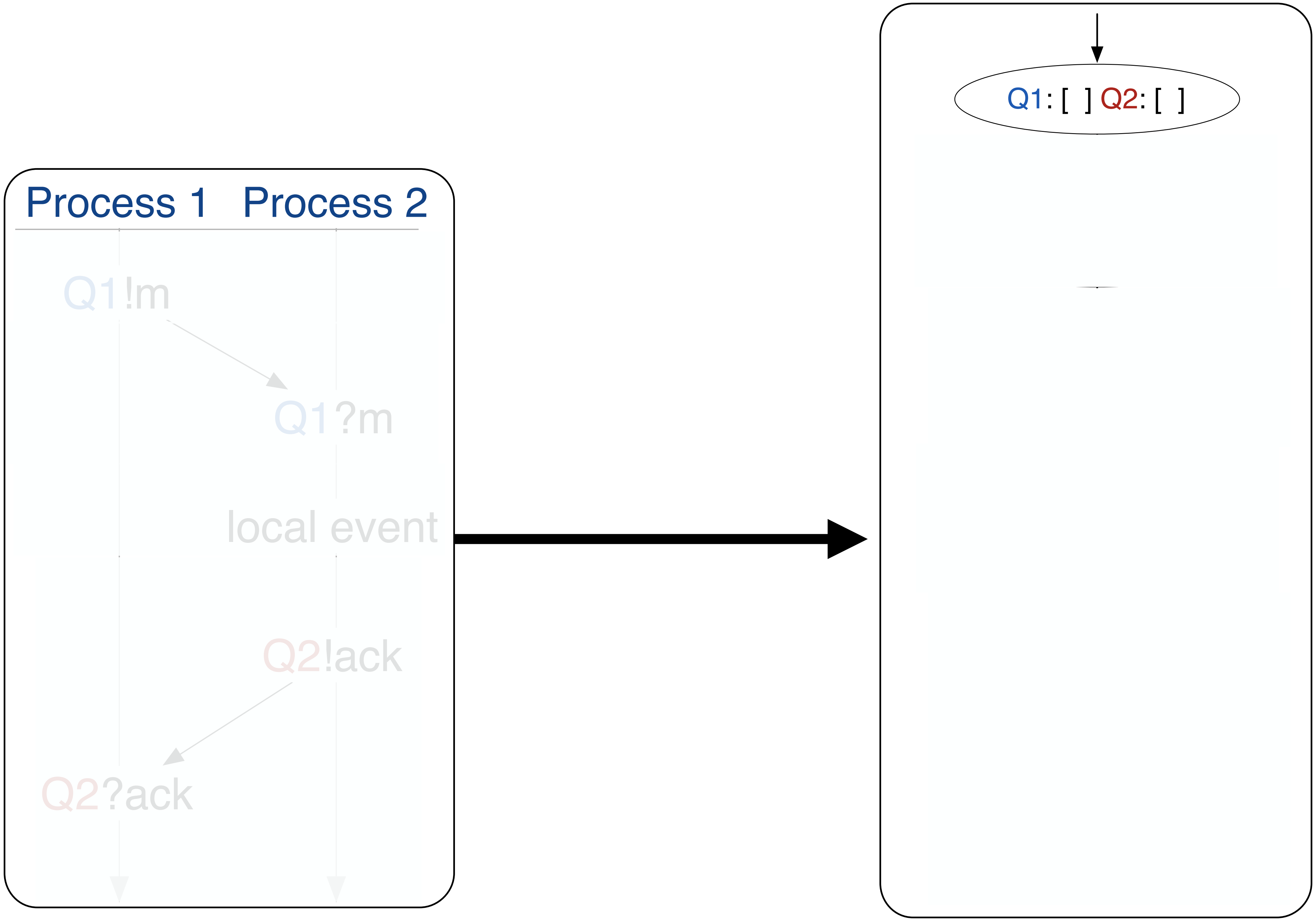


Time-Space DAG



State-based DAG

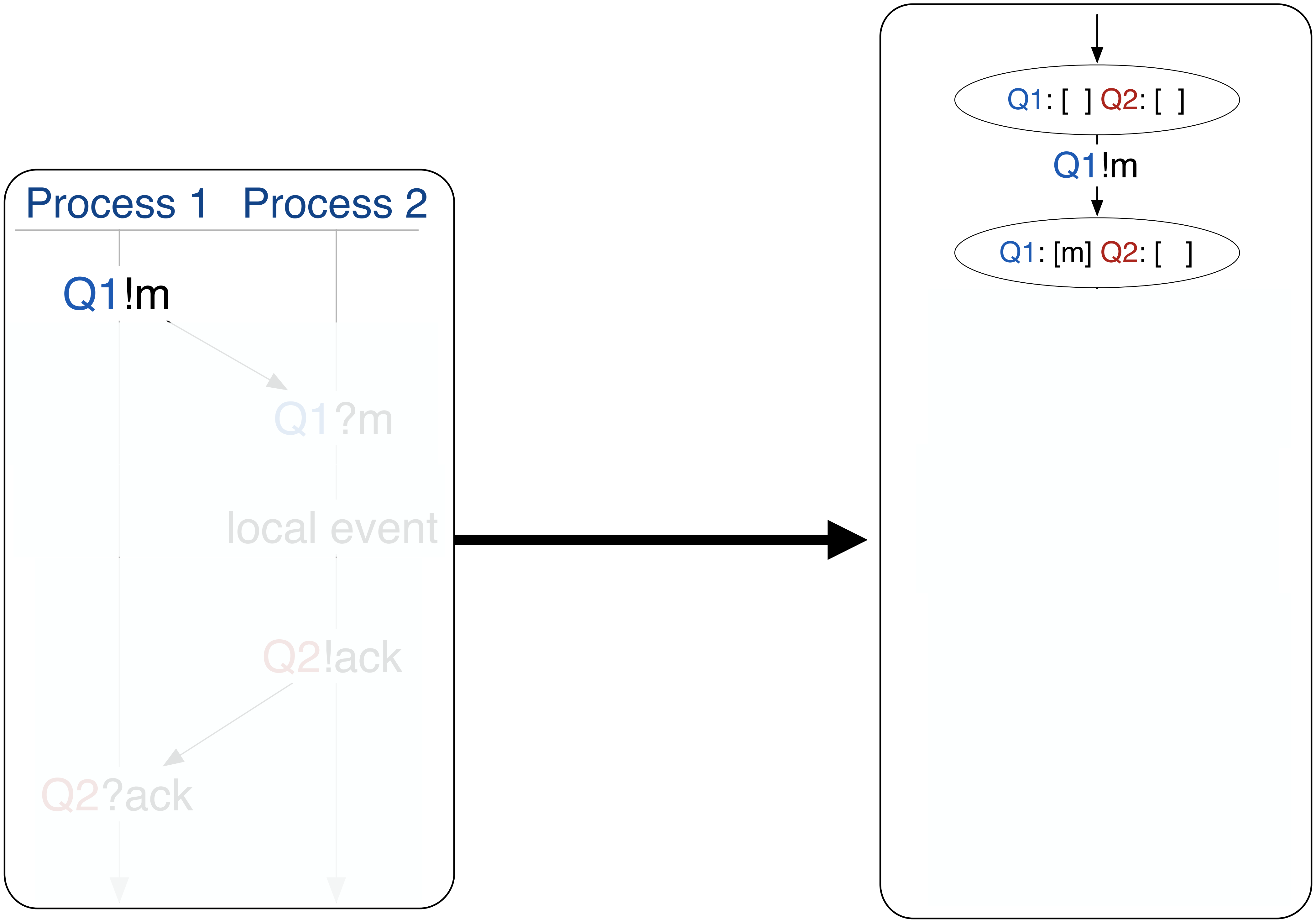
Recover concrete queue states



Time-Space DAG

State-based DAG

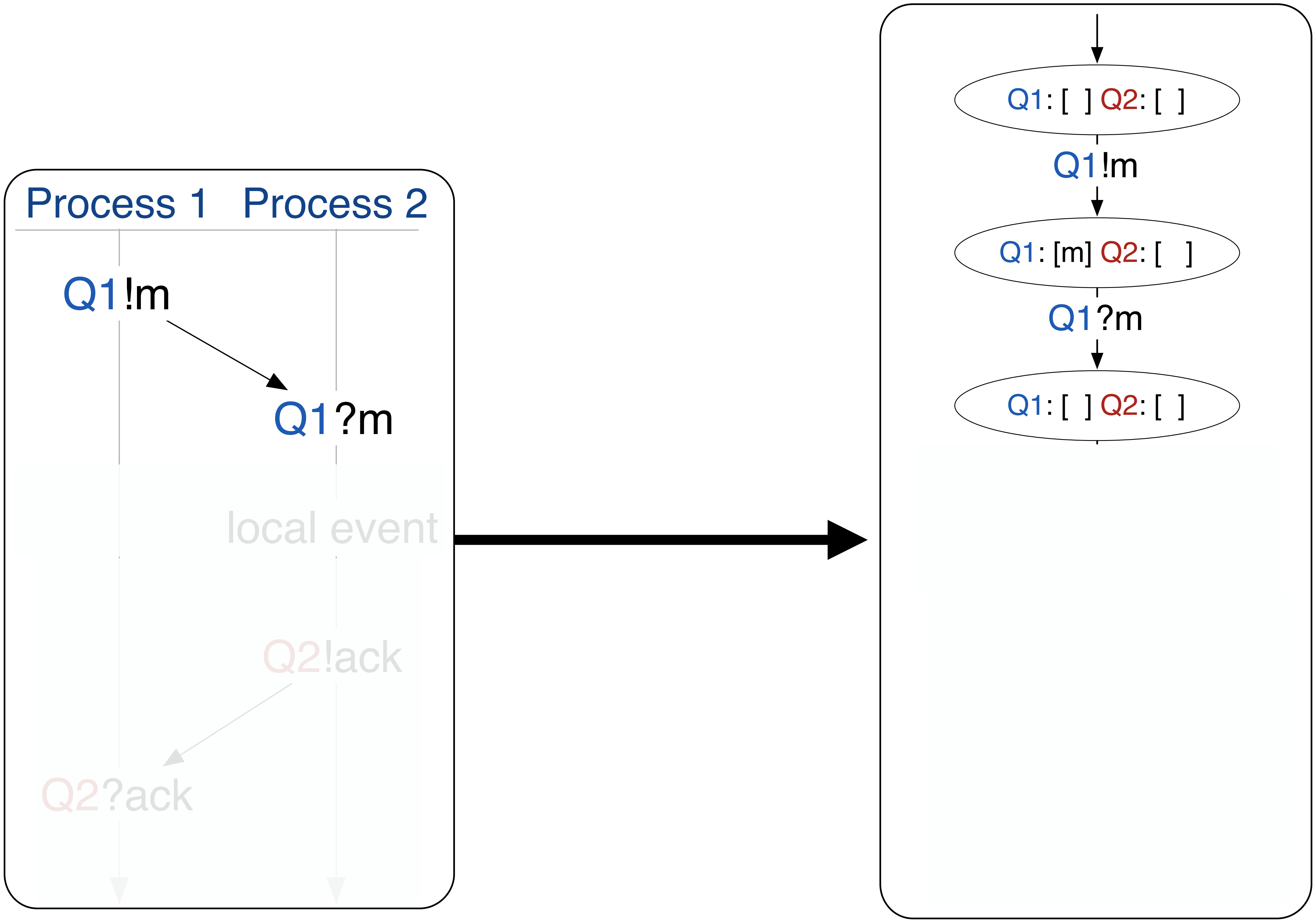
Recover concrete queue states



Time-Space DAG

State-based DAG

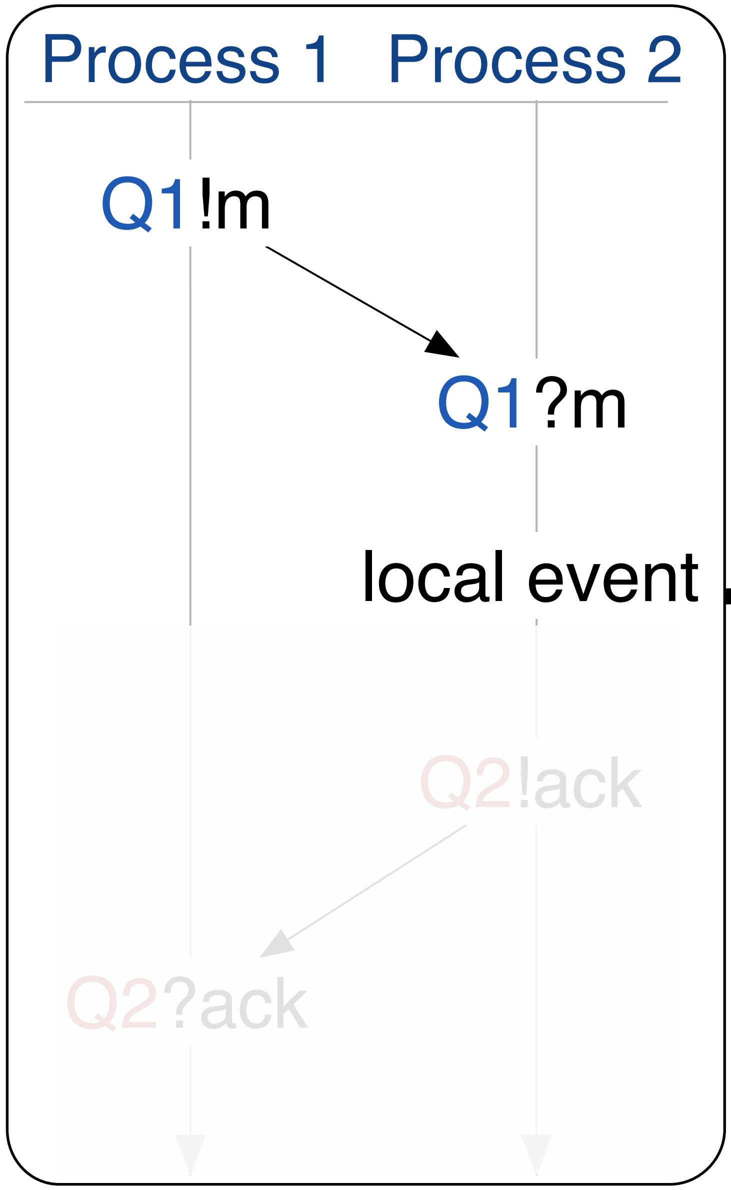
Recover concrete queue states



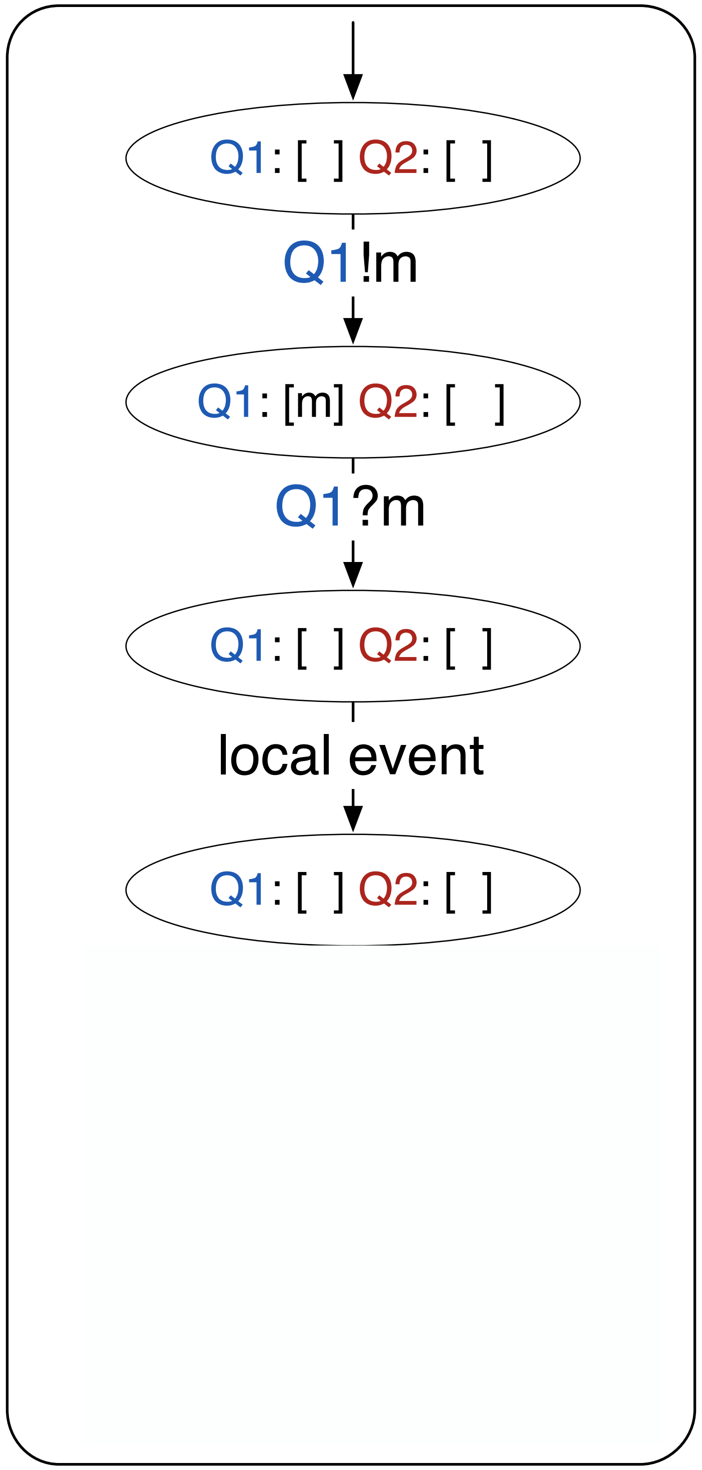
Time-Space DAG

State-based DAG

Recover concrete queue states

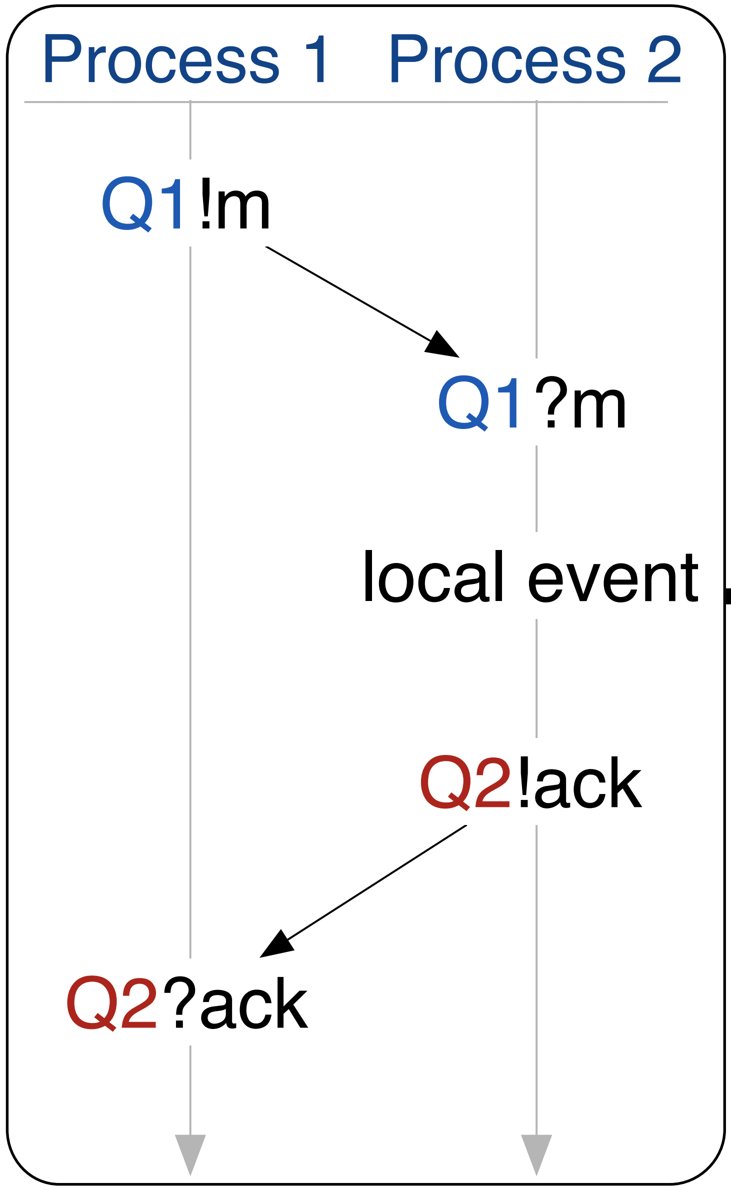


Time-Space DAG

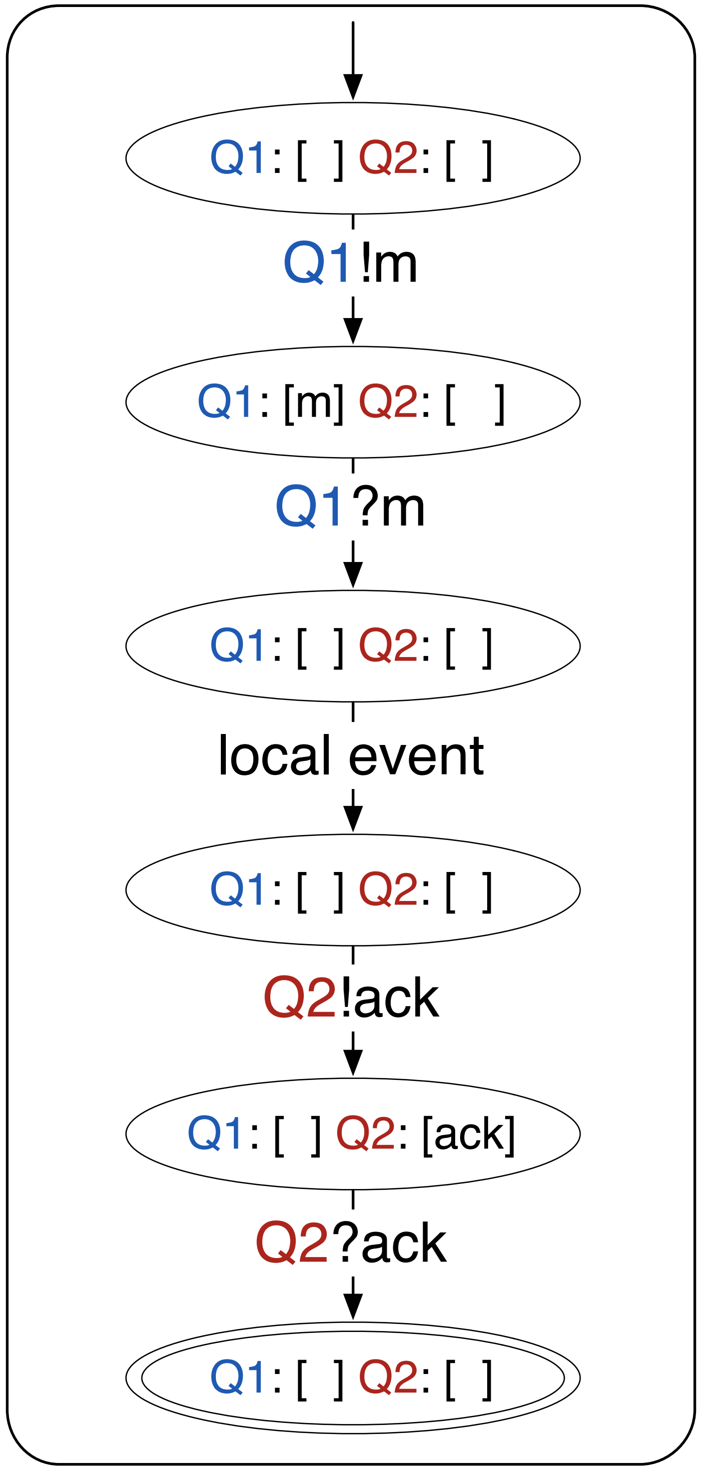


State-based DAG

Recover concrete queue states

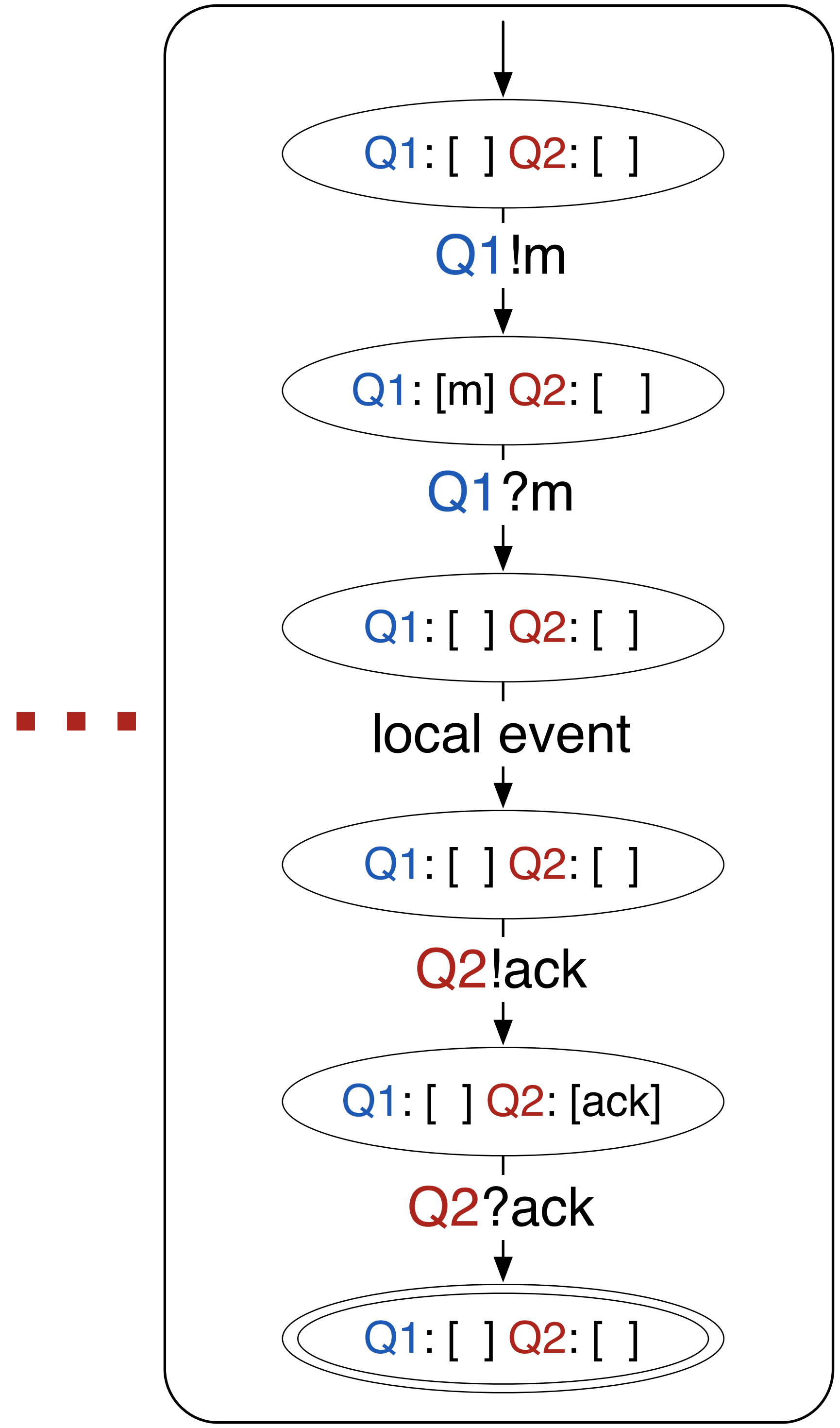


Time-Space DAG



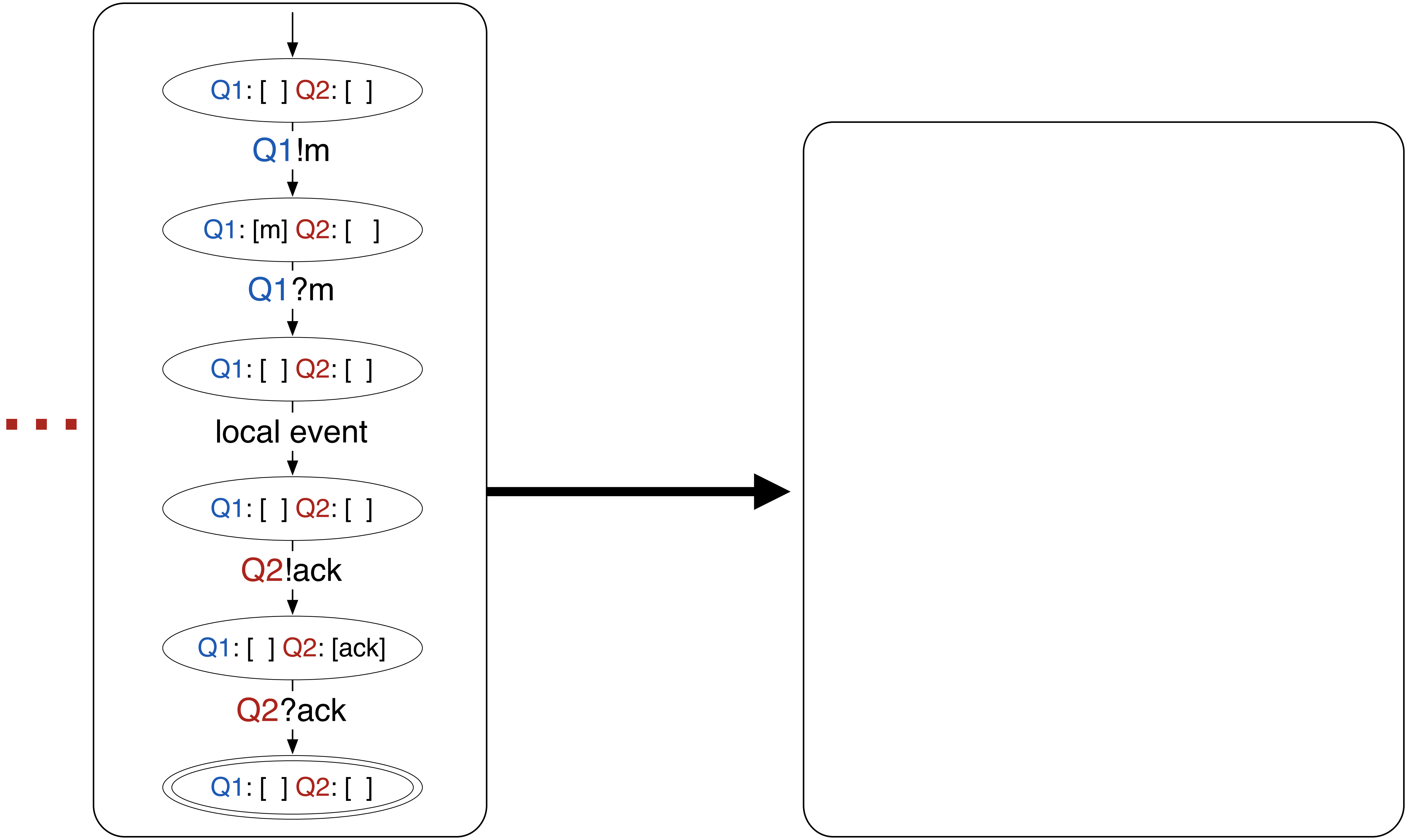
State-based DAG

Abstract concrete queue states



State-based DAGs

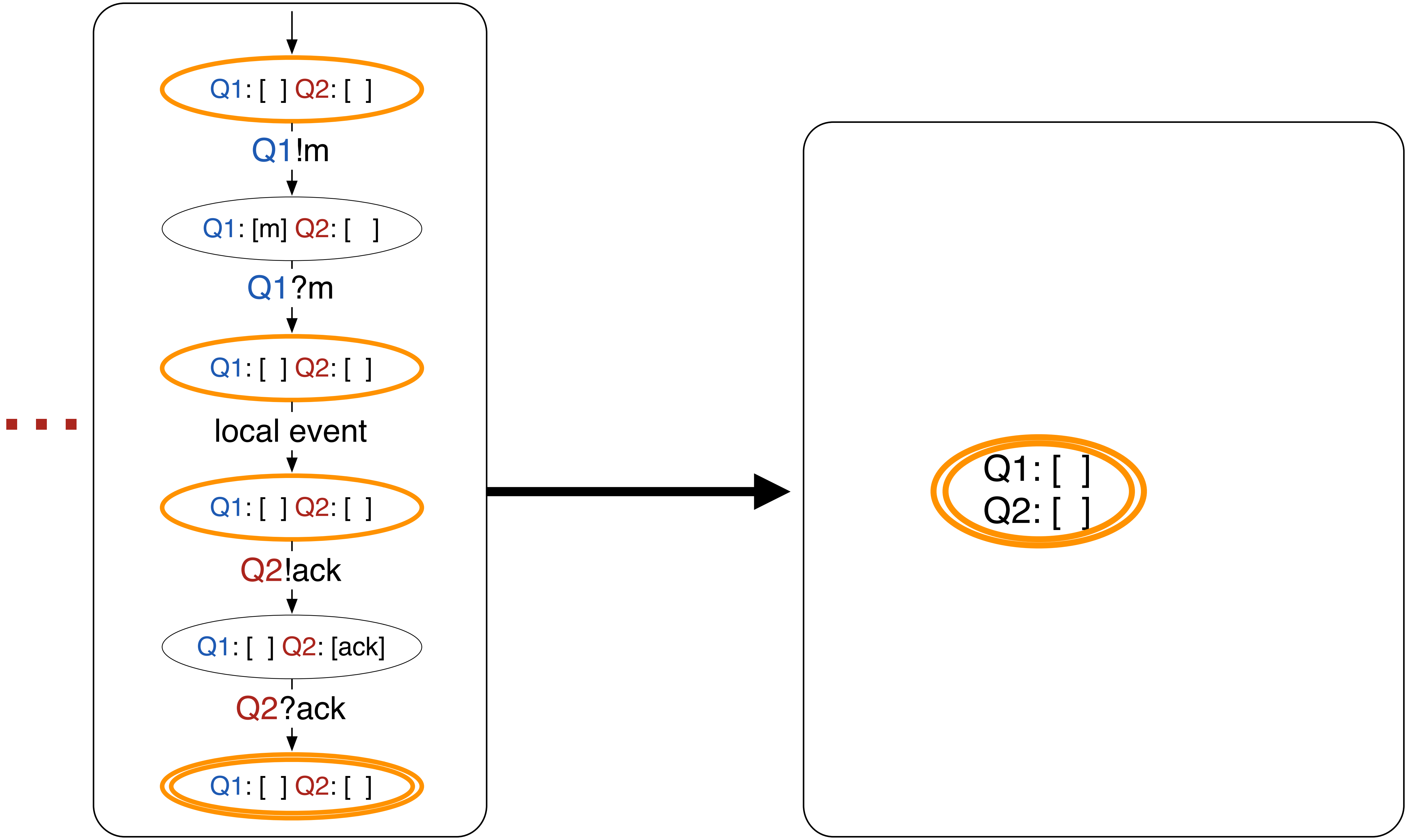
Abstract concrete queue states



State-based DAGs

Initial global model

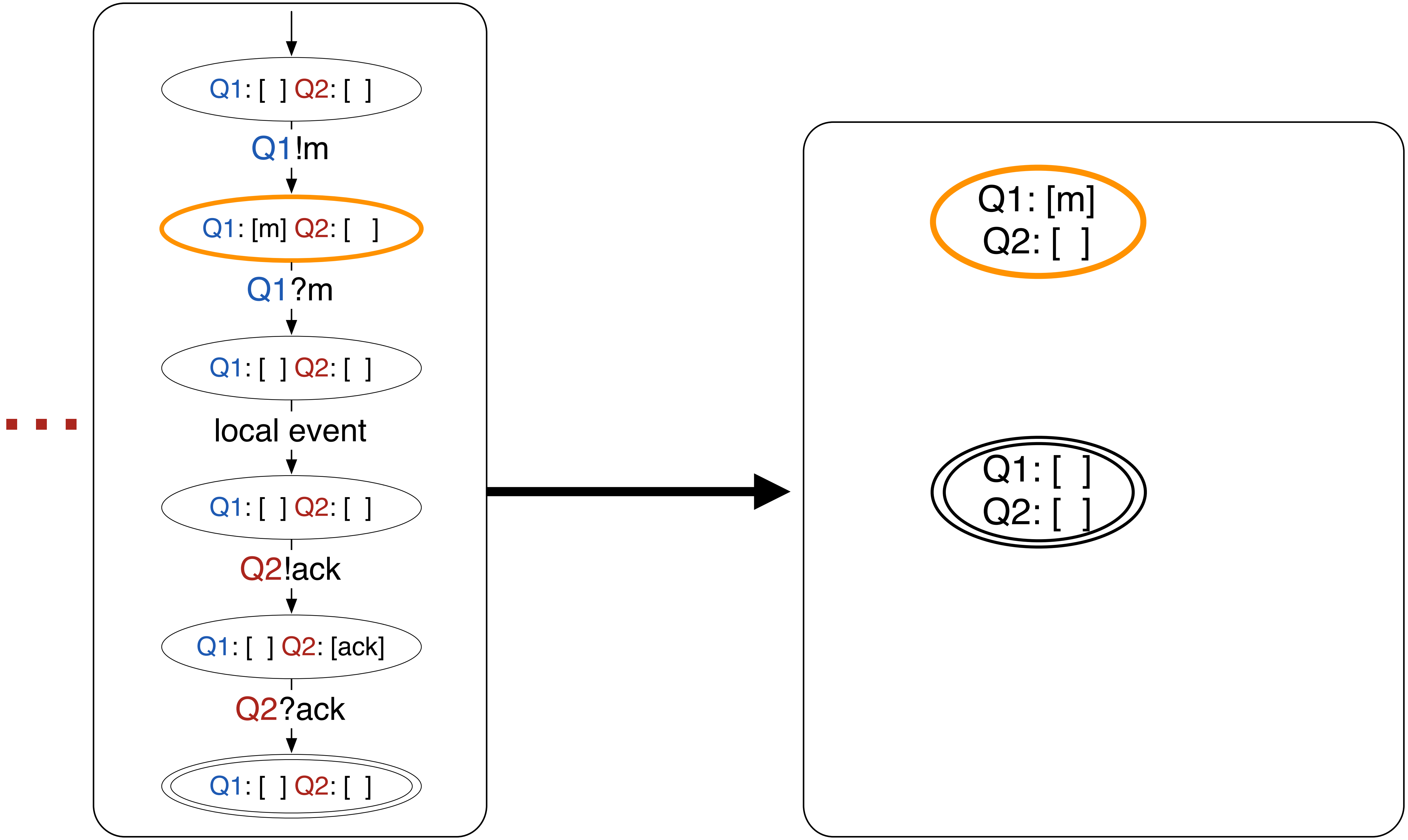
“Merge” concrete states into abstract states



State-based DAGs

Initial global model

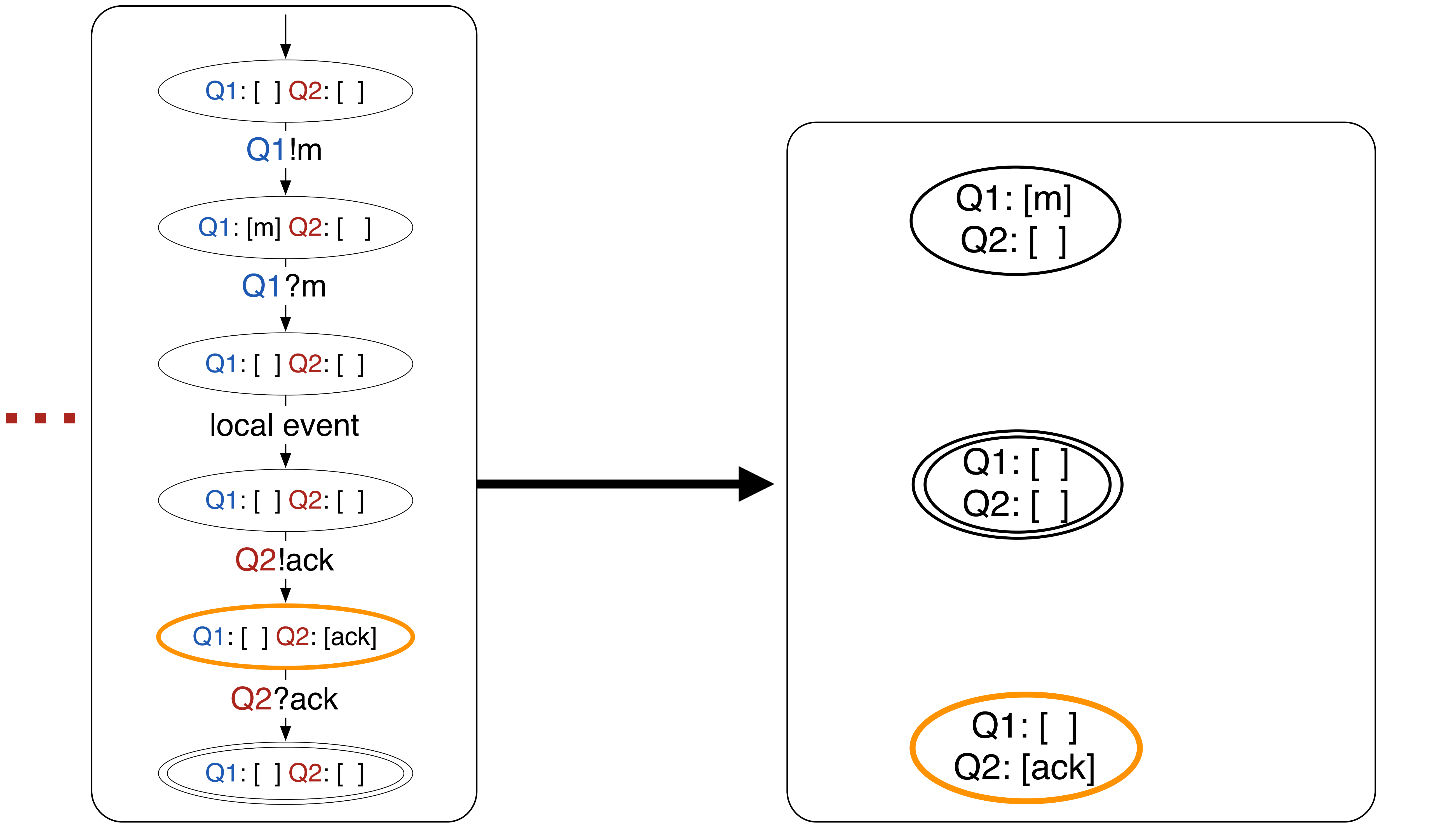
"Merge" concrete states into abstract states



State-based DAGs

Initial global model

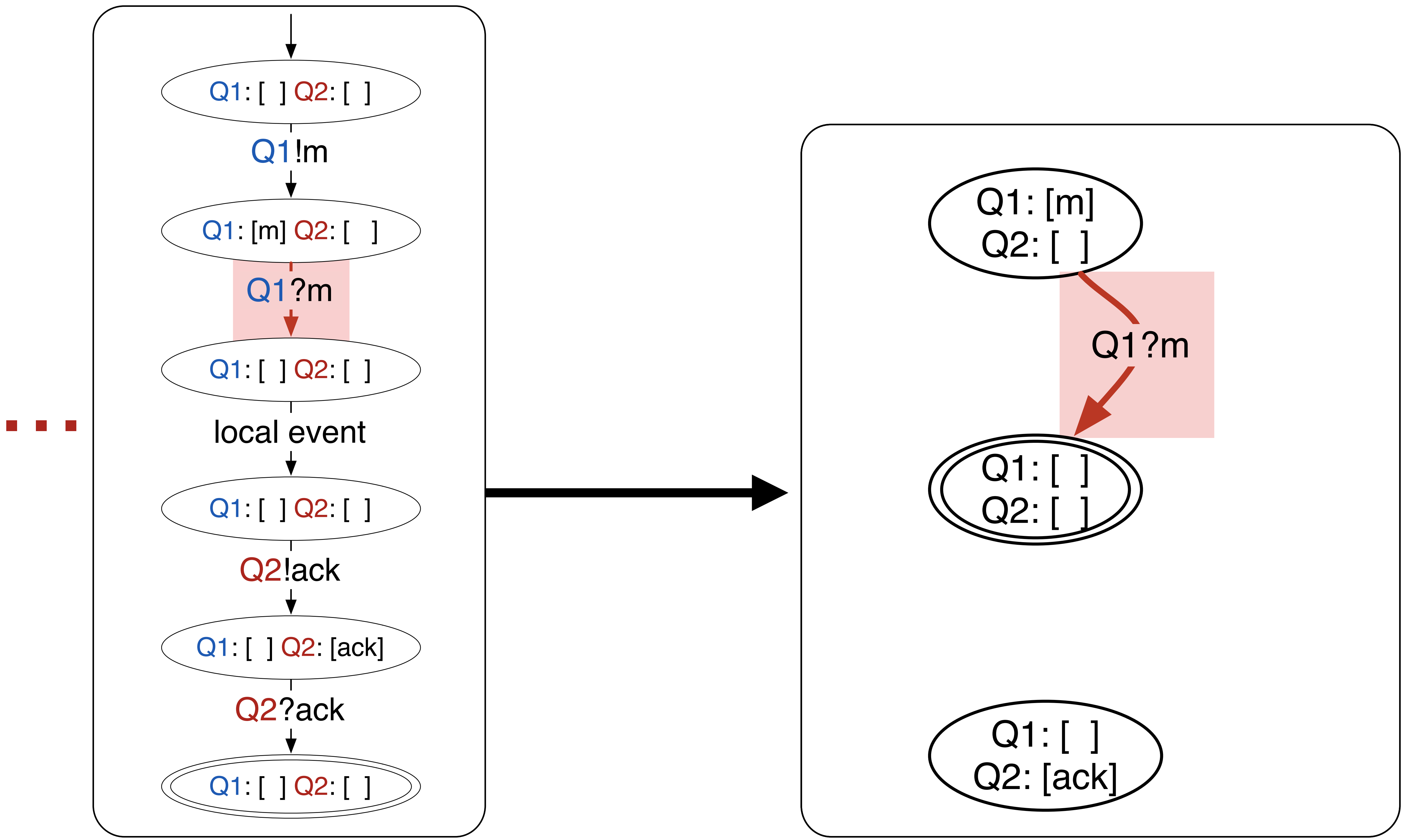
“Merge” concrete states into abstract states



State-based DAGs

Initial global model

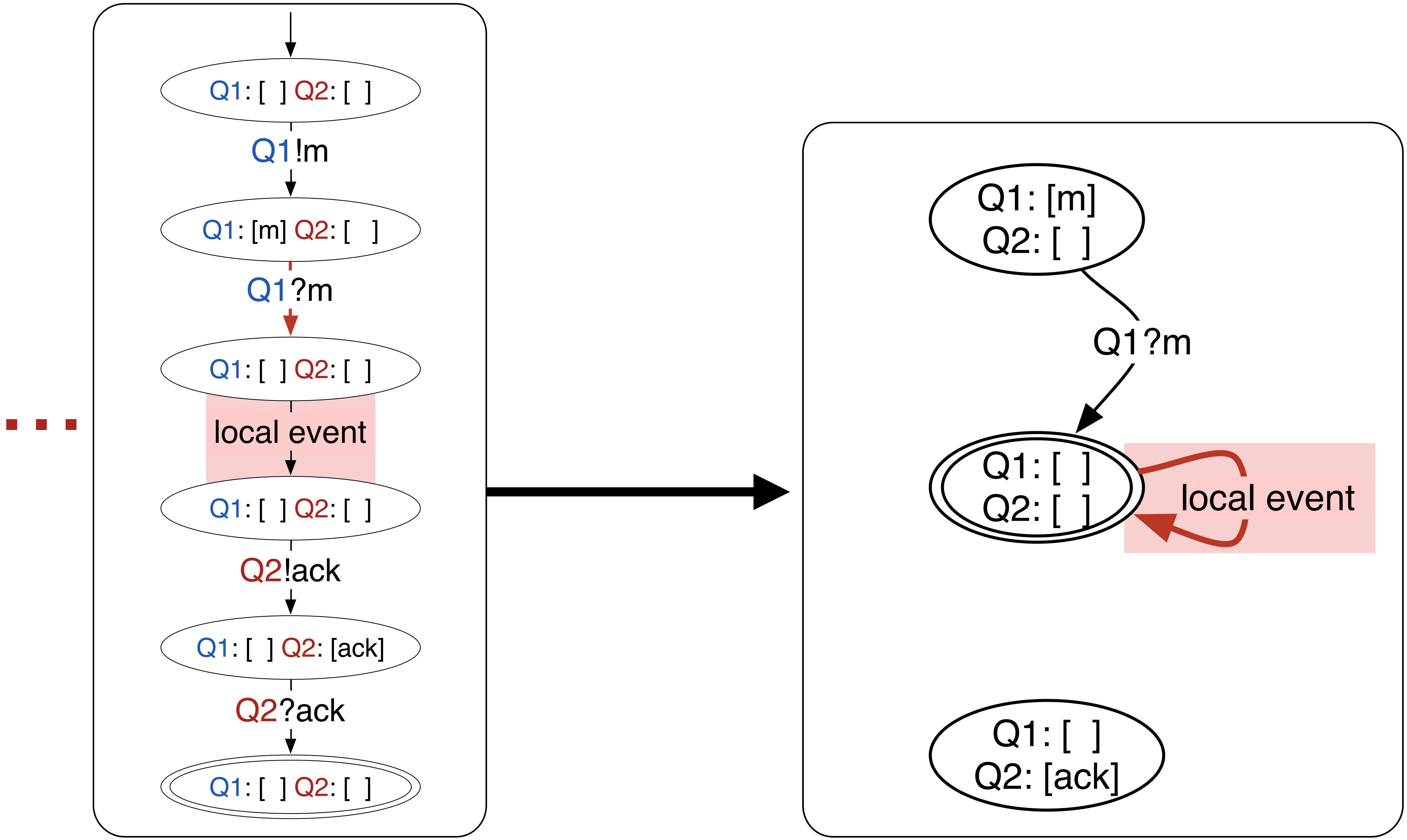
Generate abstract edges



State-based DAGs

Initial global model

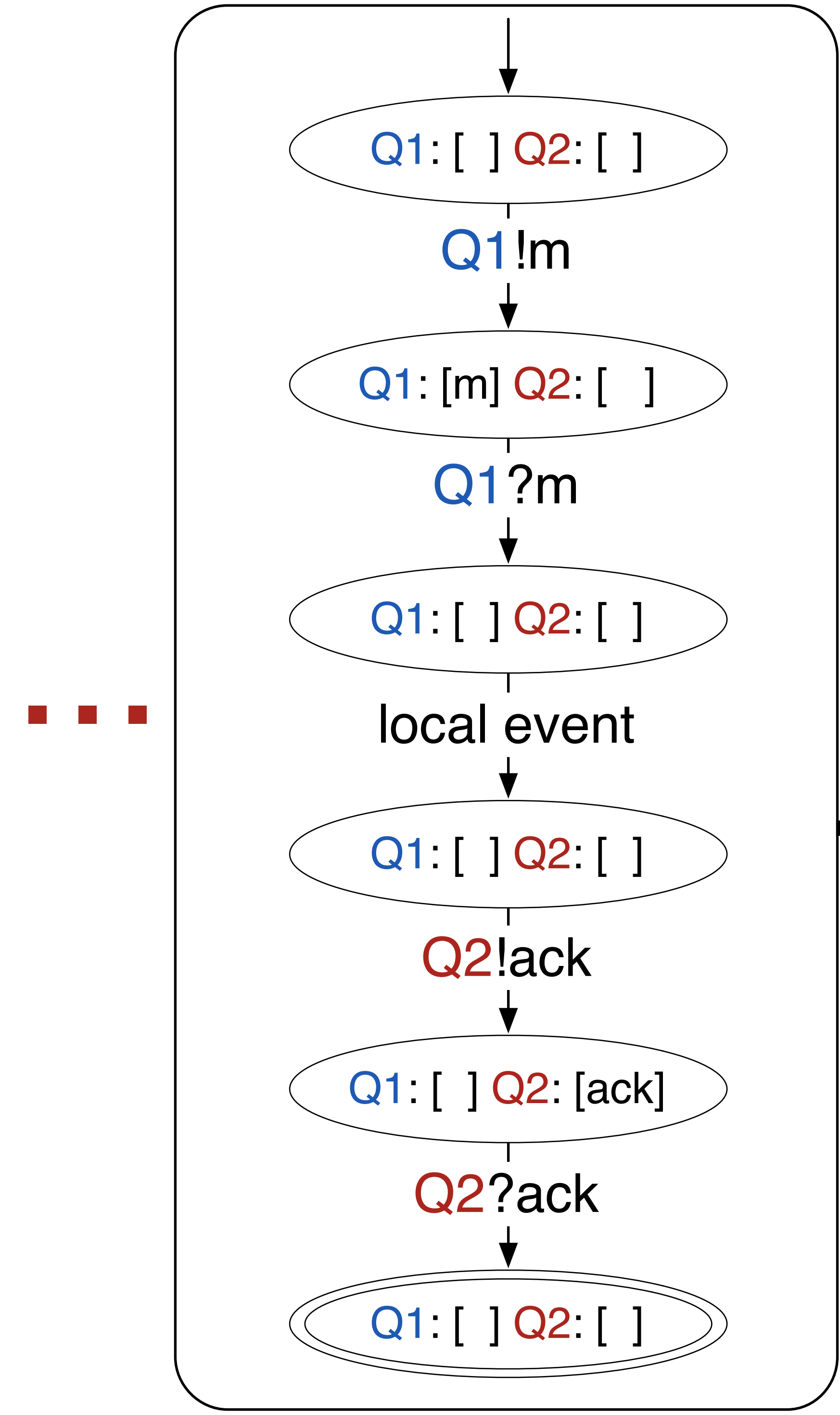
Generate abstract edges



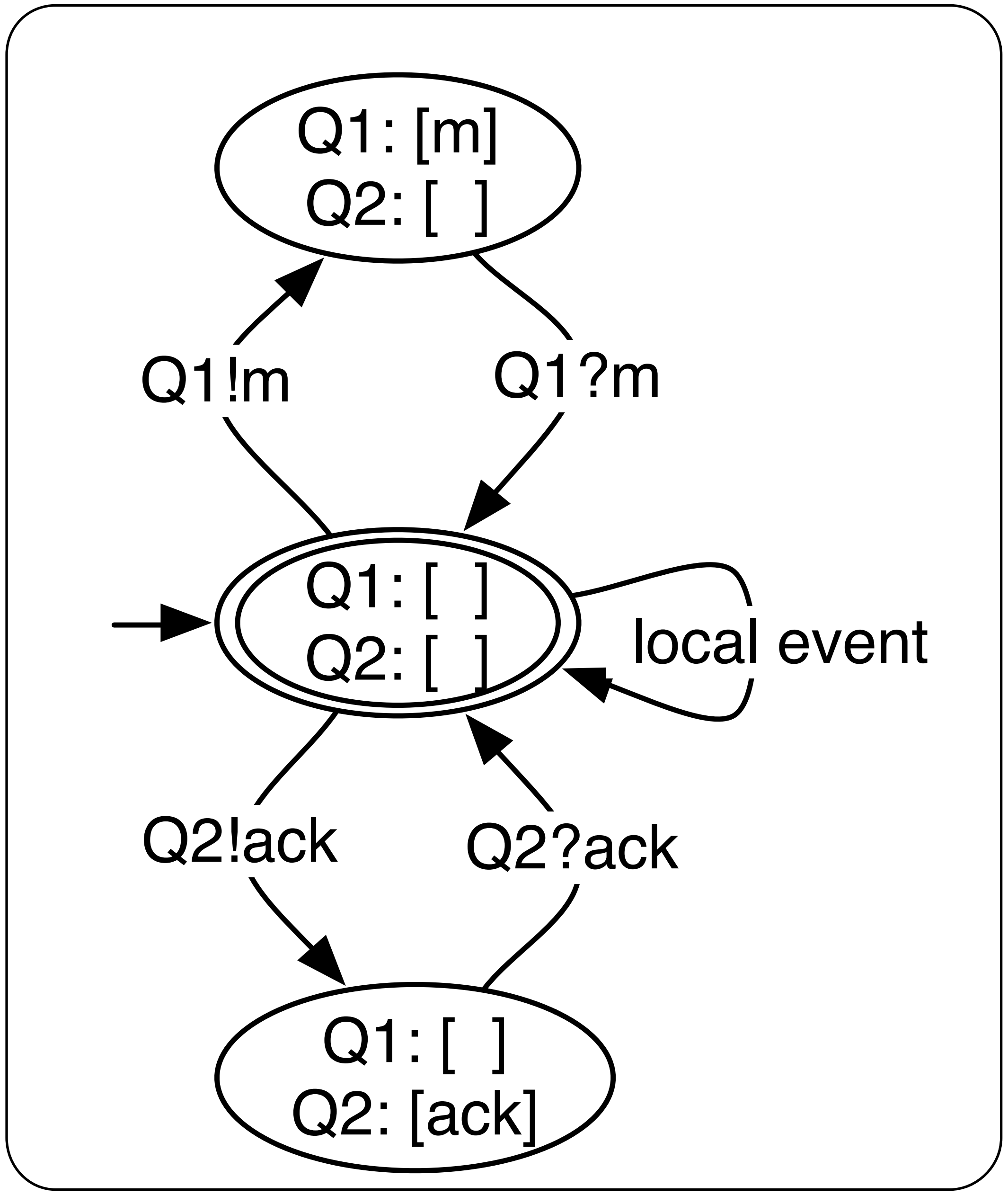
State-based DAGs

Initial global model

Generate abstract edges

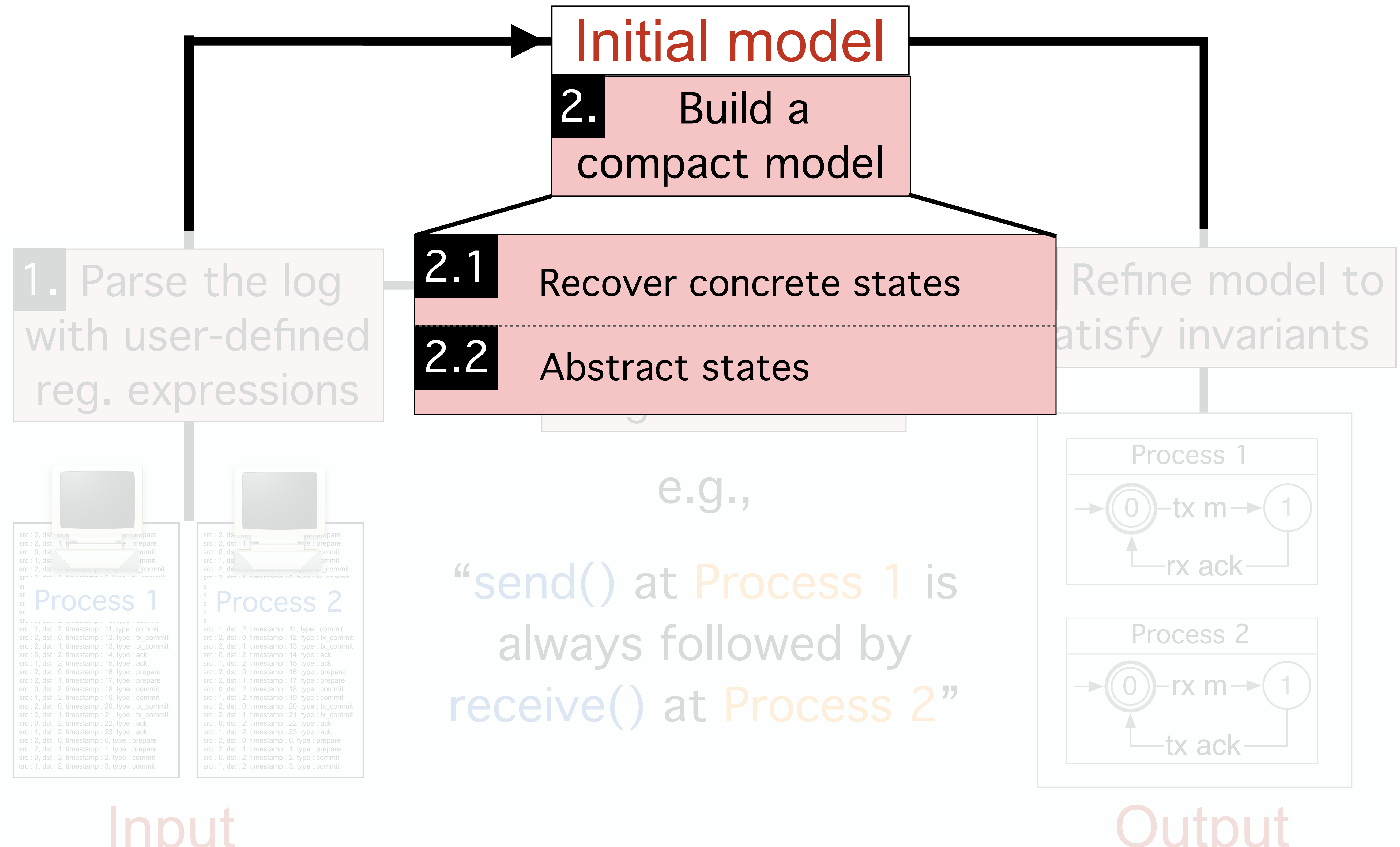


State-based DAGs

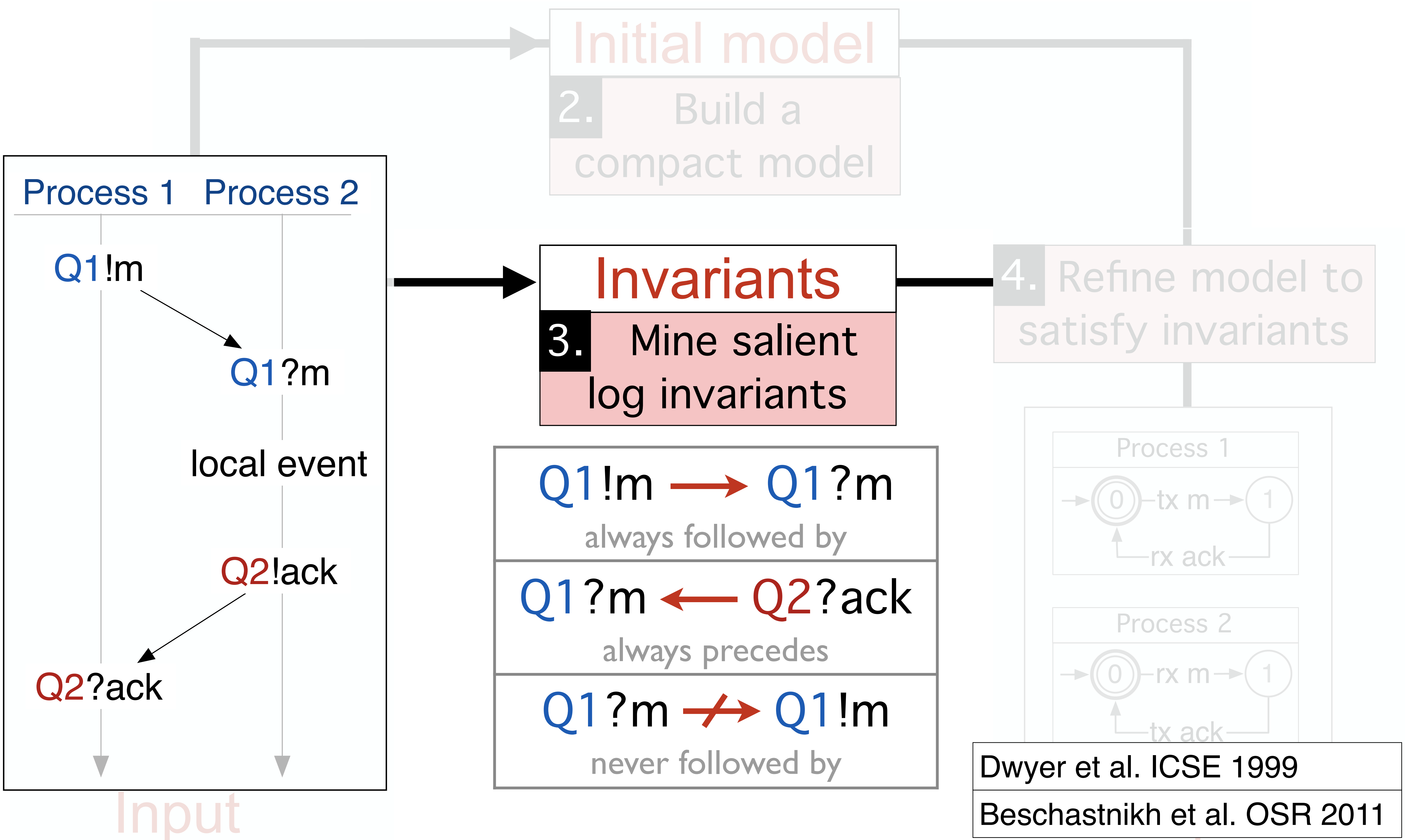


Initial global model

CSight approach



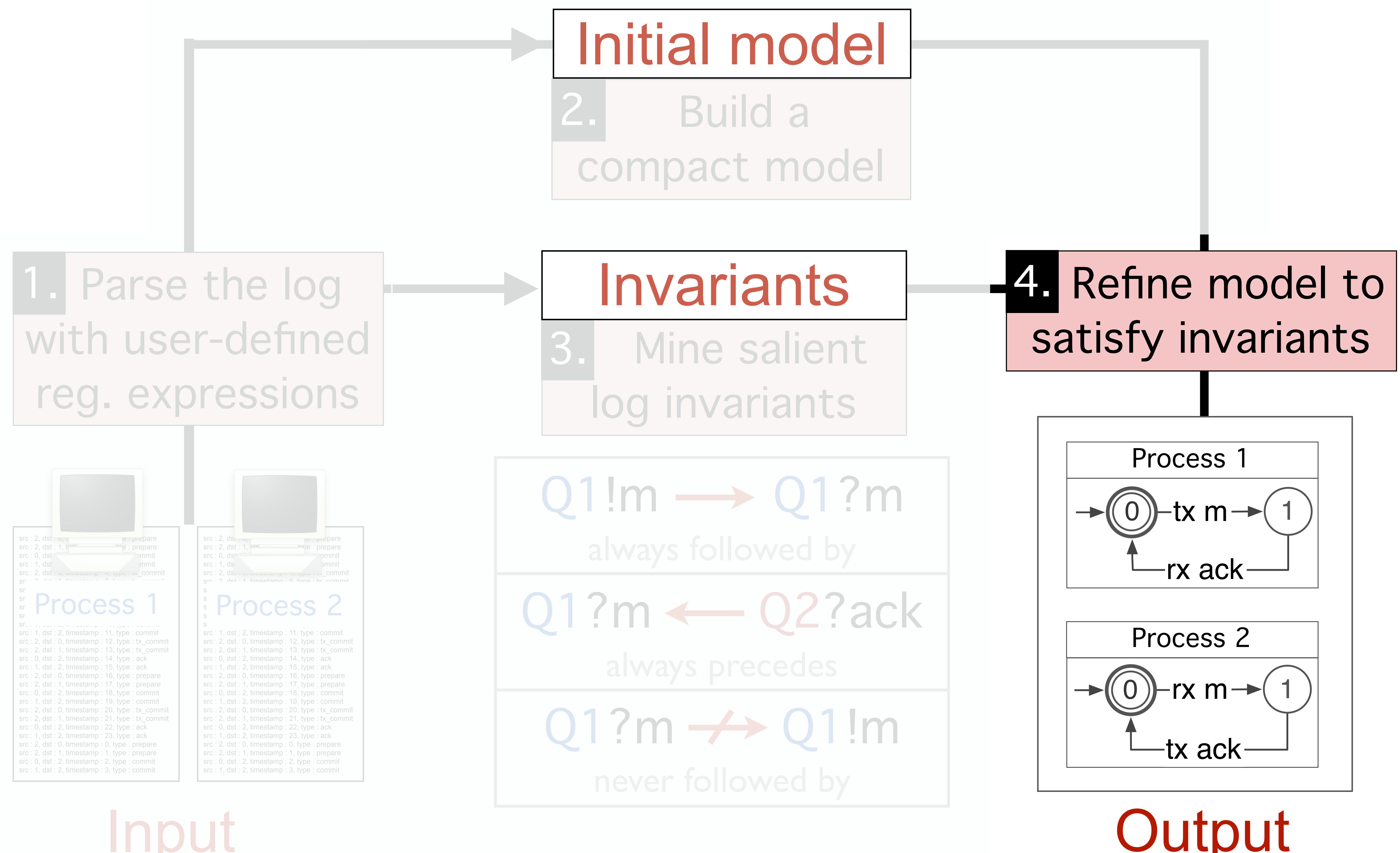
CSight approach



Dwyer et al. ICSE 1999

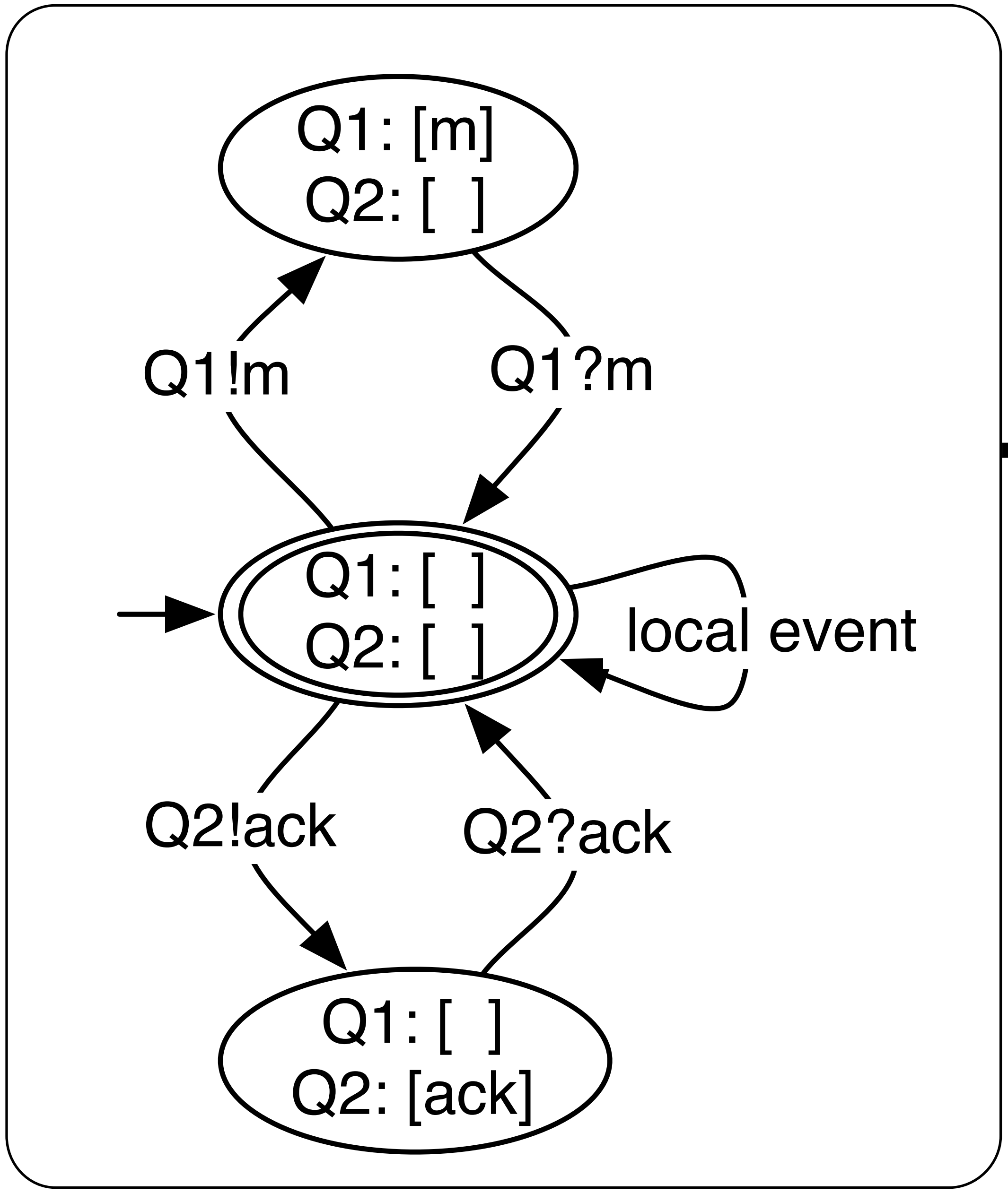
Beschastnikh et al. OSR 2011

CSight approach

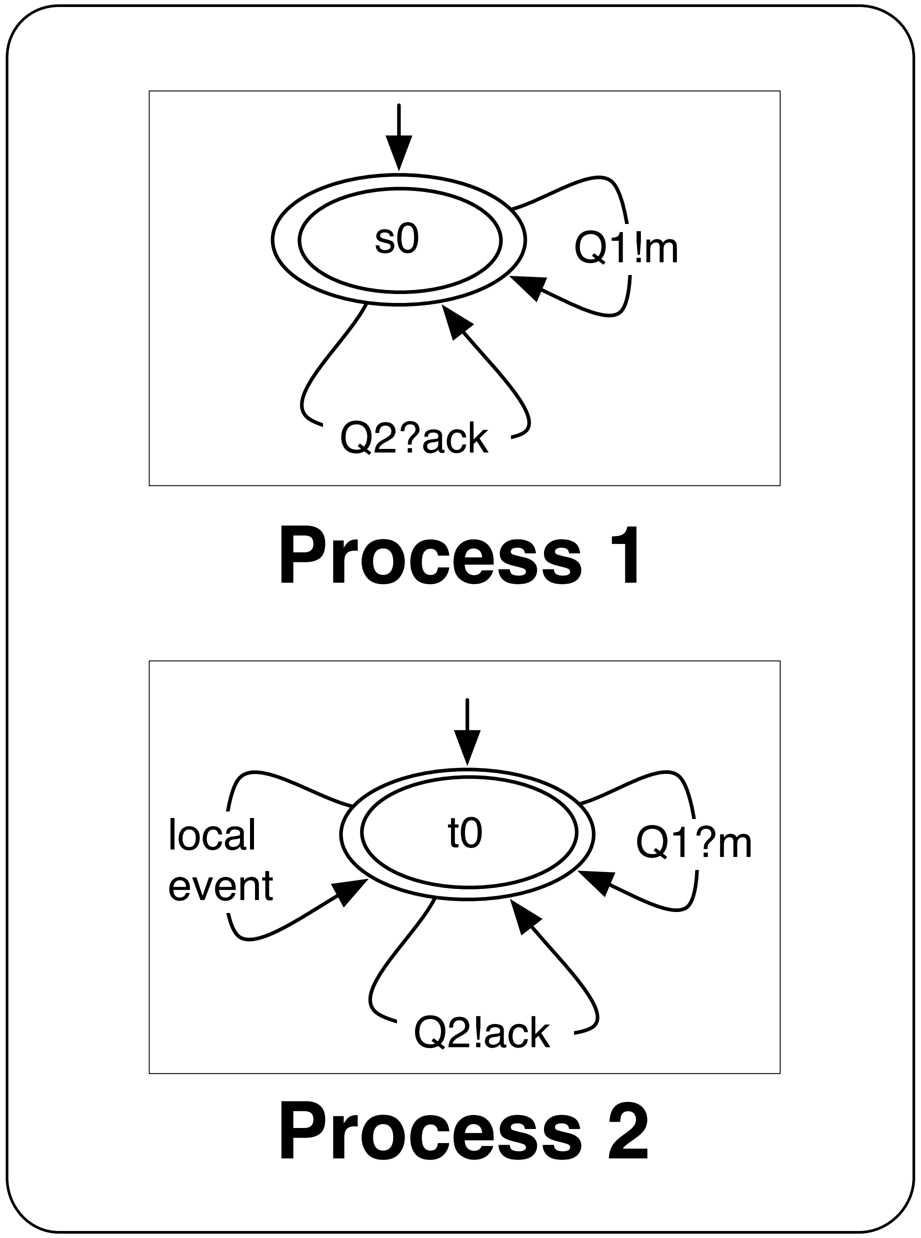


Deriving a Communicating FSM

- Decompose the global model into per-process FSMs

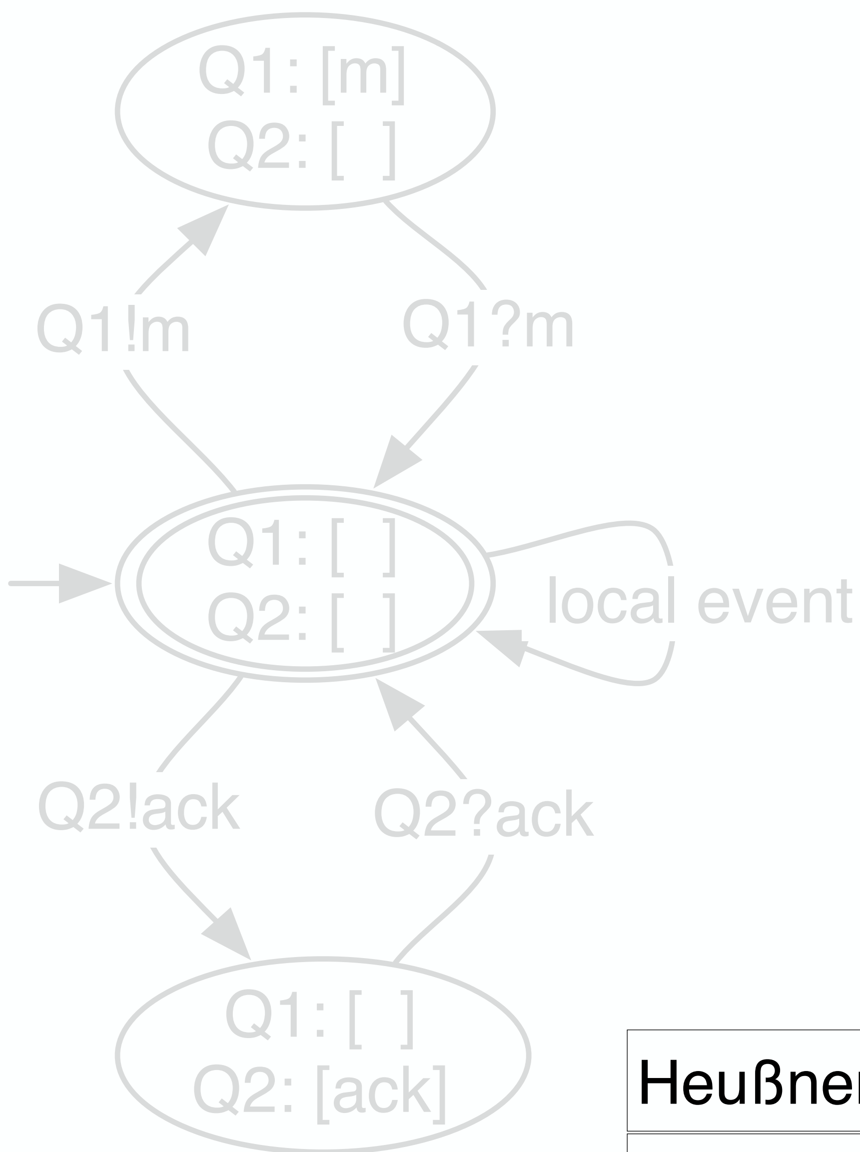


Compact global model

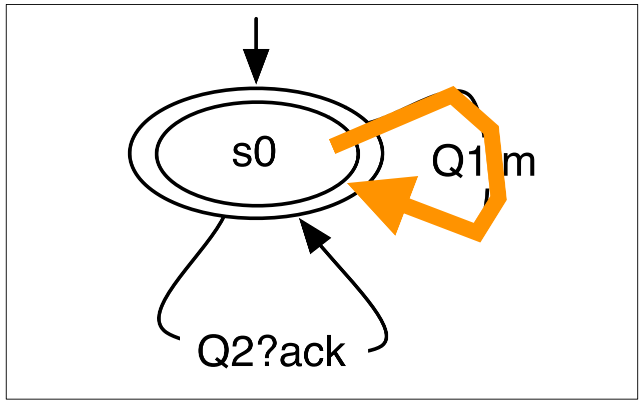


CFSM model

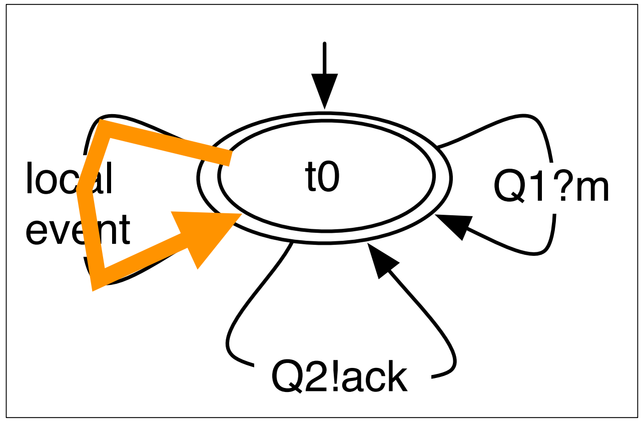
Making the compact model accurate



Check an invariant in the CFSM
Q1!m → Q1?m



Process 1



Process 2

Heußner et al. TACAS 2012. (McScM)

Holzmann TSE 1997. (SPIN)

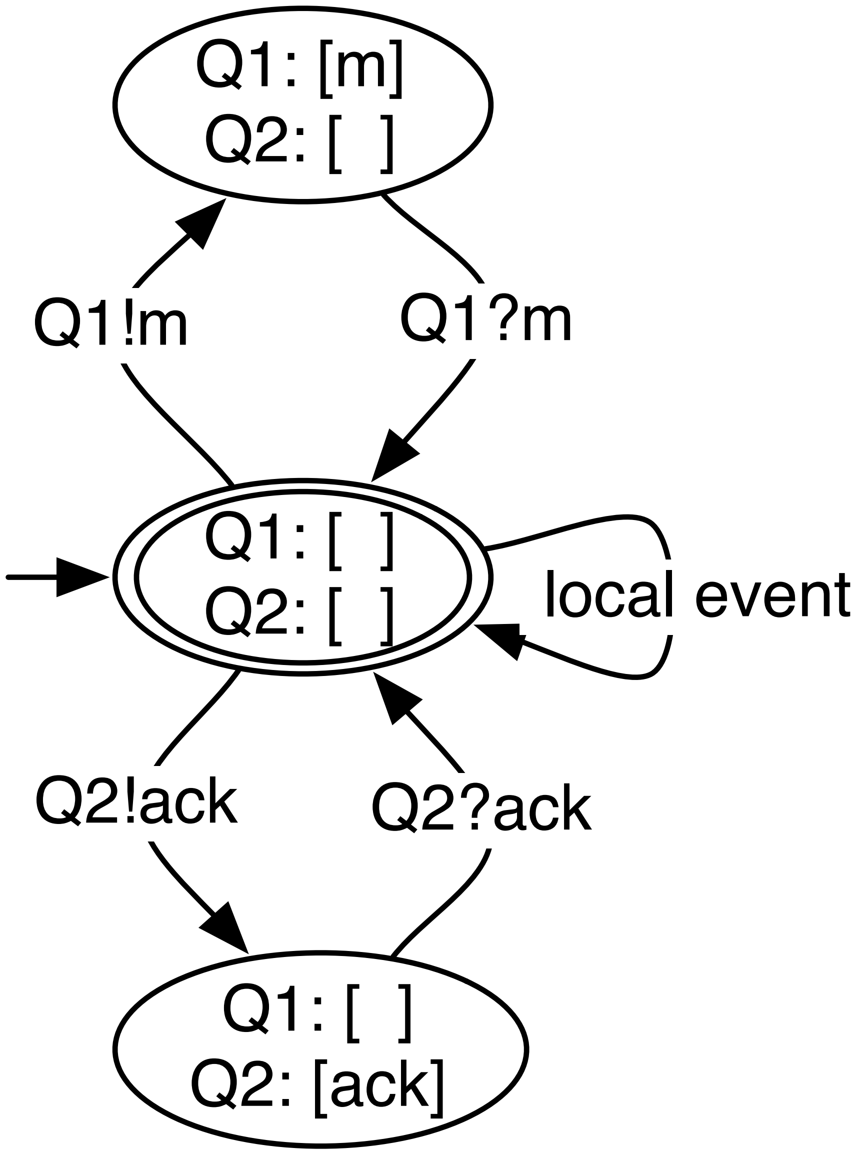
Making the compact model accurate

Refine global model to remove invariant counter-example

1. Q1!m
2. local event

Check an invariant in the CFSM

Q1!m → Q1?m



Process 1

Process 2

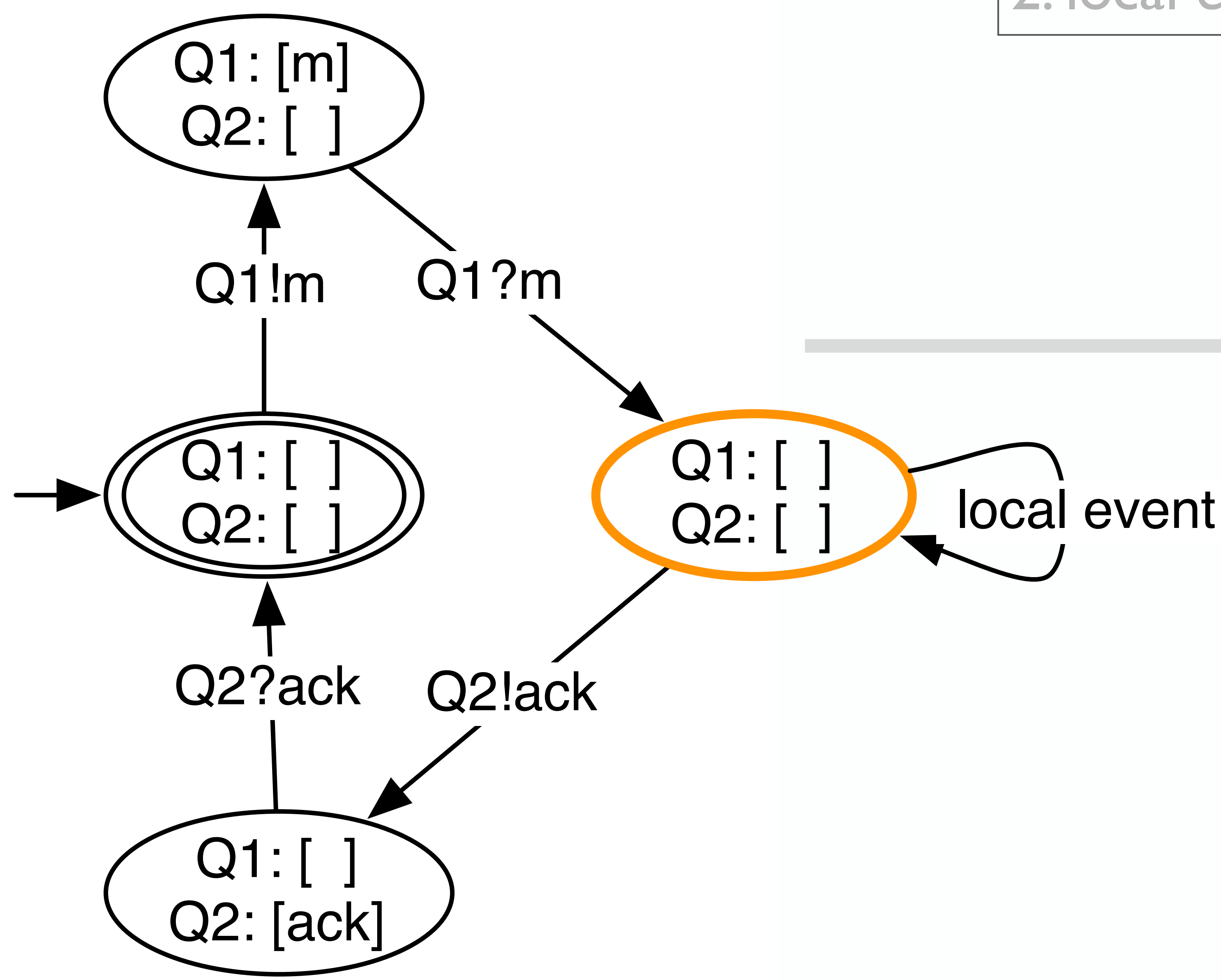
Making the compact model accurate

Refine global model to remove invariant counter-example

Check an invariant in the CFSM

Q1!m → Q1?m

1. Q1!m
2. local event



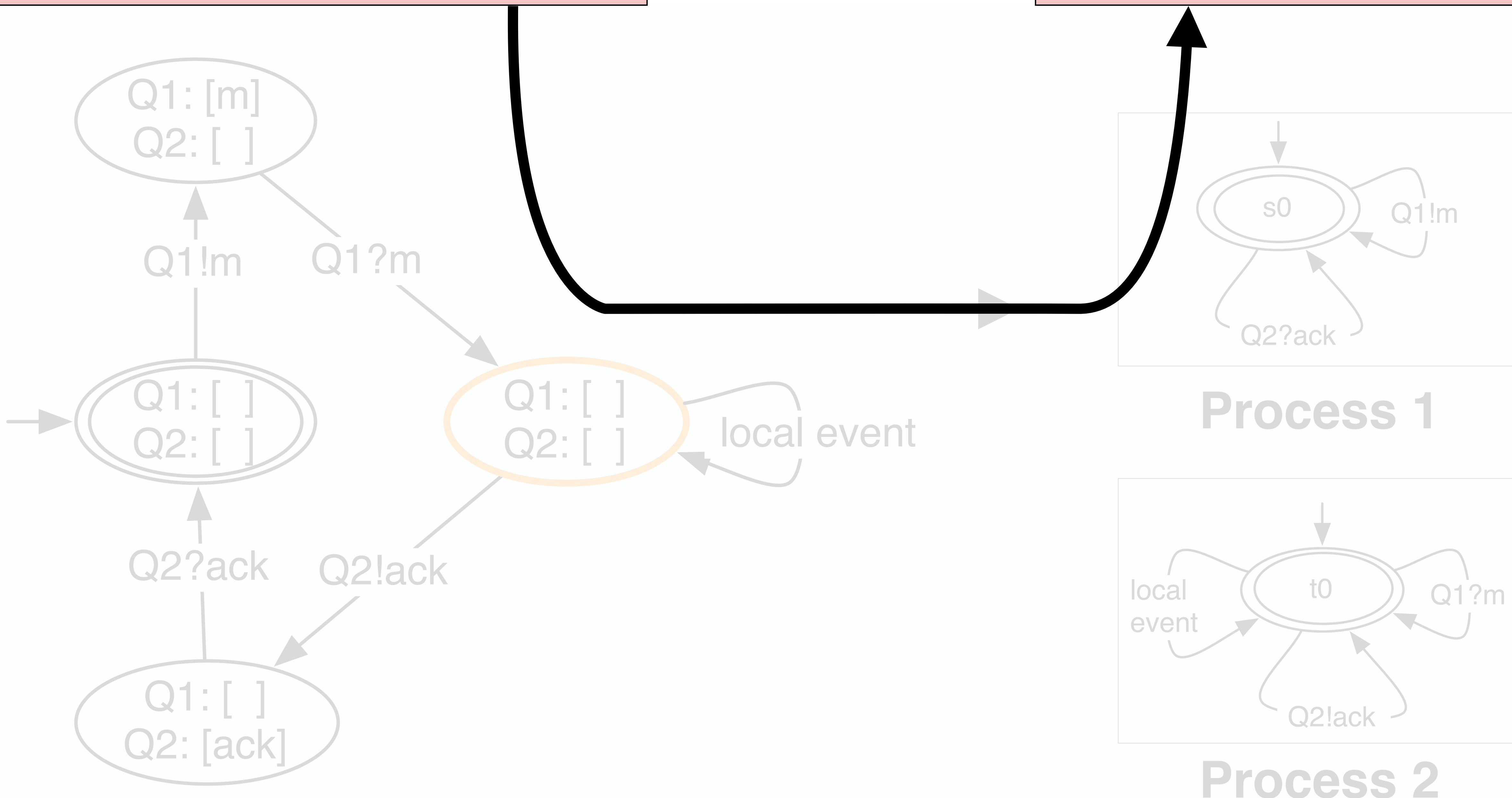
Process 1

Process 2

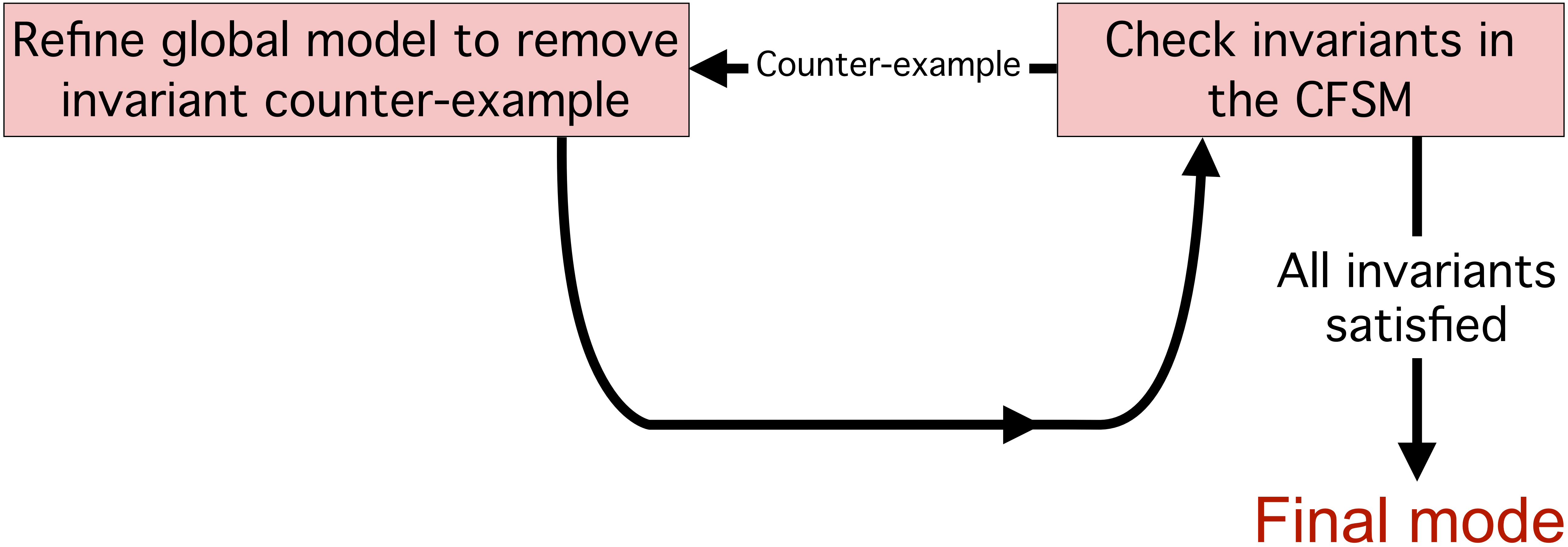
Making the compact model accurate

Refine global model to remove invariant counter-example

Check an invariant in the CFSM



Making the compact model accurate

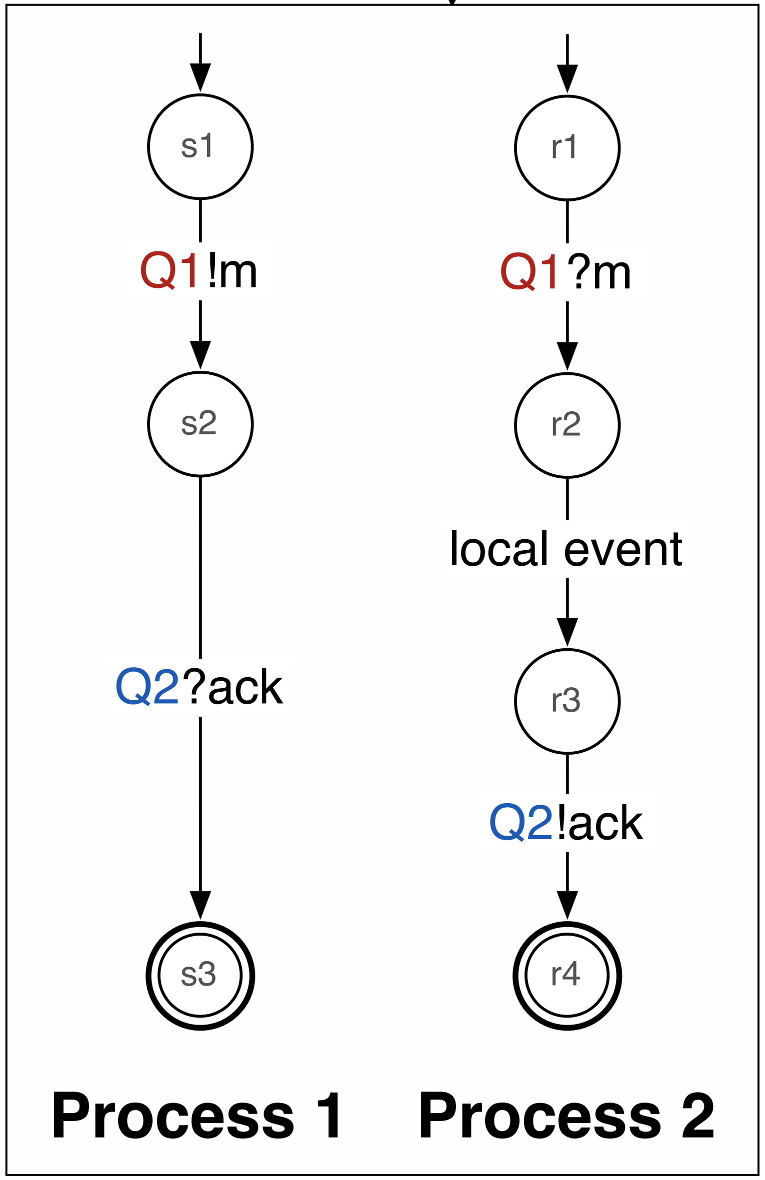


Making the compact model accurate

Refine global model to remove invariant counter-example

Counter-example

Check invariants in the CFSM



Final model

See the paper for more details!

CSight talk outline

- Motivation
- Background
  - Logging: partial order and vector clocks
  - Modeling: communicating FSMs
- CSight approach
- Evaluation

http://synoptic.googlecode.com

CSight evaluation

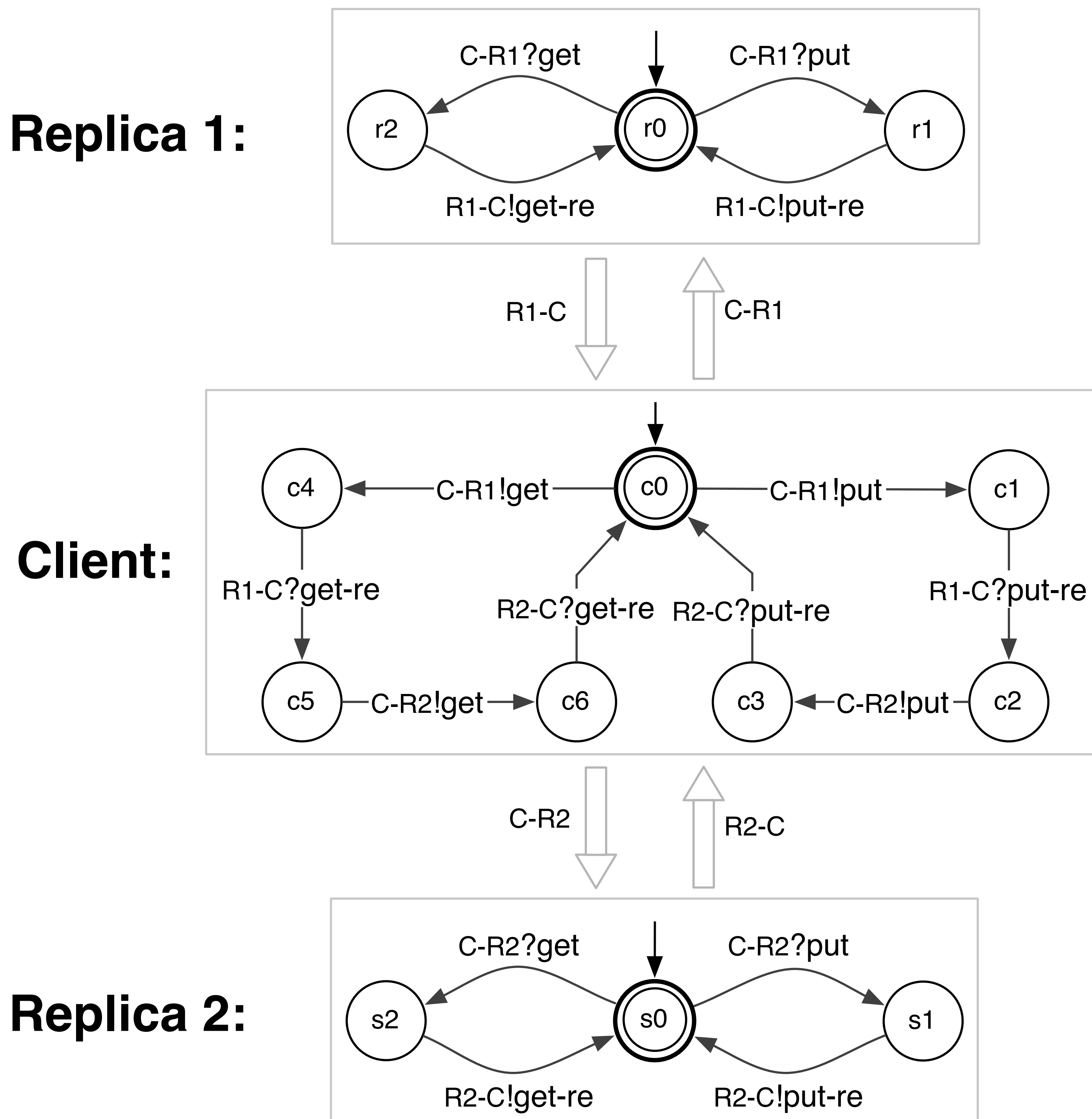
- Efficiency and correctness (proof)

- Utility (case studies and user study)
  - Voldemort DHT replication protocol
  - TCP opening/closing handshakes
  - User study: students found bugs using CFSM models

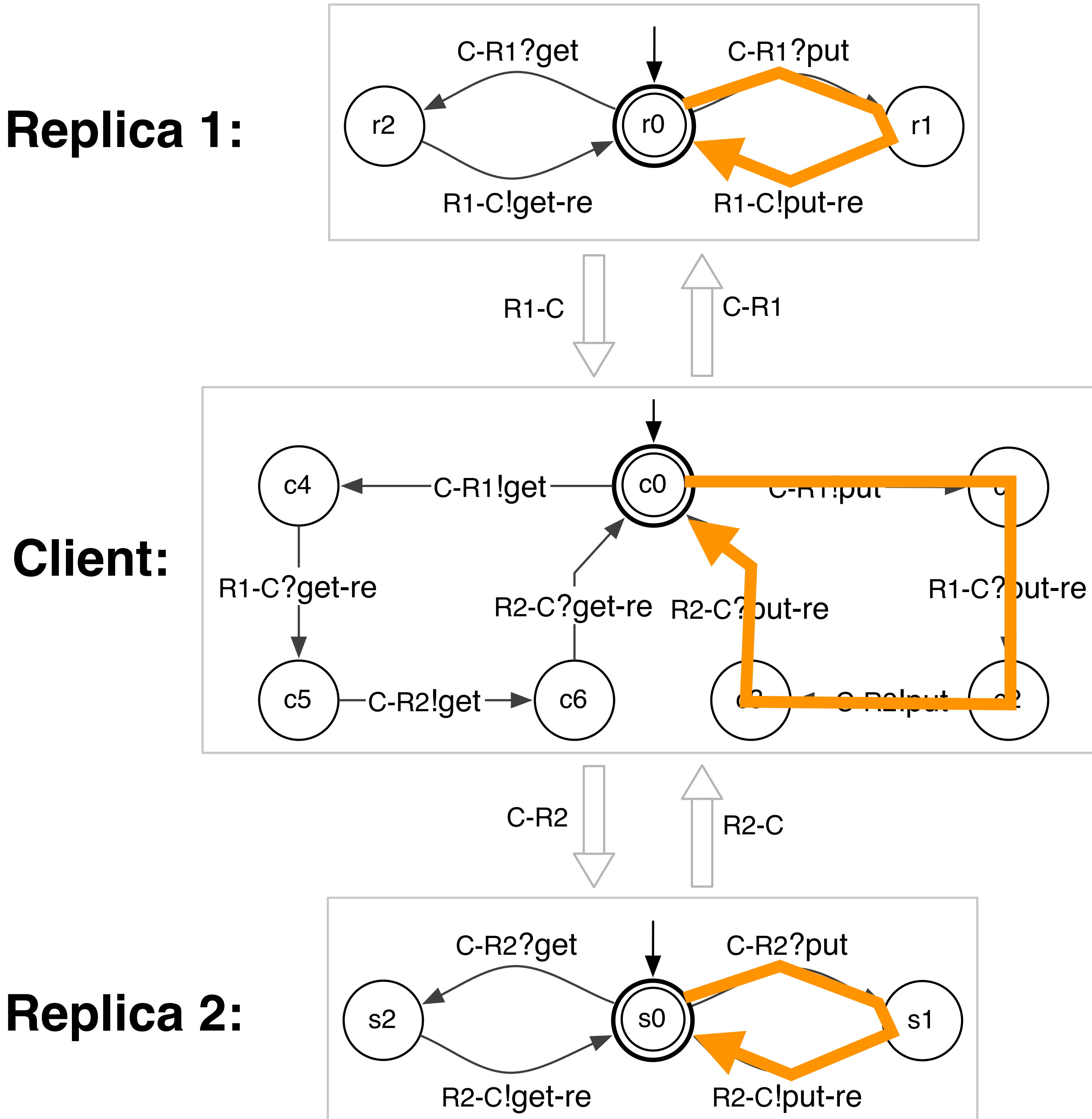
Voldemort Distributed Hash Table

- Voldemort DeCandia et al. SOSP 2007
  - Implements a distributed hash table
    - `put(key,value)`, `get(key)`
  - Deployed at LinkedIn
  - Code is the best protocol documentation
- Ran CSight on logs generated with unit tests
  - Targeted replication protocol messages
  - No modifications to Voldemort

Voldemort replication protocol



Voldemort replication protocol



put(key,value) execution path

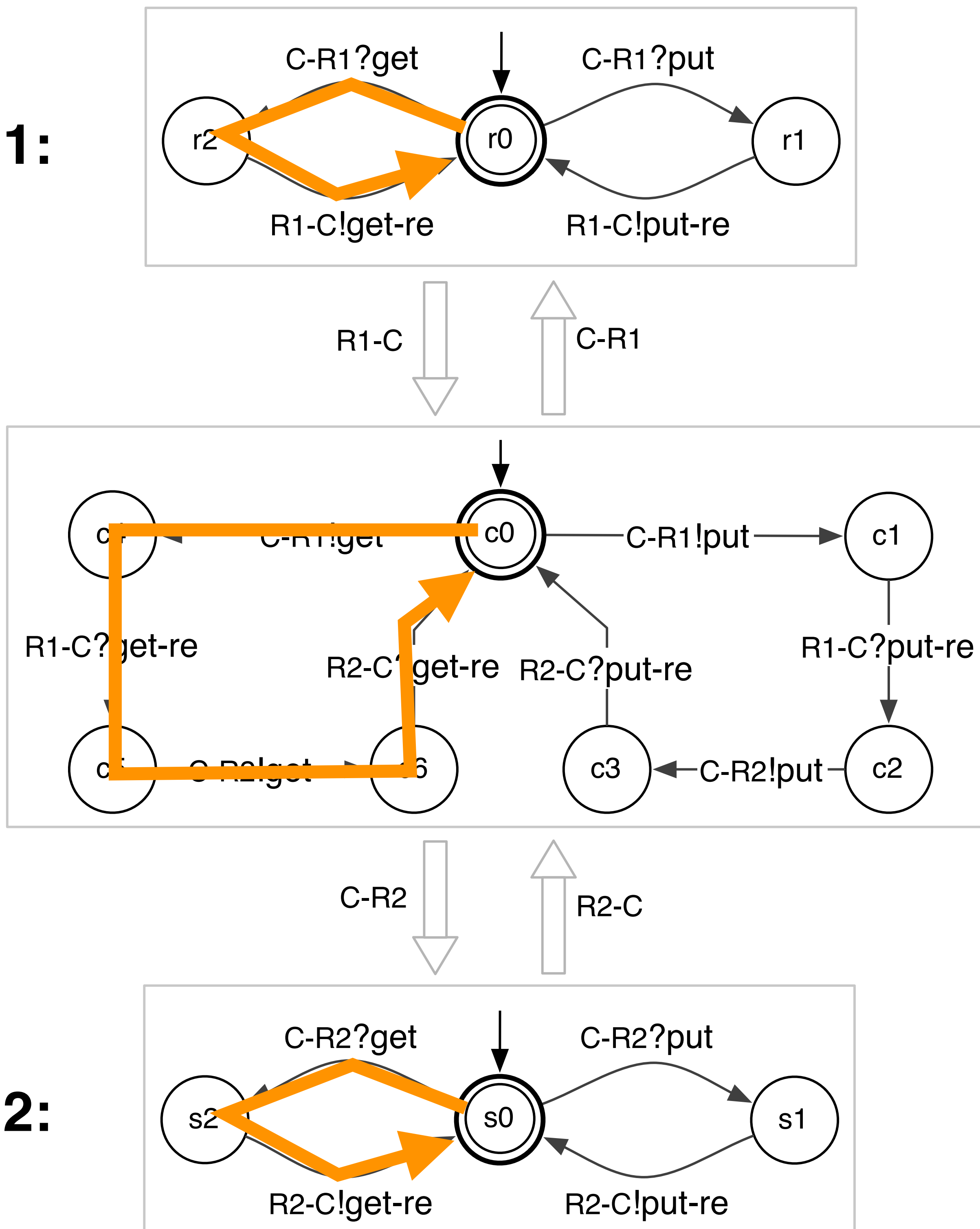
Voldemort replication protocol

get(key)
execution path

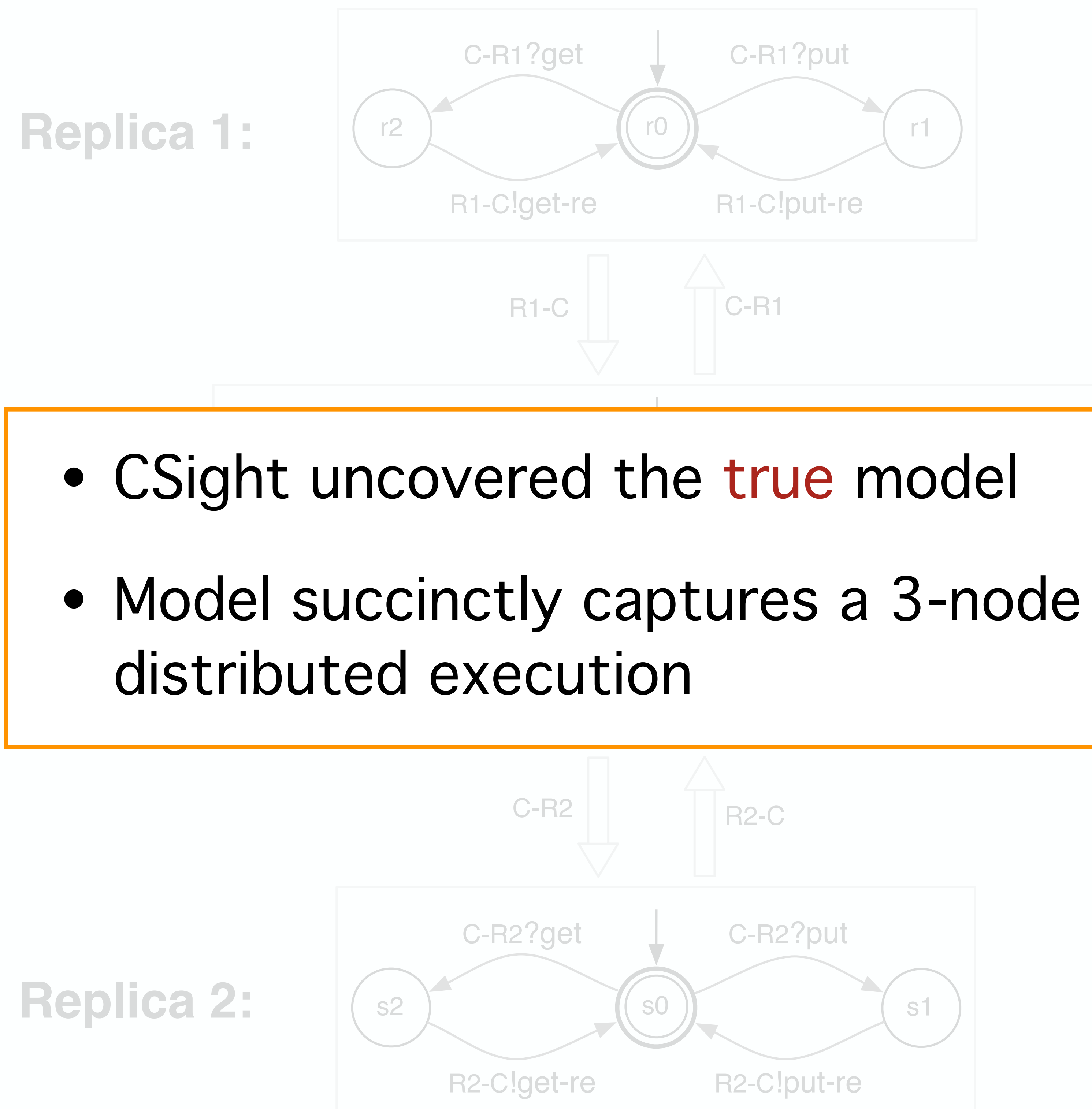
Replica 1:

Client:

Replica 2:



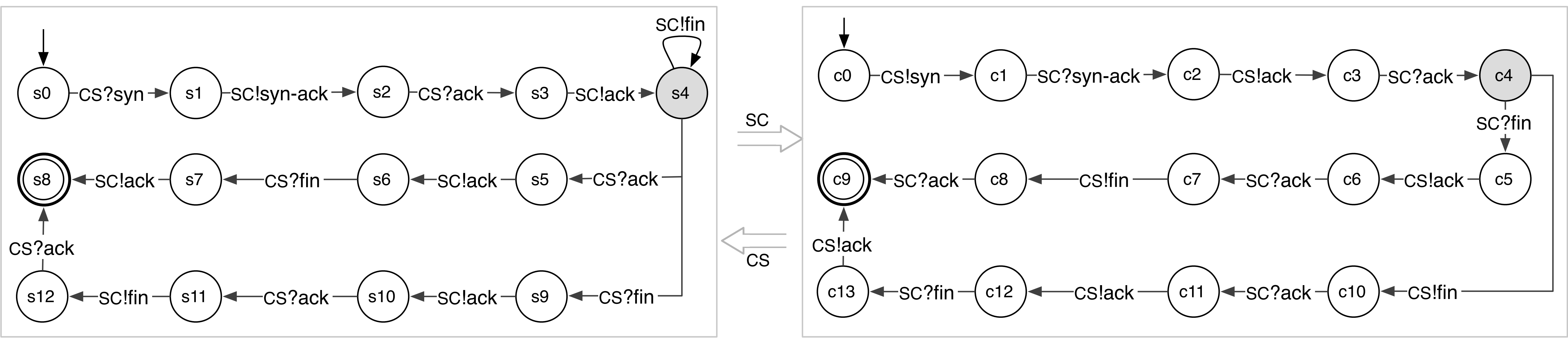
Voldemort replication protocol



- CSight uncovered the true model
- Model succinctly captures a 3-node distributed execution

TCP opening/closing handshake

- Log generated using netcat/dummynet/tcpdump
- Semi-automatically annotated with vector timestamps
- CSight ran on traces with opening/closing handshakes

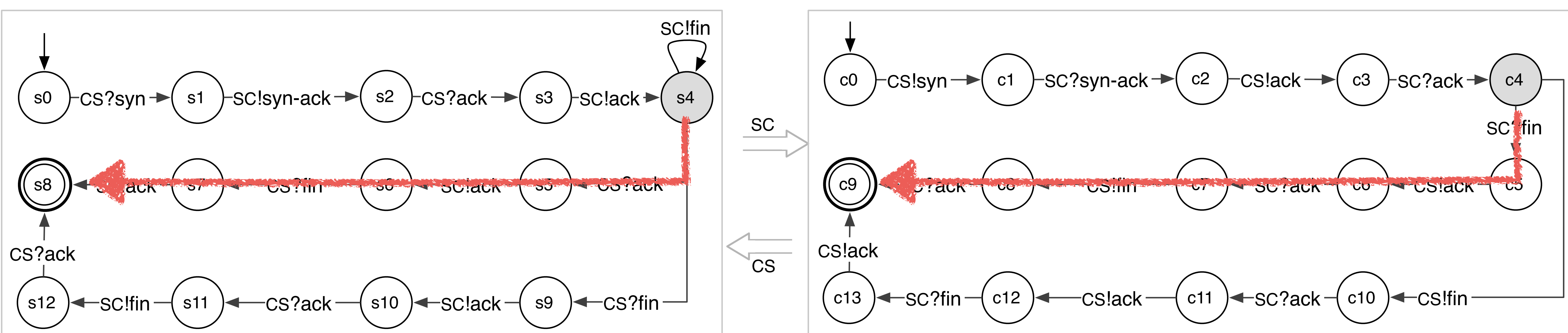


(a) TCP server

(b) TCP client

TCP opening/closing handshake

- Log generated using netcat/dummynet/tcpdump
- Semi-automatically annotated with vector timestamps
- CSight ran on traces with opening/closing handshakes

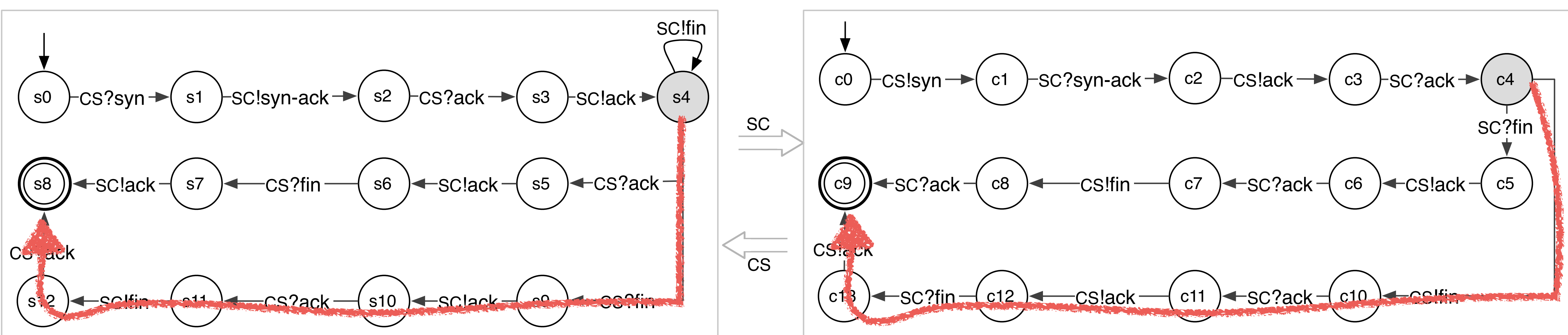


(a) TCP server **(b) TCP client**

Server-initiated teardown

TCP opening/closing handshake

- Log generated using netcat/dummynet/tcpdump
- Semi-automatically annotated with vector timestamps
- CSight ran on traces with opening/closing handshakes



(a) TCP server

(b) TCP client

Client-initiated teardown

TCP opening/closing handshake

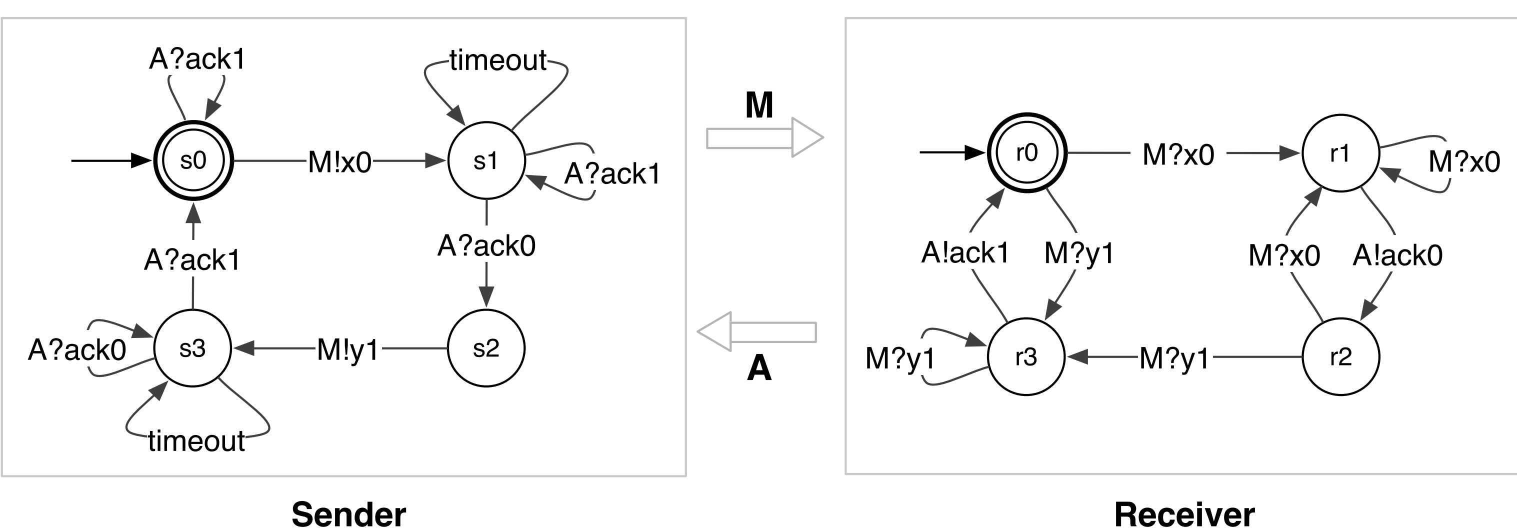
- TCP logs generated using netcat/dummynet/tcpdump
- Semi-au amps
- Ran CSi dshakes

- Customized level of abstraction
- False positive transition, limited expressiveness of the model-checker

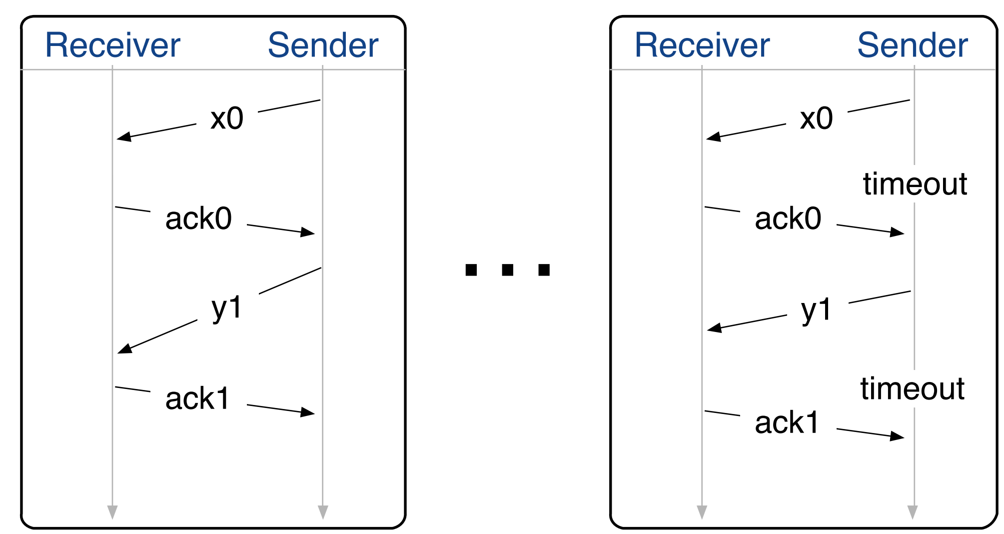
(a) TCP server (b) TCP client

User study on CFSM models

- Compared efficacy of CFSM models and space-time diagrams in bug finding
- Two buggy systems, 39 student participants
- Each participant took a mini-tutorial before bug finding



Example model



Example space-time diagrams

User study on CFSM models

- Compared efficacy of CFSM models and space-time diagrams in bug finding
- Tw
- Ea ... ding

- Students found bugs as well with CFSM models as with space-time diagrams
- Oral feedback indicates that students preferred CFSM models:
  - Could follow process states
  - More concise and easier to follow than long space-time diagrams

A?ack0

timeout

Sender

Receiver

See the paper for study details!

CSight limitations

- Quality and usefulness of the model depends on logged information
- May be difficult to interpret a log you did not generate
- Dynamic analysis: executions may not be representative of the system
- Limited property diversity (can add more)
- Focused on model conciseness, but small model size may be irrelevant

CSight contributions

Logs are rich with information, but inscrutible

Challenges: Contributions:

- Modeling concurrency - Communicating FSM inference
  - Accuracy - Refinement
  - Efficiency - Leveraging mined invariants

CSight contributions

Logs are rich with information, but inscrutible

Challenges:	Contributions:
• Modeling concurrency	- Communicating FSM inference
• Accuracy	- Refinement
• Efficiency	- Leveraging mined invariants

- Elucidates distributed protocols
- Beginner developers can find bugs with CFSMs
- Proved useful properties about the approach

- Open source: synoptic.googlecode.com