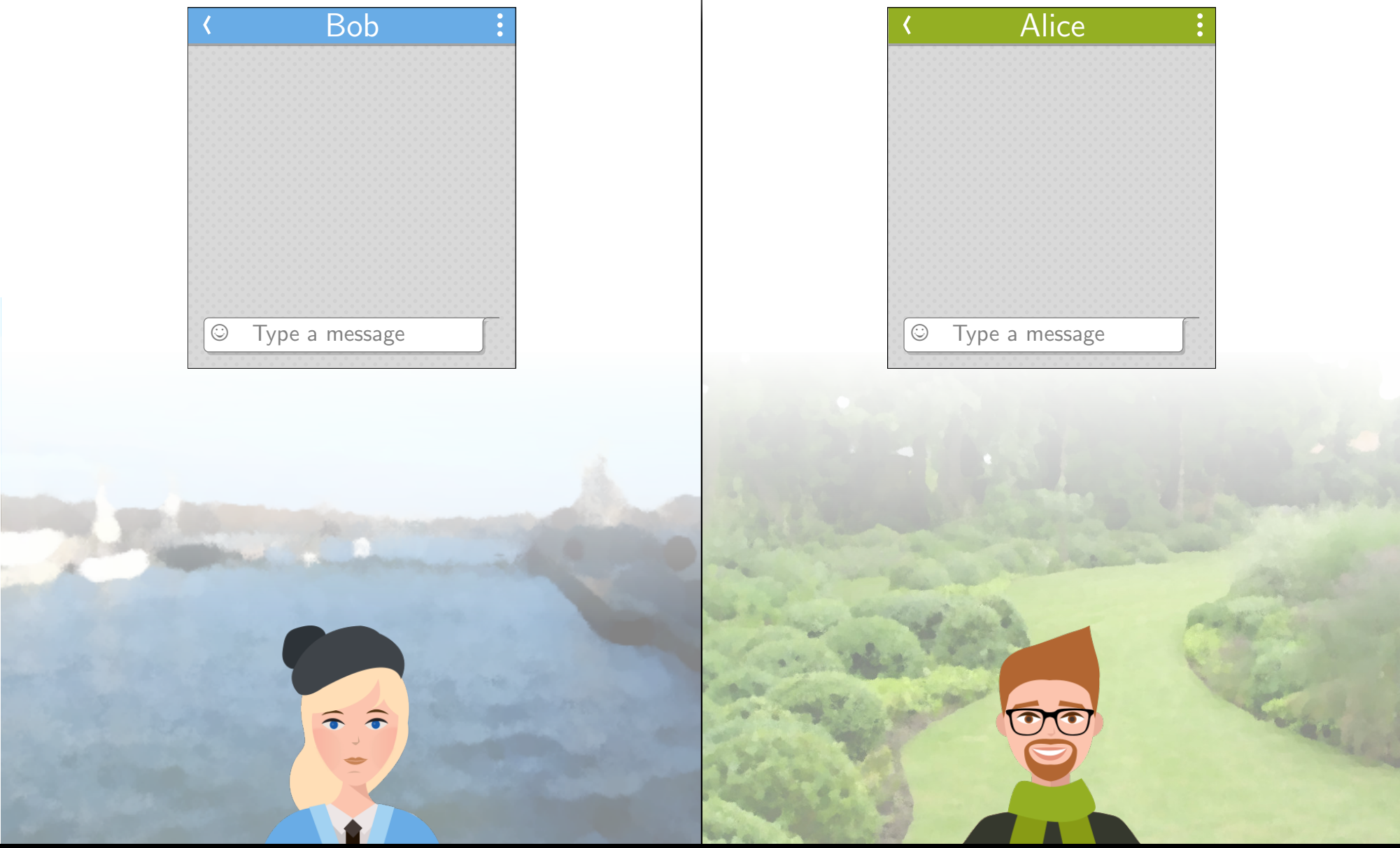
Does the Operational Model Capture Partition Tolerance in Distributed Systems?

Grégoire Bonin Achour Mostéfaoui Matthieu Perrin

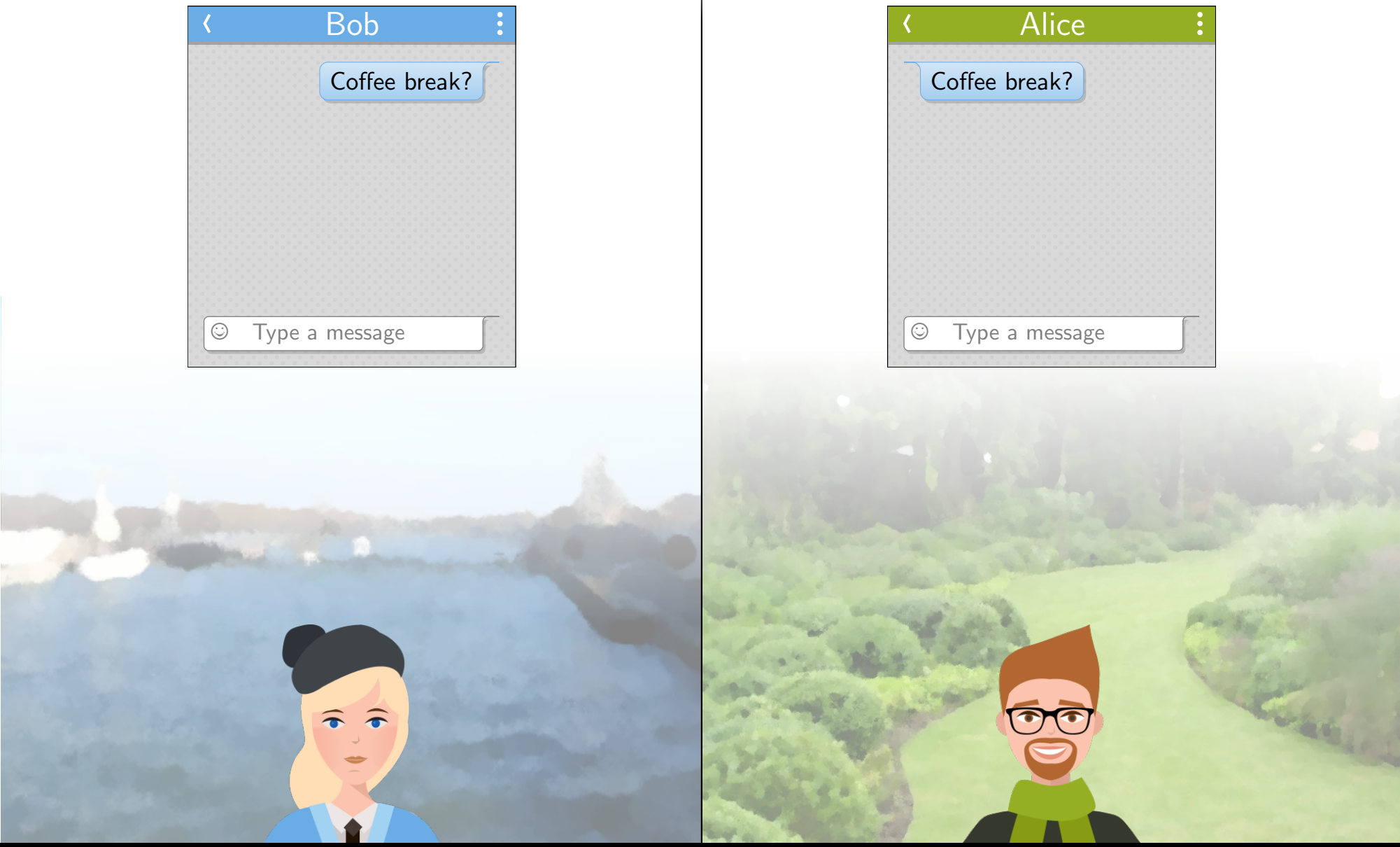
LS2N - Université de Nantes

21/08/2019

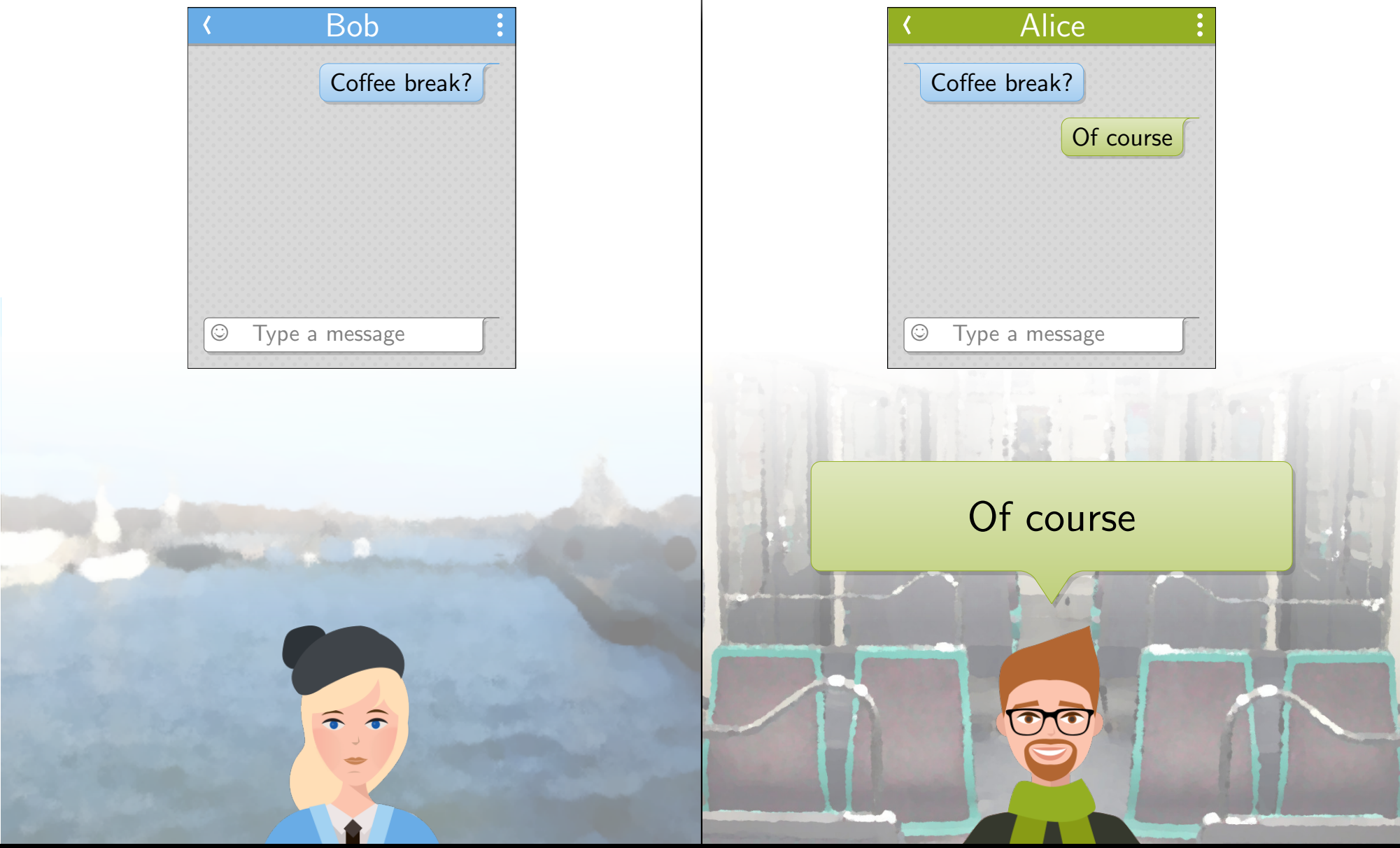
Motivation



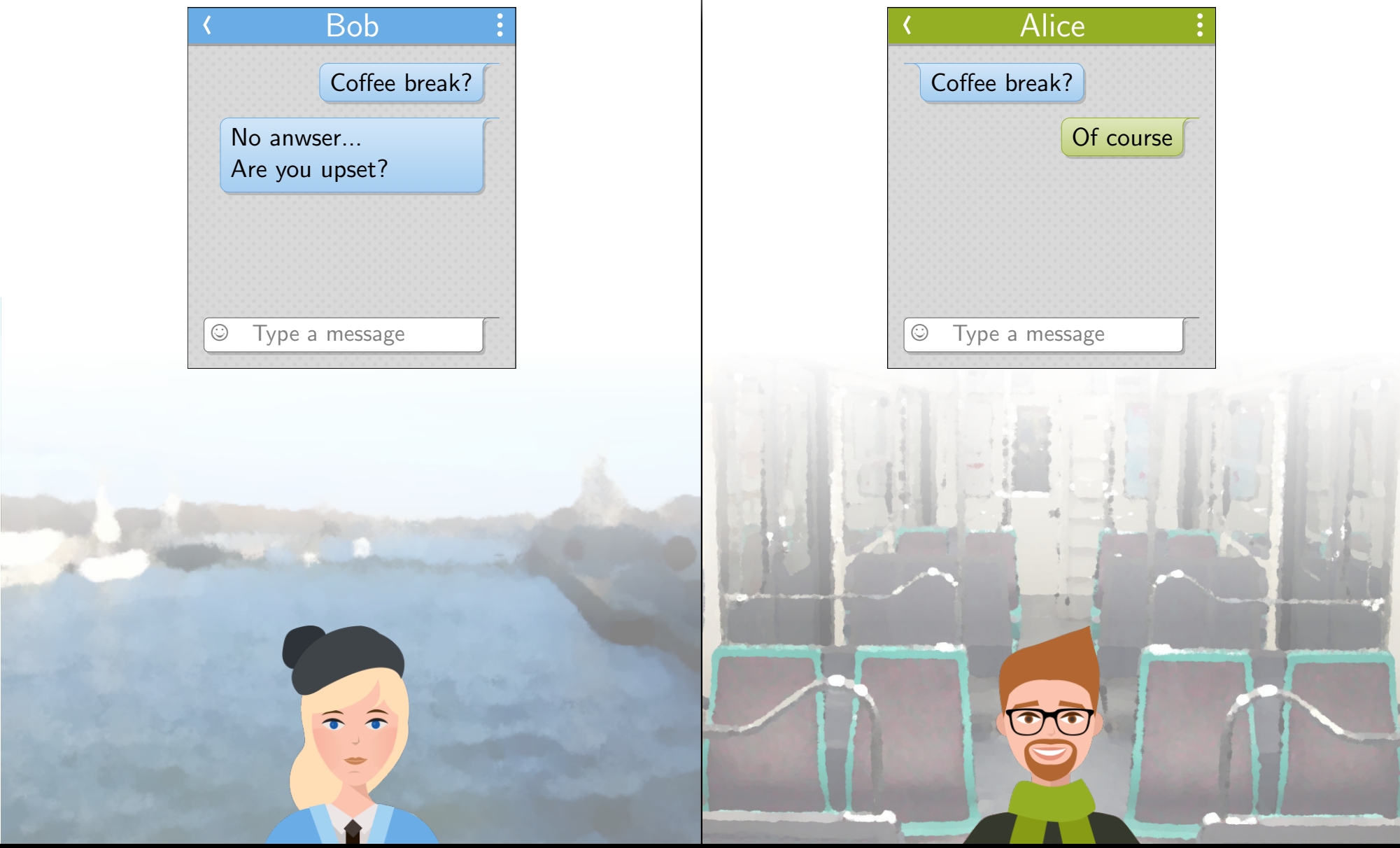
Motivation


Bob
Coffee break?
Type a message
Alice
Coffee break?
Type a message

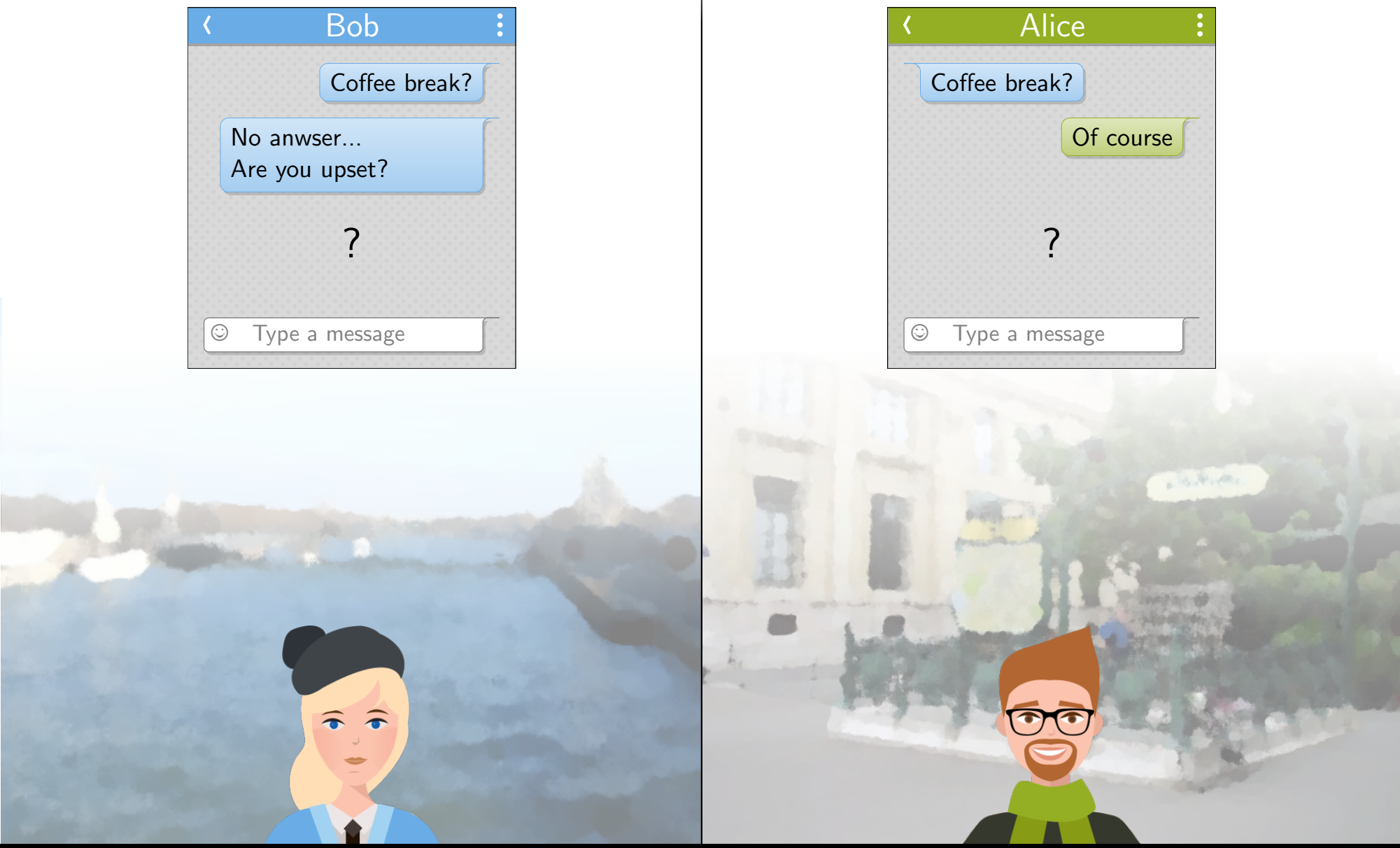
Motivation


Bob
Coffee break?
Type a message
Alice
Coffee break?
Of course
Type a message
Of course

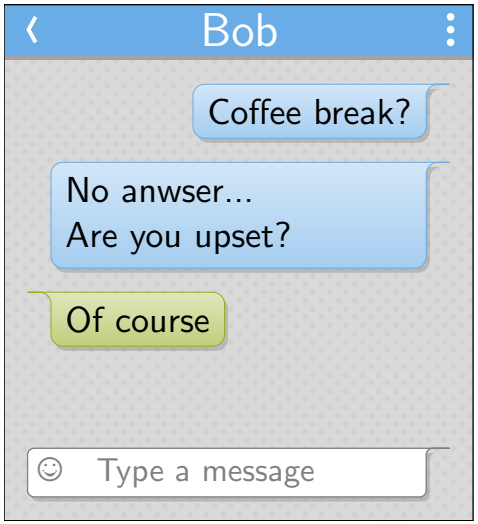
Motivation

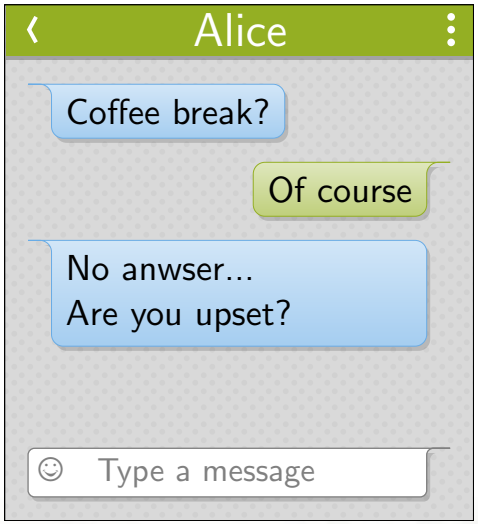

Bob
Coffee break?
No anwser...
Are you upset?
Type a message
Alice
Coffee break?
Of course
Type a message

Motivation

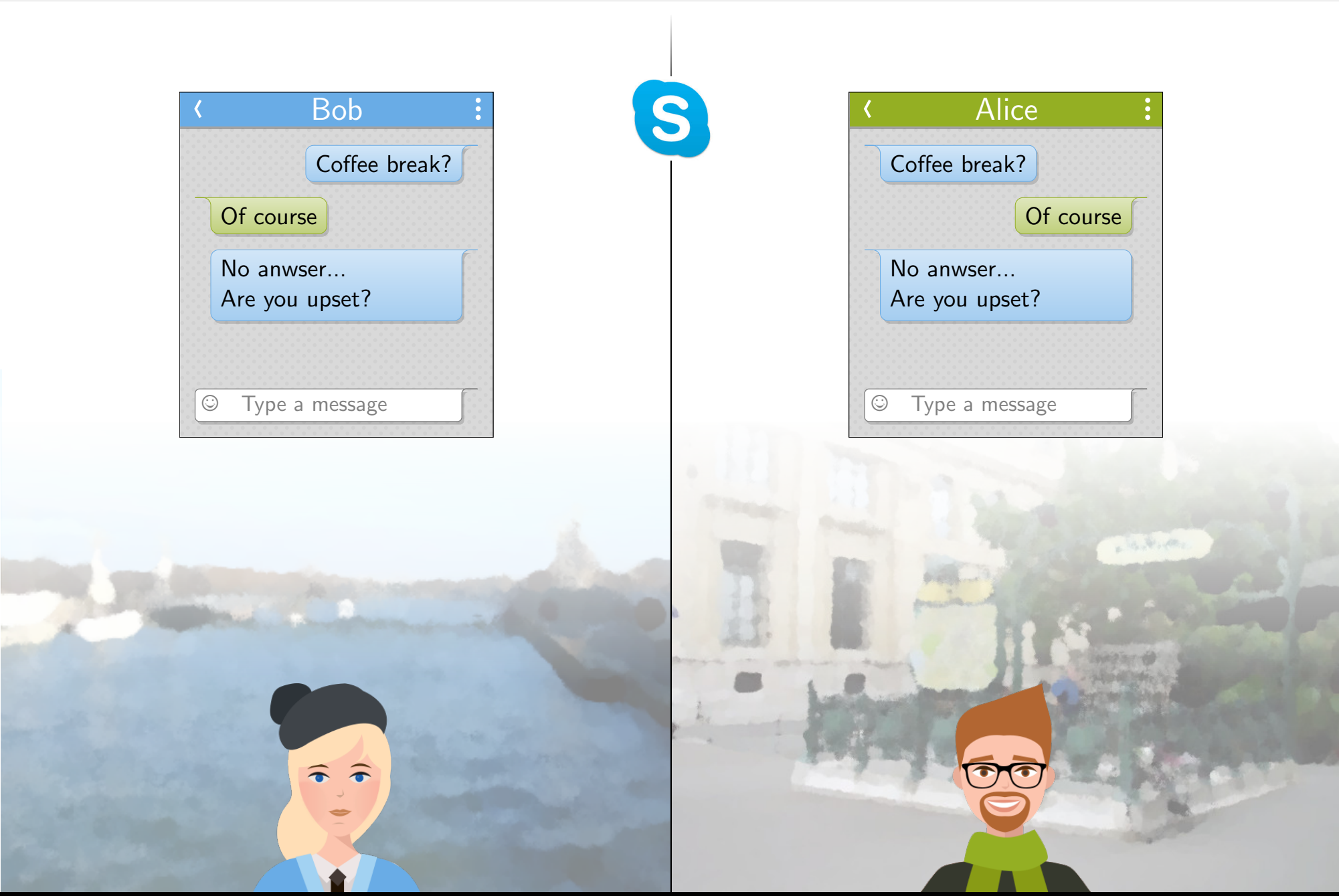

Bob
Coffee break?
No anwser...
Are you upset?
?
Type a message
Alice
Coffee break?
Of course
?
Type a message

Motivation : WhatsApp





Motivation : Skype



Issue

What is the local memory complexity of such eventually consistent objects ?

Current results on eventually consistent shared objects:

- Sets ($\mathcal{O}(n \log(m))$), Counters ($\mathcal{O}(n)$), Registers ($\mathcal{O}(\log(m))$), Multi-value Registers ($\mathcal{O}(n \log(m))$) [1].
- Data Stores (Sets ($\mathcal{O}(n \log(m))$), Multi-value Registers ($\mathcal{O}(n \log(m))$)) [2].
- Collaborative Editors ($\mathcal{O}(m)$) [3].

[1] Burckhardt S, Gotsman Al, Yang H, Zawirski M : Replicated data types: specification, verification, optimality
[2] Attiya H, Ellen F, Morrison A : Limitations of Highly-Available Eventually-Consistent Data Stores
[3] Attiya H, Burckhardt S, Gotsman A, Morrison A, Yang H, Zawirski M : Specification and complexity of collaborative text editing

The Operational Model

Hypothesis: Restrictive on the type of algorithms

- Objects are fully replicated
- Read operations are local
- Messages can only be sent during update operations

Remark:

- Optimal in number of messages used

The Wait-Free Model

Processes

- Asynchronous : no bounds on the execution time
- May crash : no active waiting possible

Communication

- Message passing
- Asynchronous : no bounds on the delay of message transmition

Problem

Are the wait-free model and the operational model equivalent in terms of complexity ?

Approach

Update consistent shared objects:

Convergence state must be obtainable in a sequential execution.

Approach

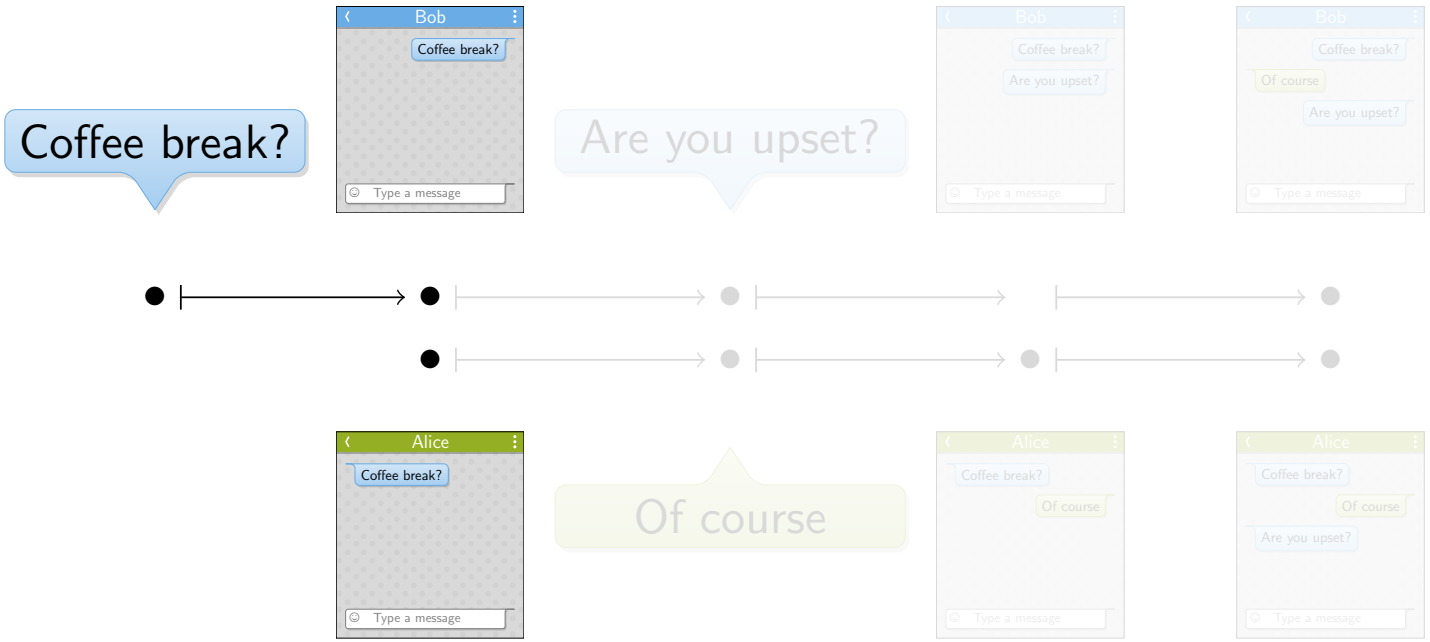
Update consistent shared objects:

Convergence state must be obtainable in a sequential execution.

Approach

Update consistent shared objects:

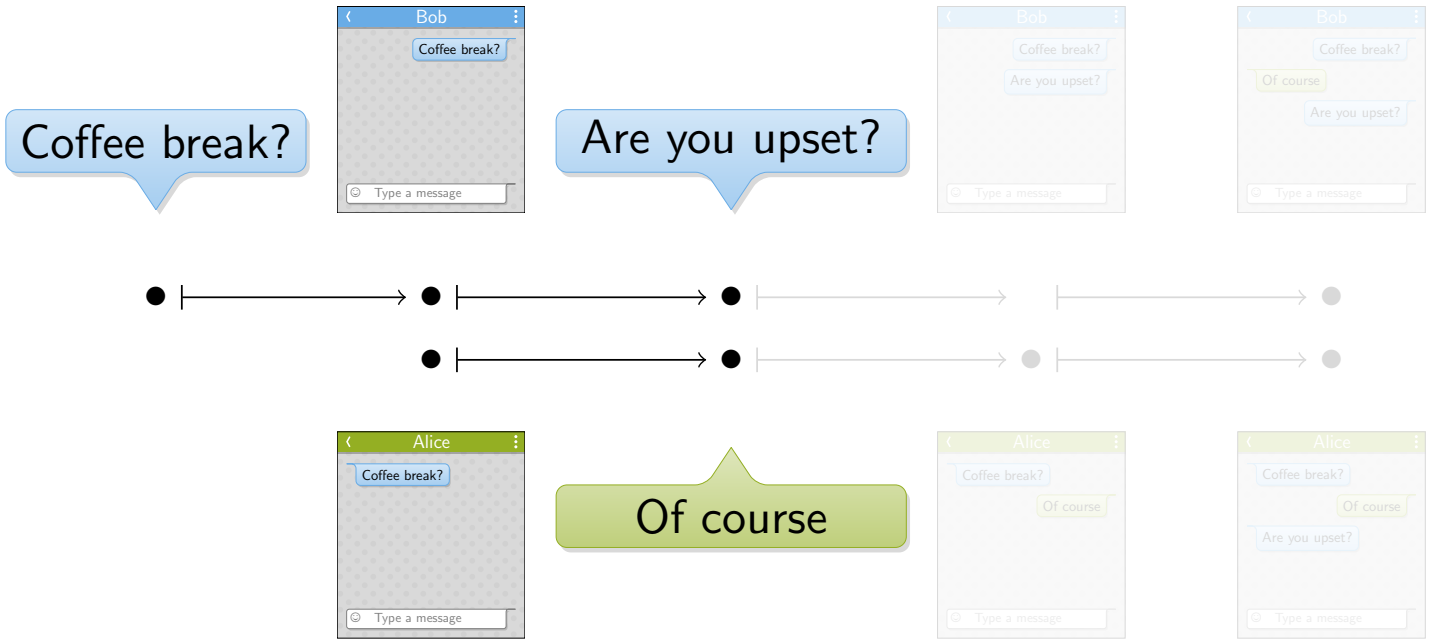
Convergence state must be obtainable in a sequential execution.



Approach

Update consistent shared objects:

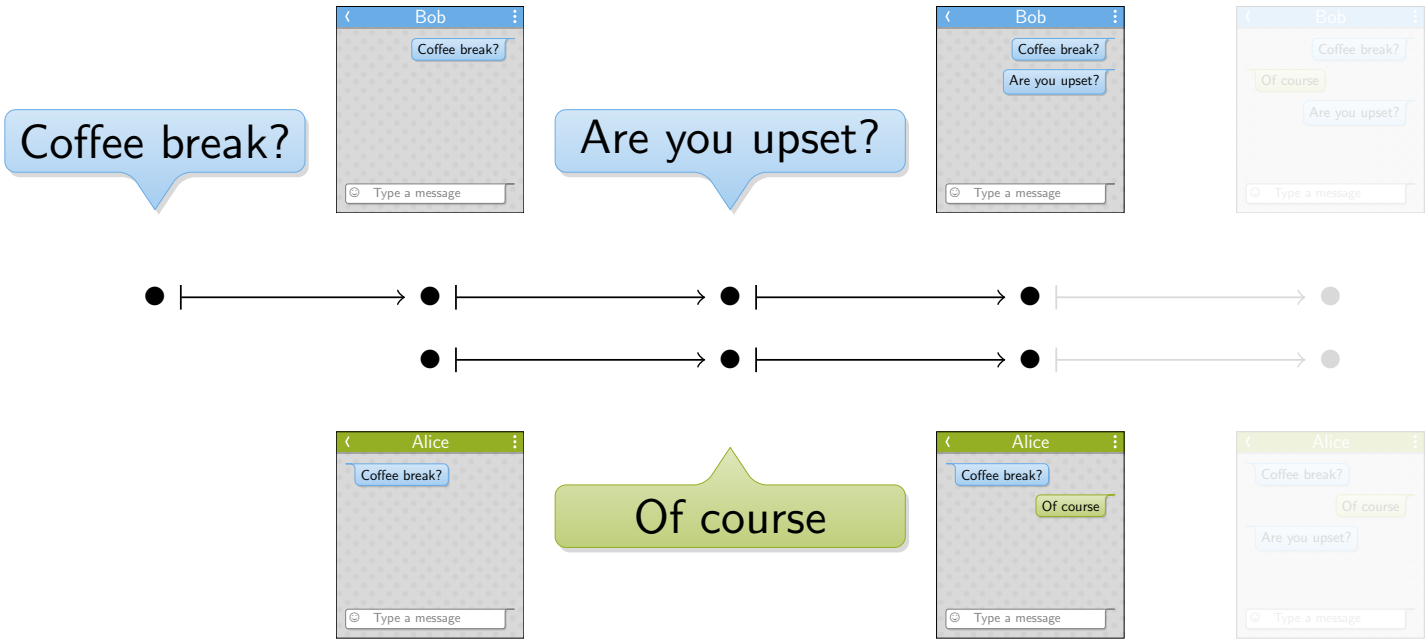
Convergence state must be obtainable in a sequential execution.



Approach

Update consistent shared objects:

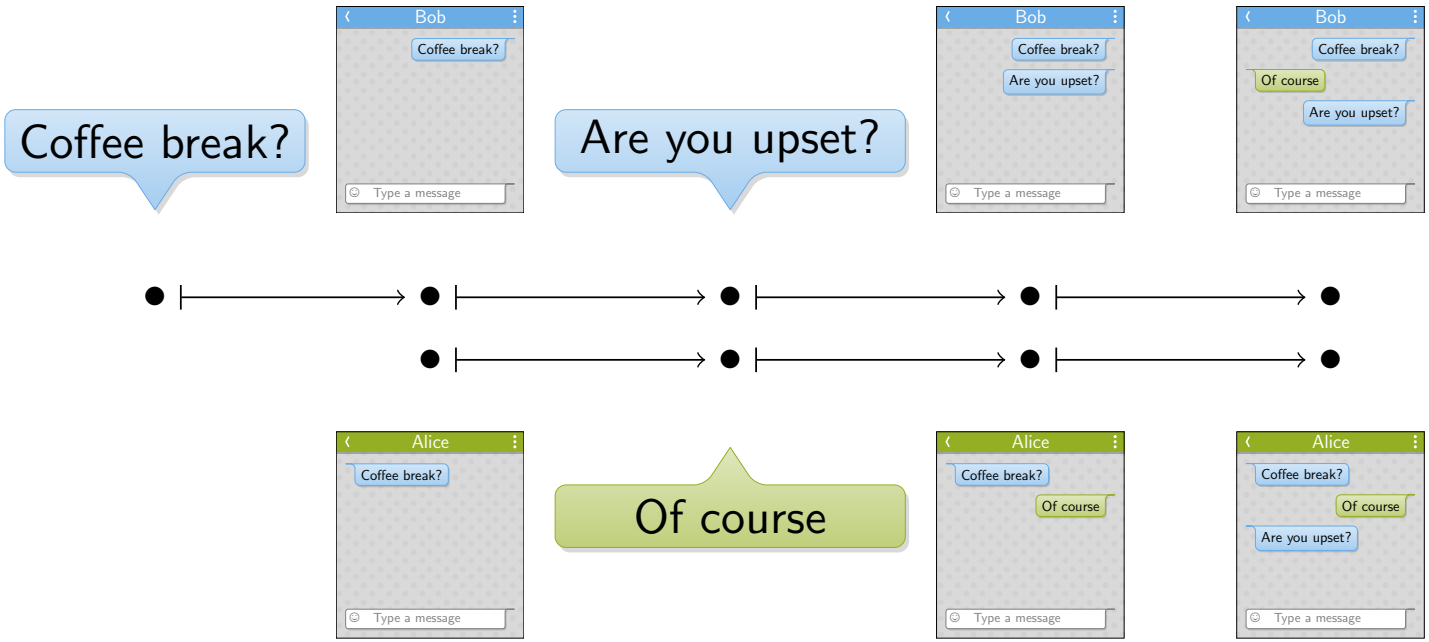
Convergence state must be obtainable in a sequential execution.



Approach

Update consistent shared objects:

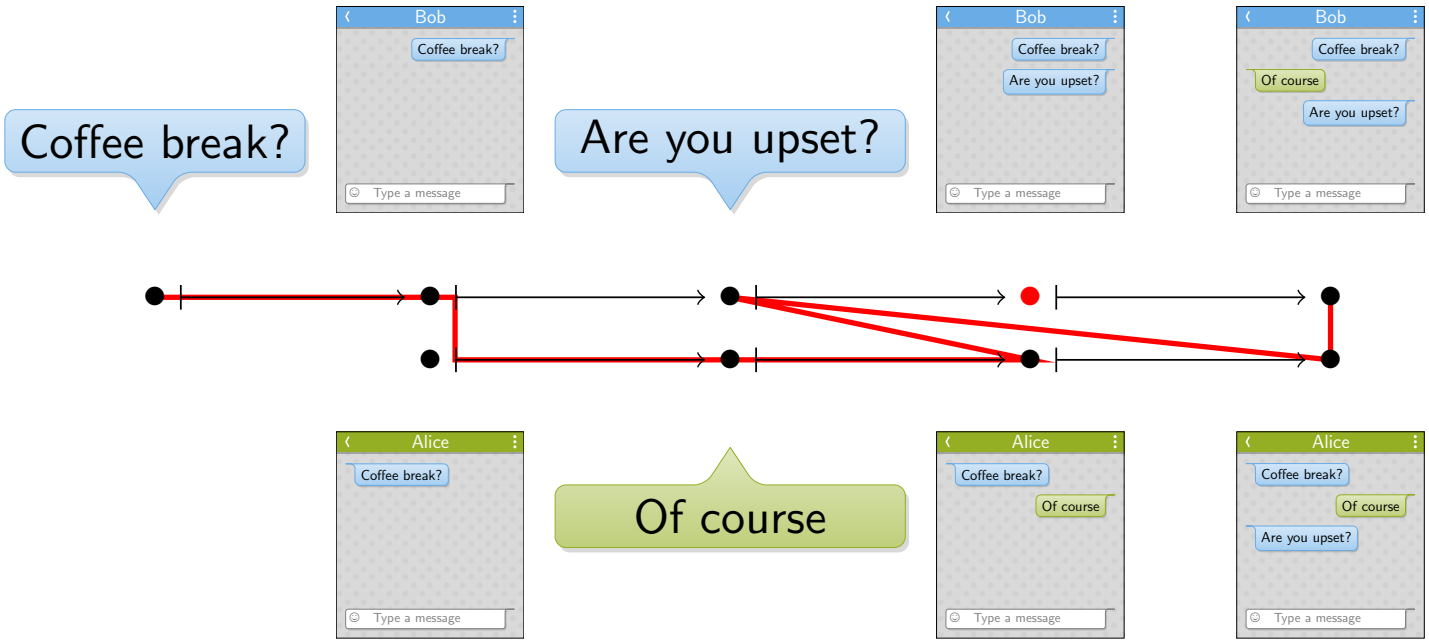
Convergence state must be obtainable in a sequential execution.



Approach

Update consistent shared objects:

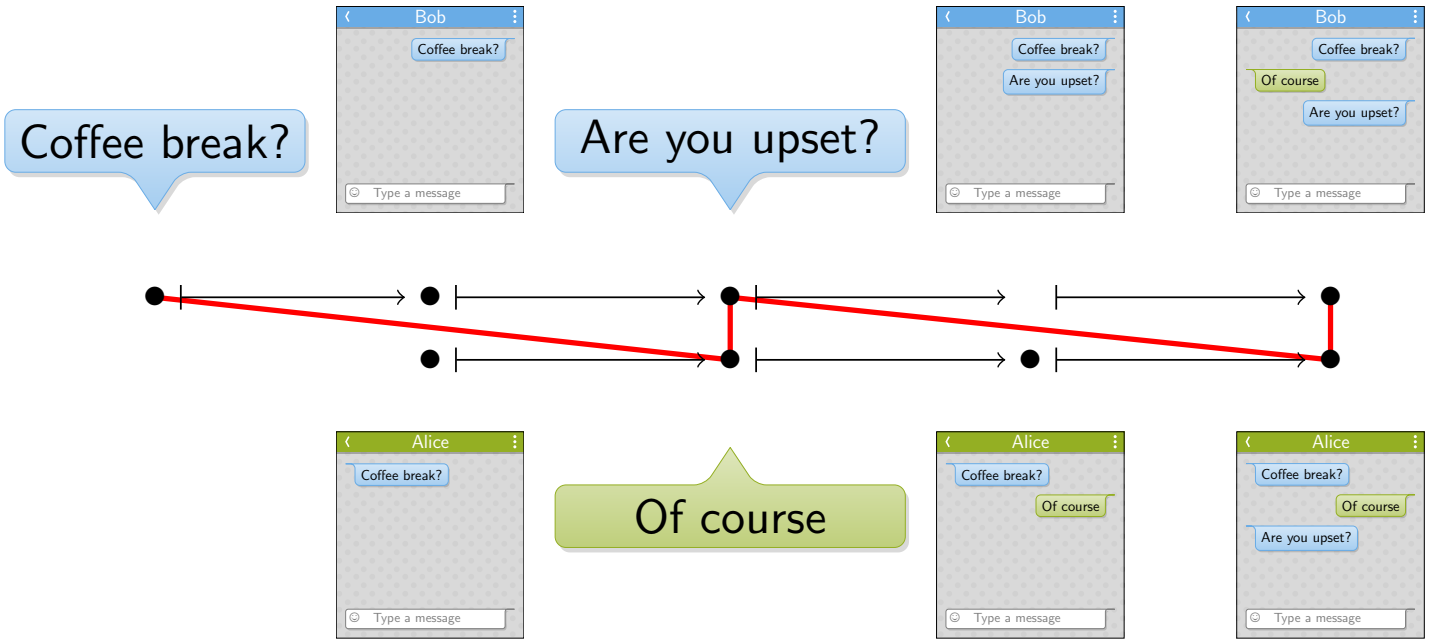
Convergence state must be obtainable in a sequential execution.



Approach

Update consistent shared objects:

Convergence state must be obtainable in a sequential execution.



Contribution

Theorem

We prove that there is:

- One object O
- One execution E
- One implementation of O in the Wait-Free model I_{WF}

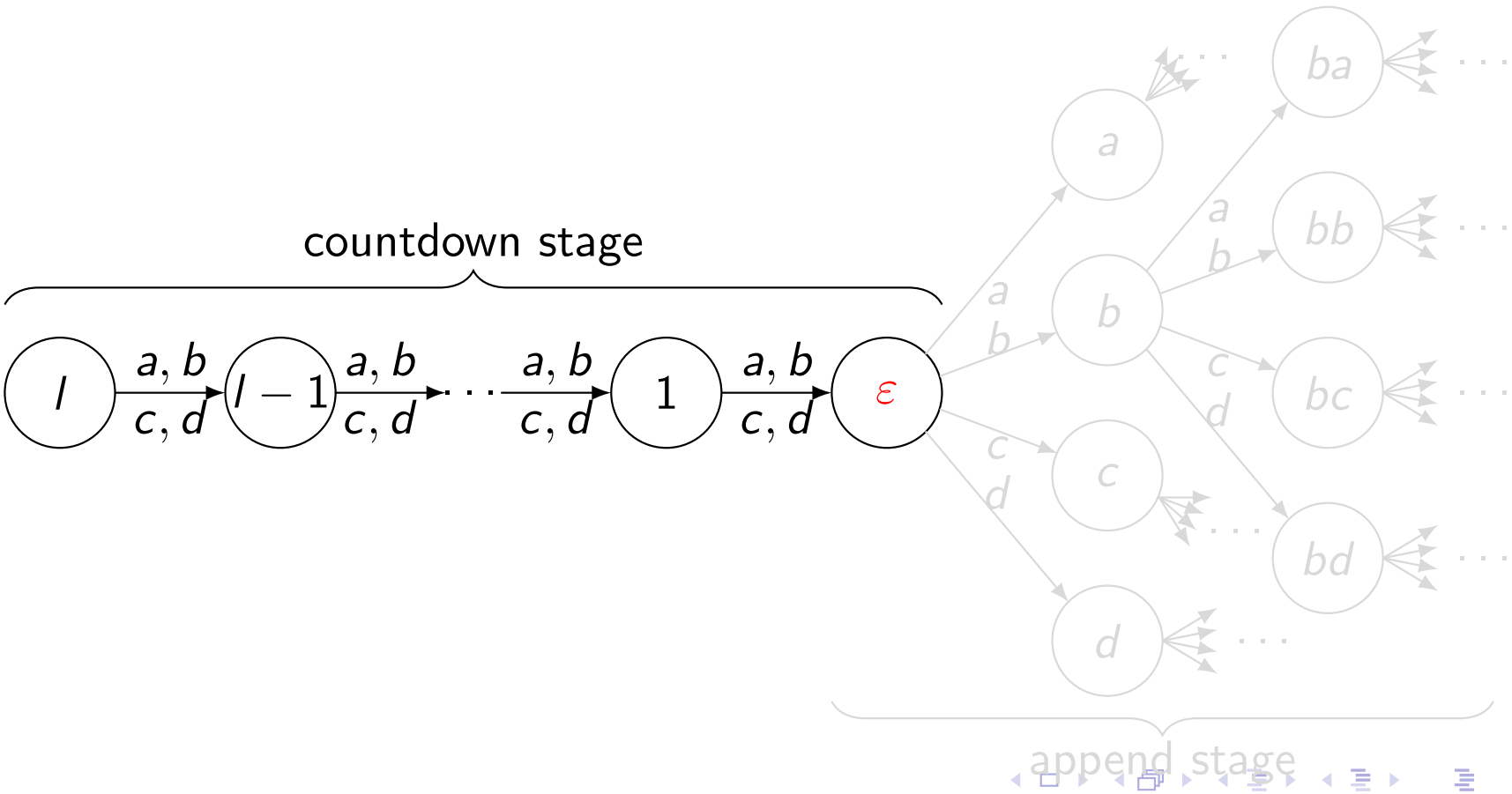
Such that:

Any implementation of O in the operational model I_{OM} takes strictly more bits of local memory than I_{WF} in E.

The Object : l-Countdown-Append

Four update operations, one query operation, two phases:

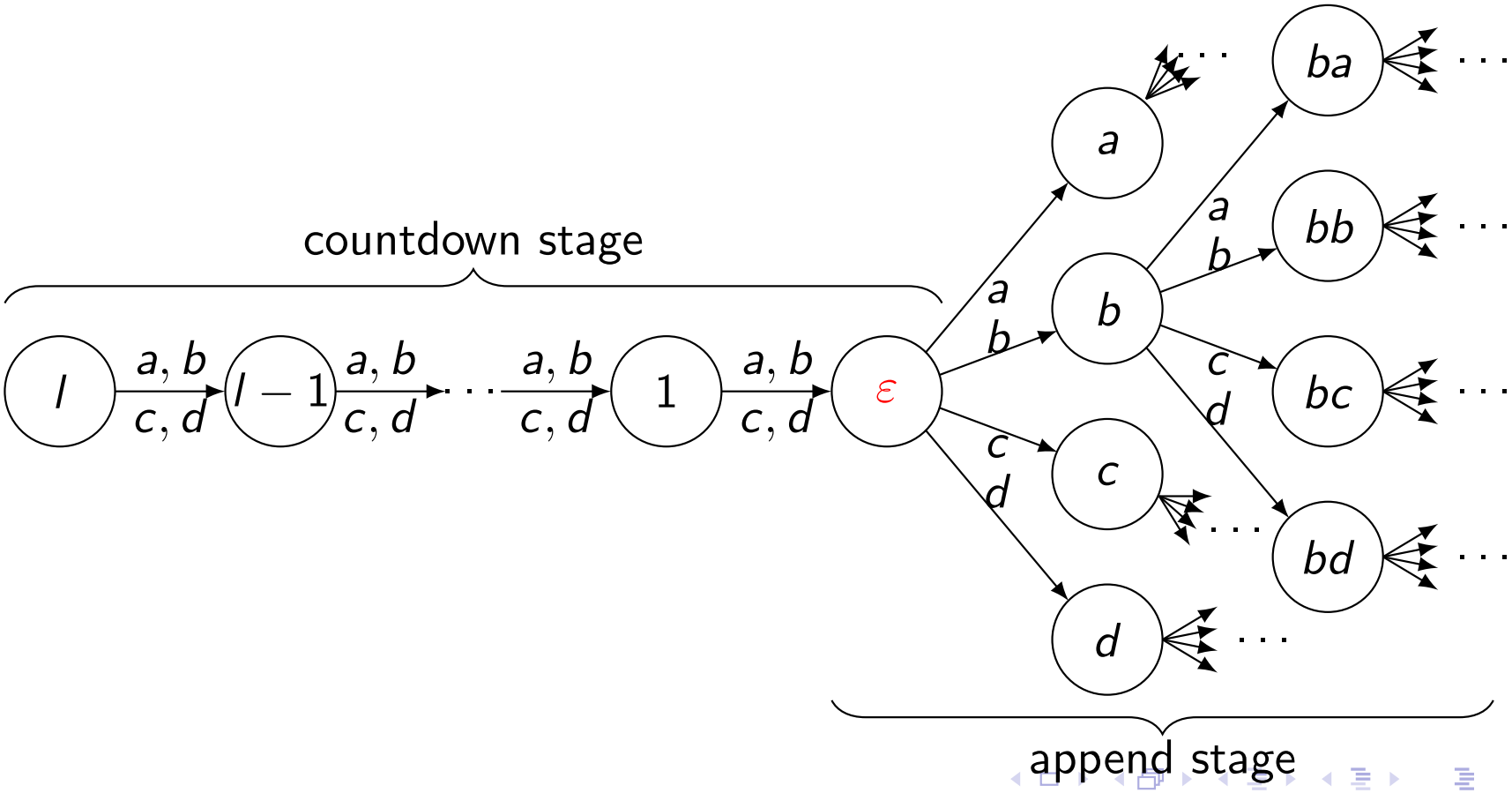
1. A countdown phase: the operations decrement a size l counter
2. An append phase : the operations are appended to a log



The Object : l-Countdown-Append

Four update operations, one query operation, two phases:

1. A countdown phase: the operations decrement a size l counter
2. An append phase : the operations are appended to a log

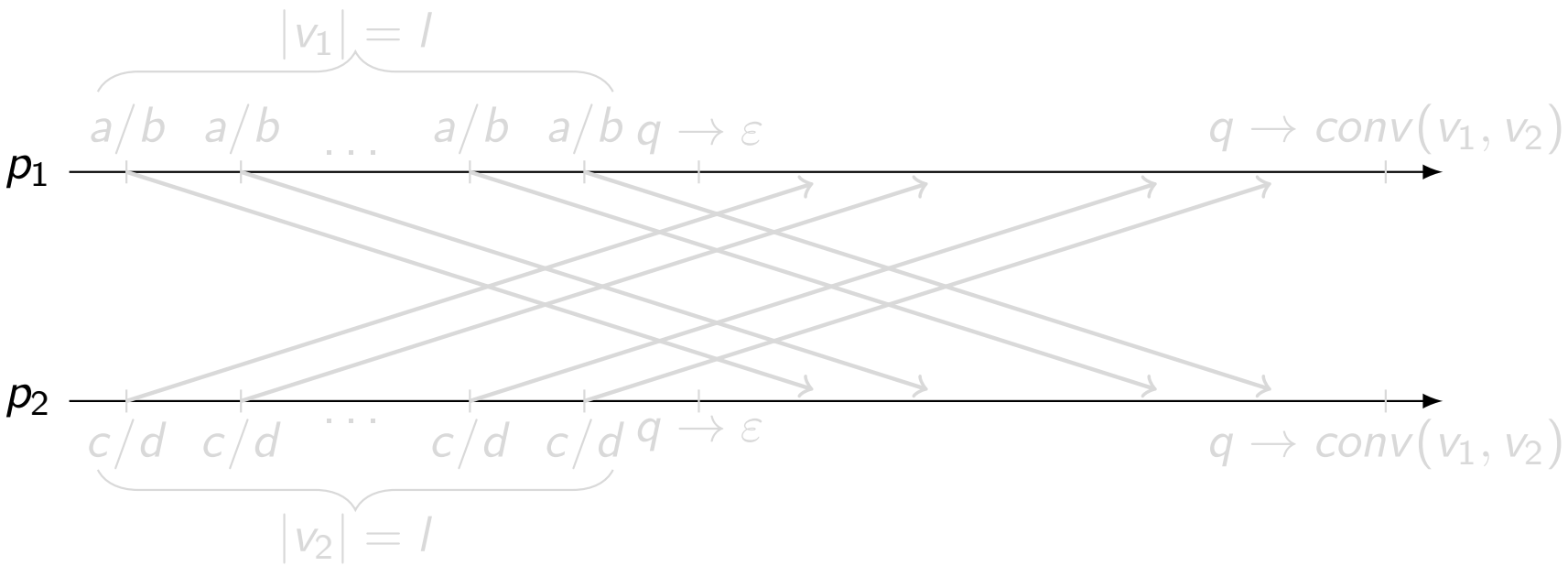


Lower Bound in the Operational Model

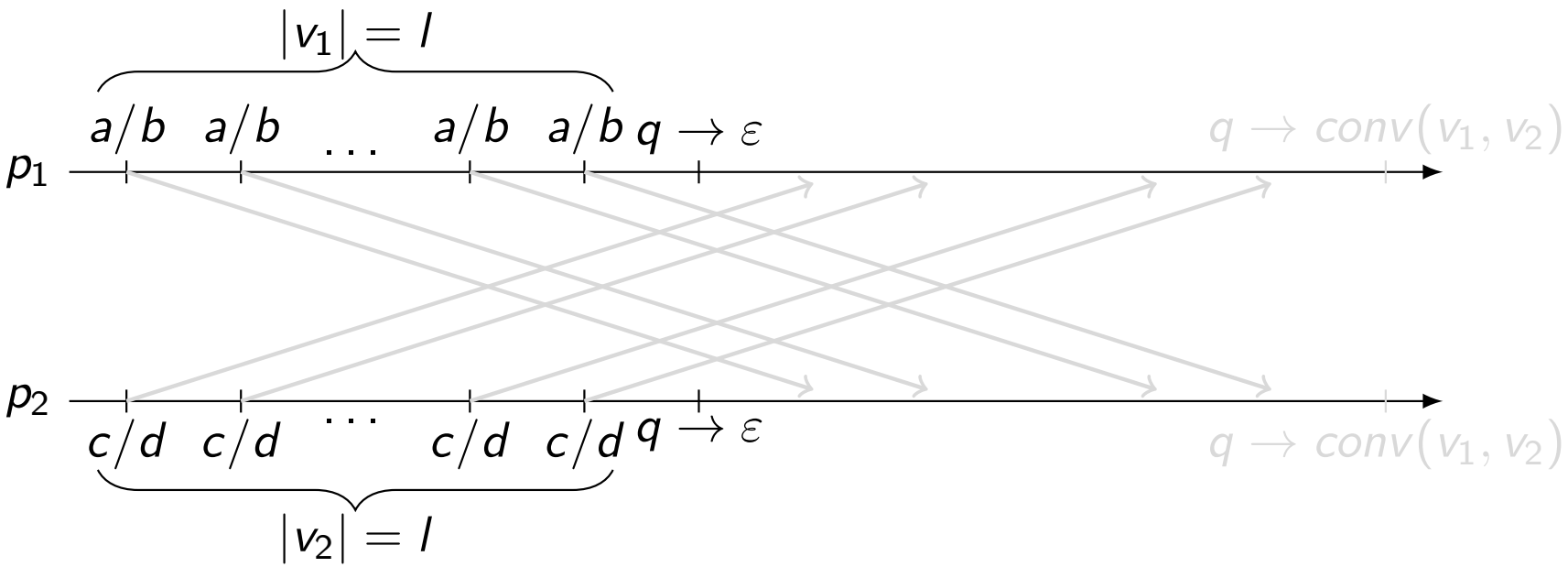
Lemma

There is an execution X n which any implementation of the l-CA in the operational model requires at least $\left(\frac{l}{2} - 1\right)$ bits of local memory in the ε state.

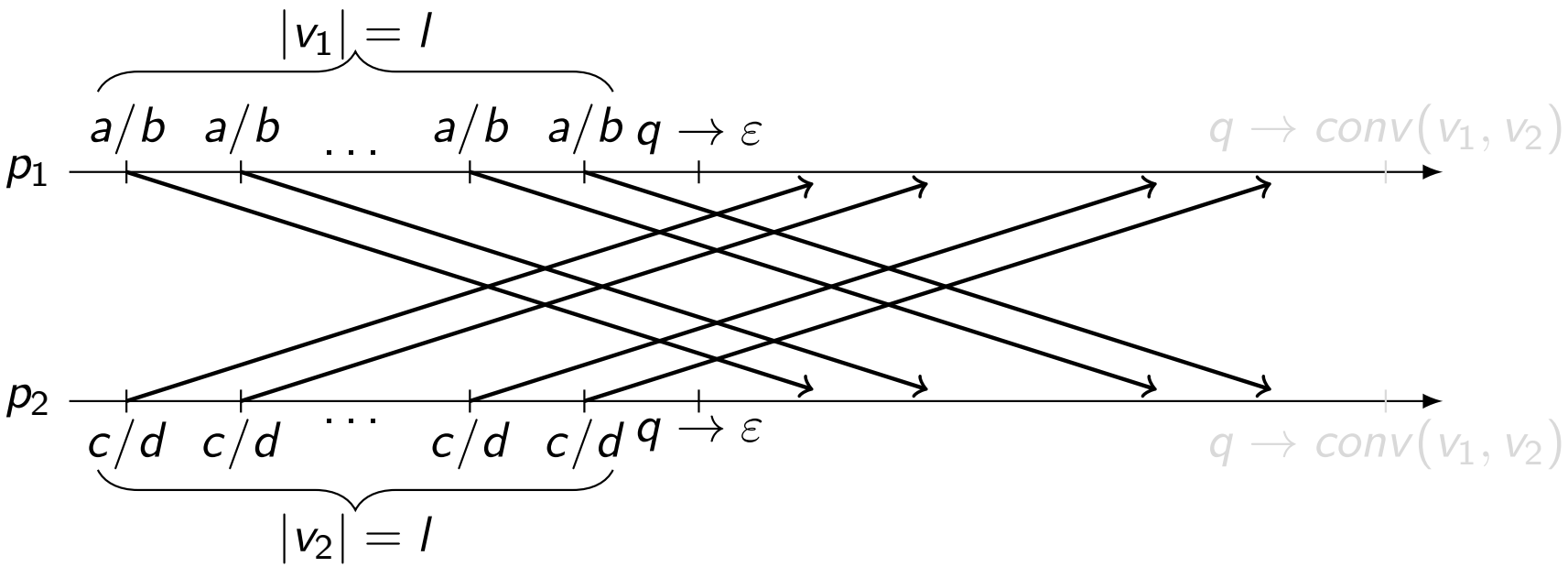
The Execution : $X_{v_1}^{v_2}$



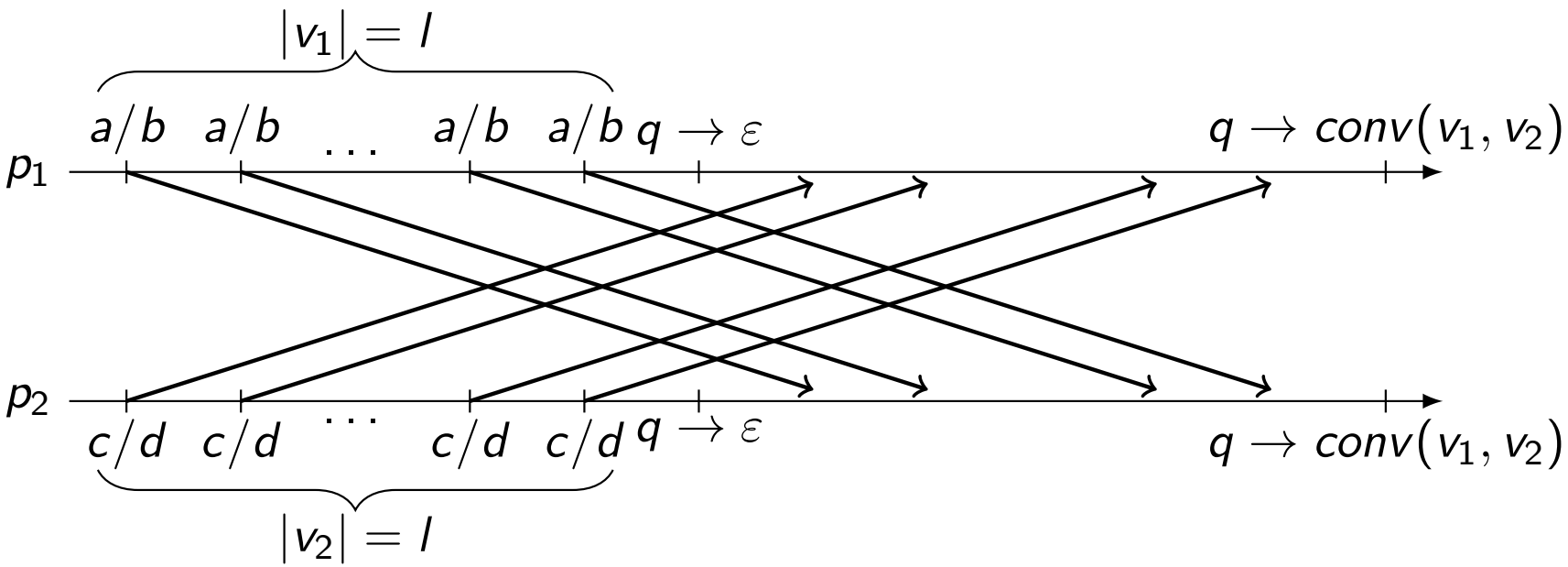
The Execution : $X_{v_1}^{v_2}$



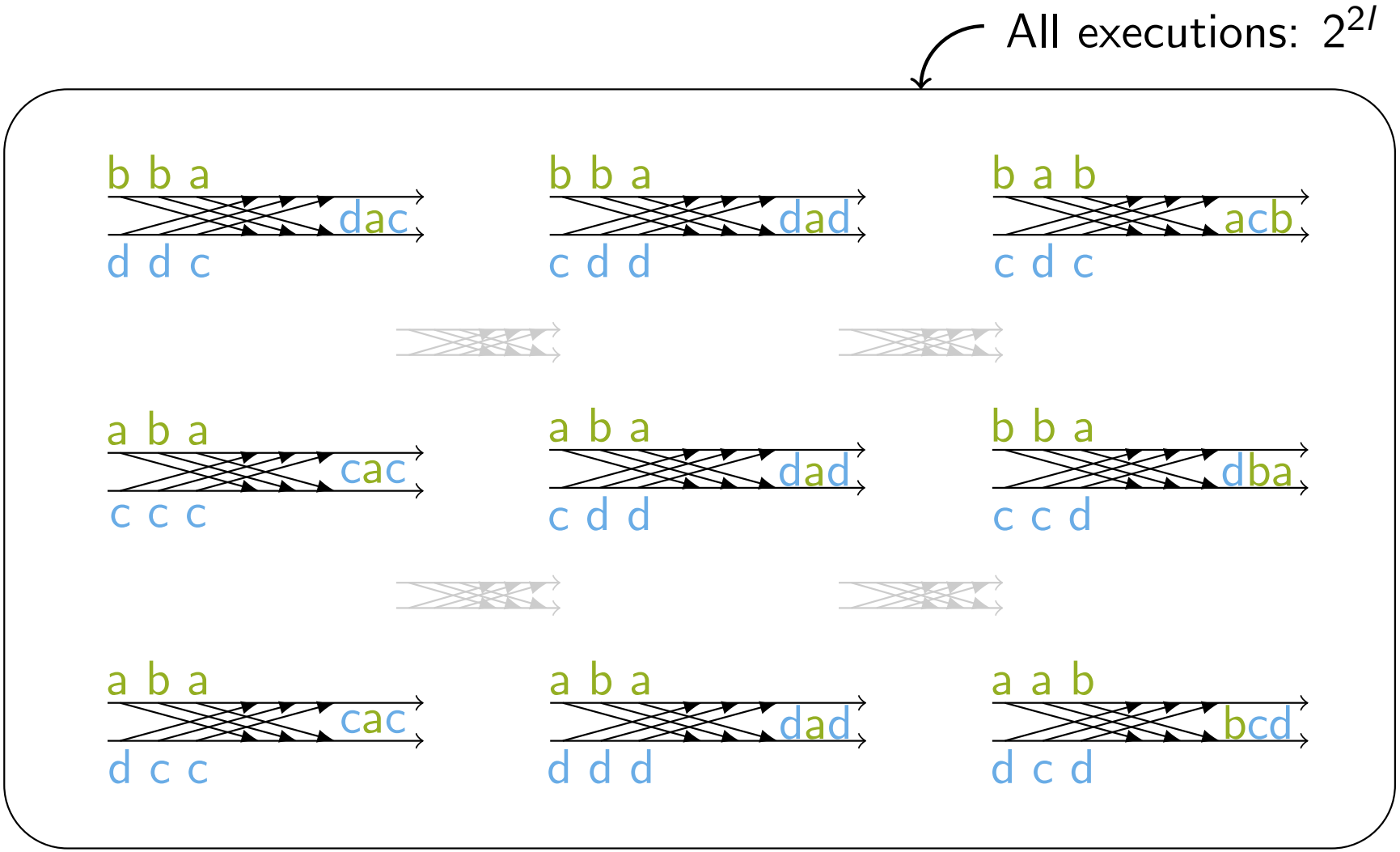
The Execution : $X_{v_1}^{v_2}$



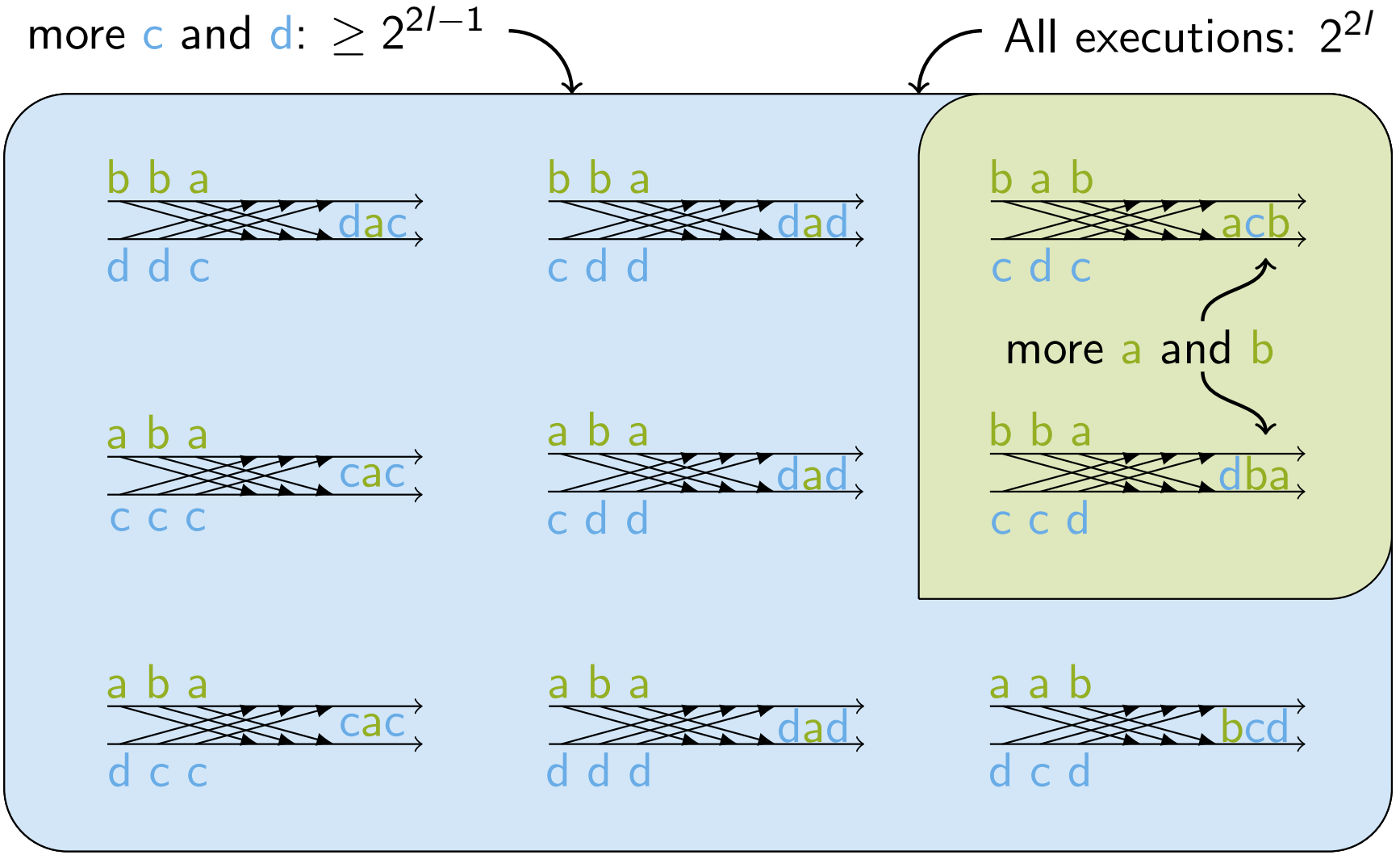
The Execution : $X_{v_1}^{v_2}$



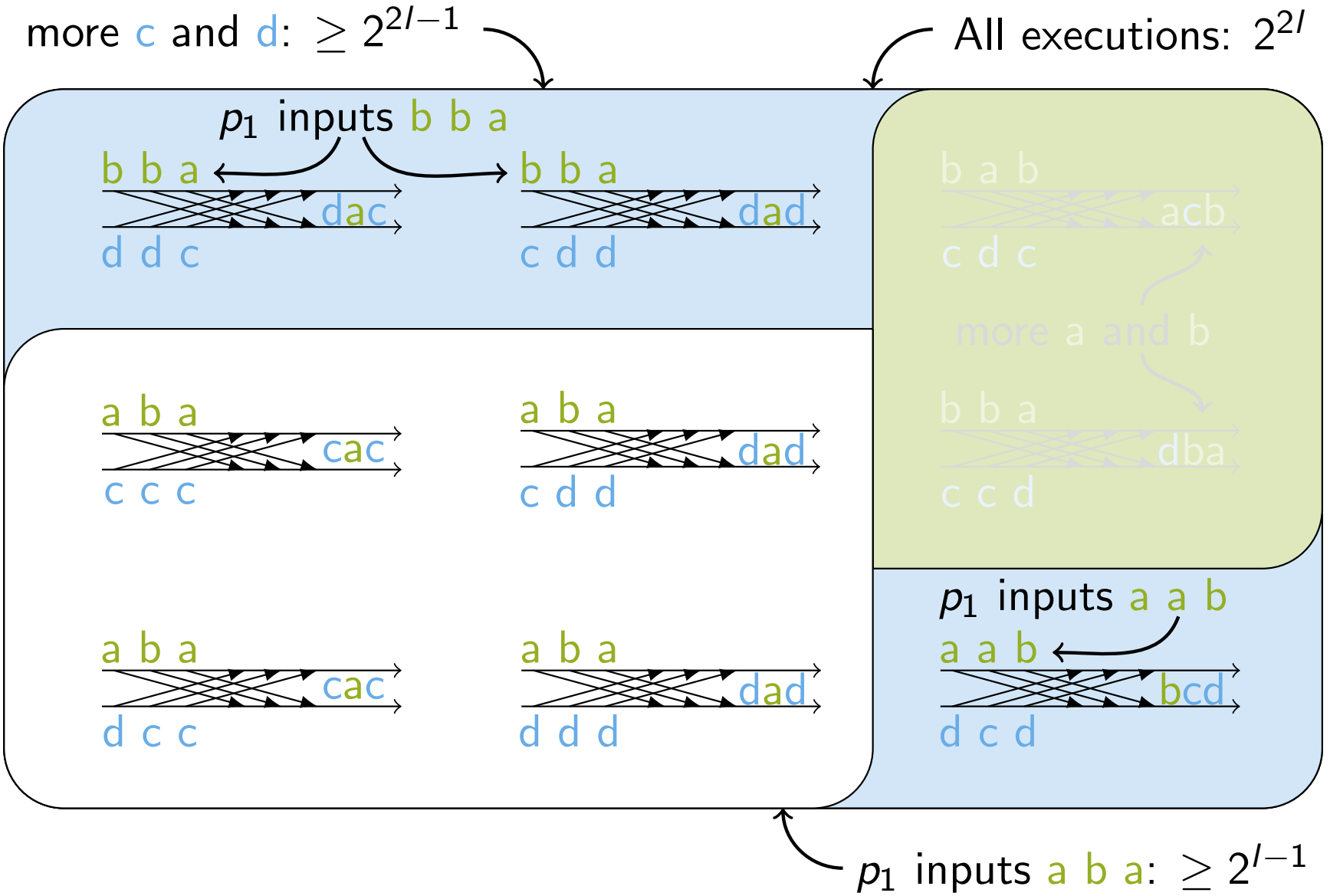
Let us count the executions



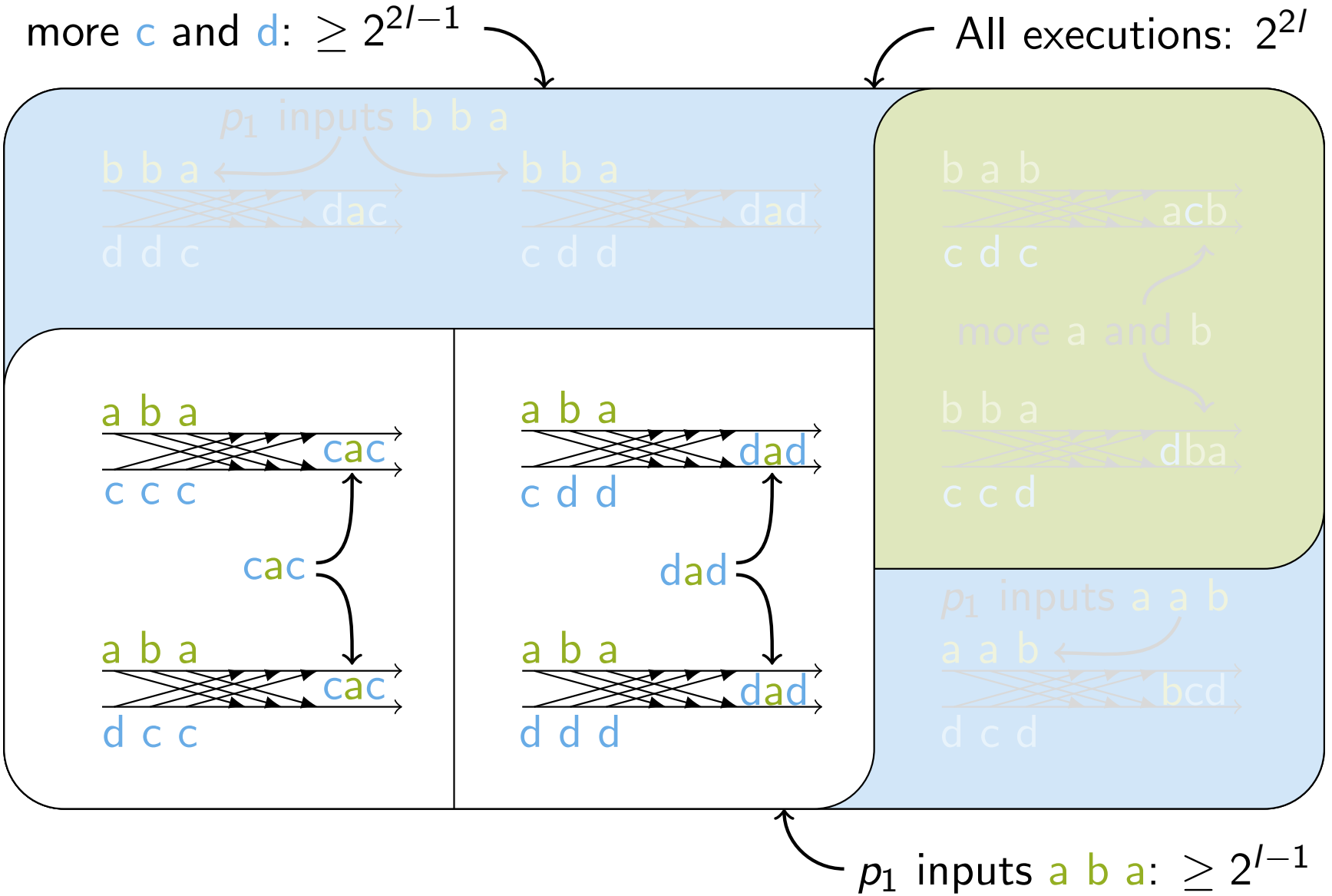
Let us count the executions



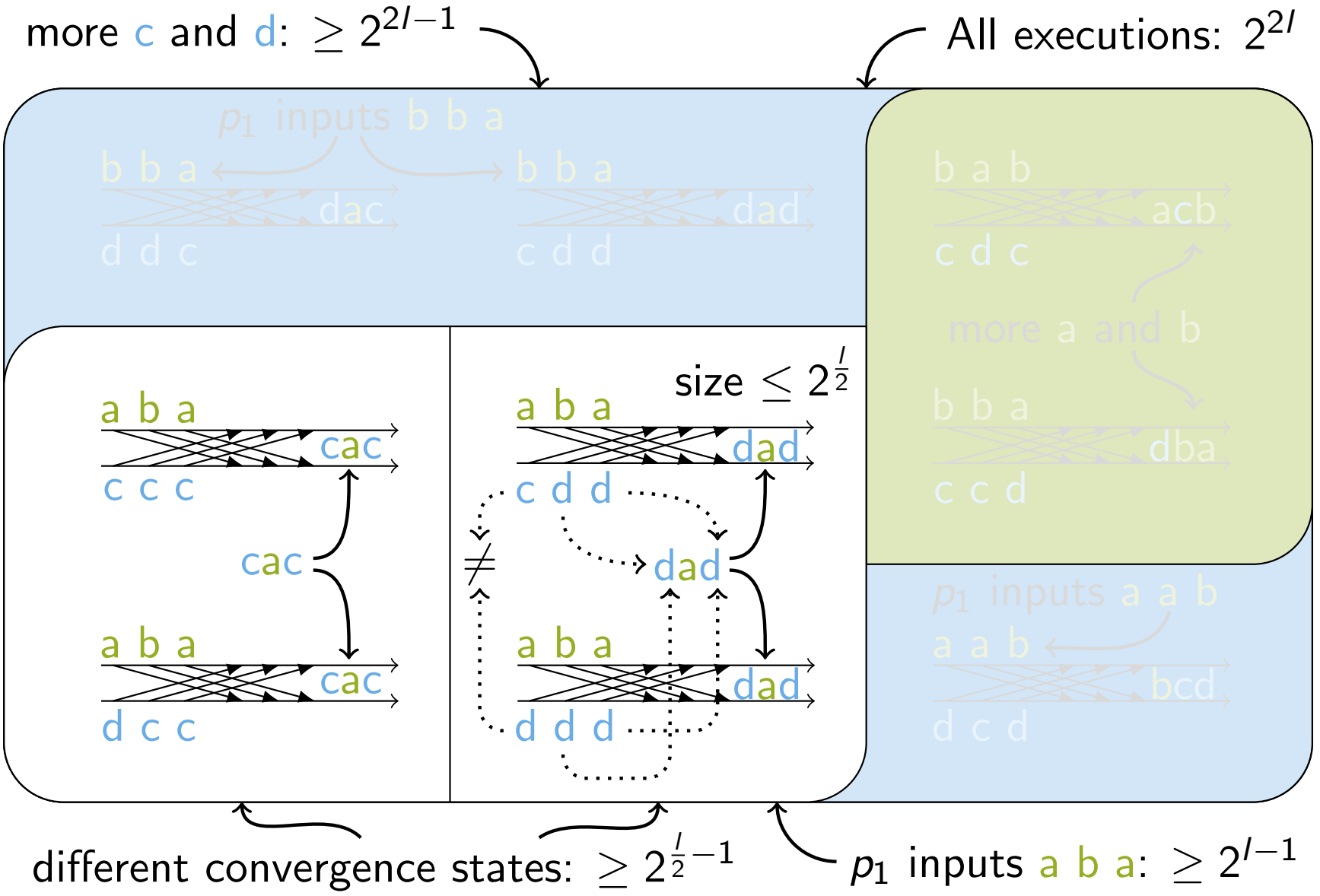
Let us count the executions



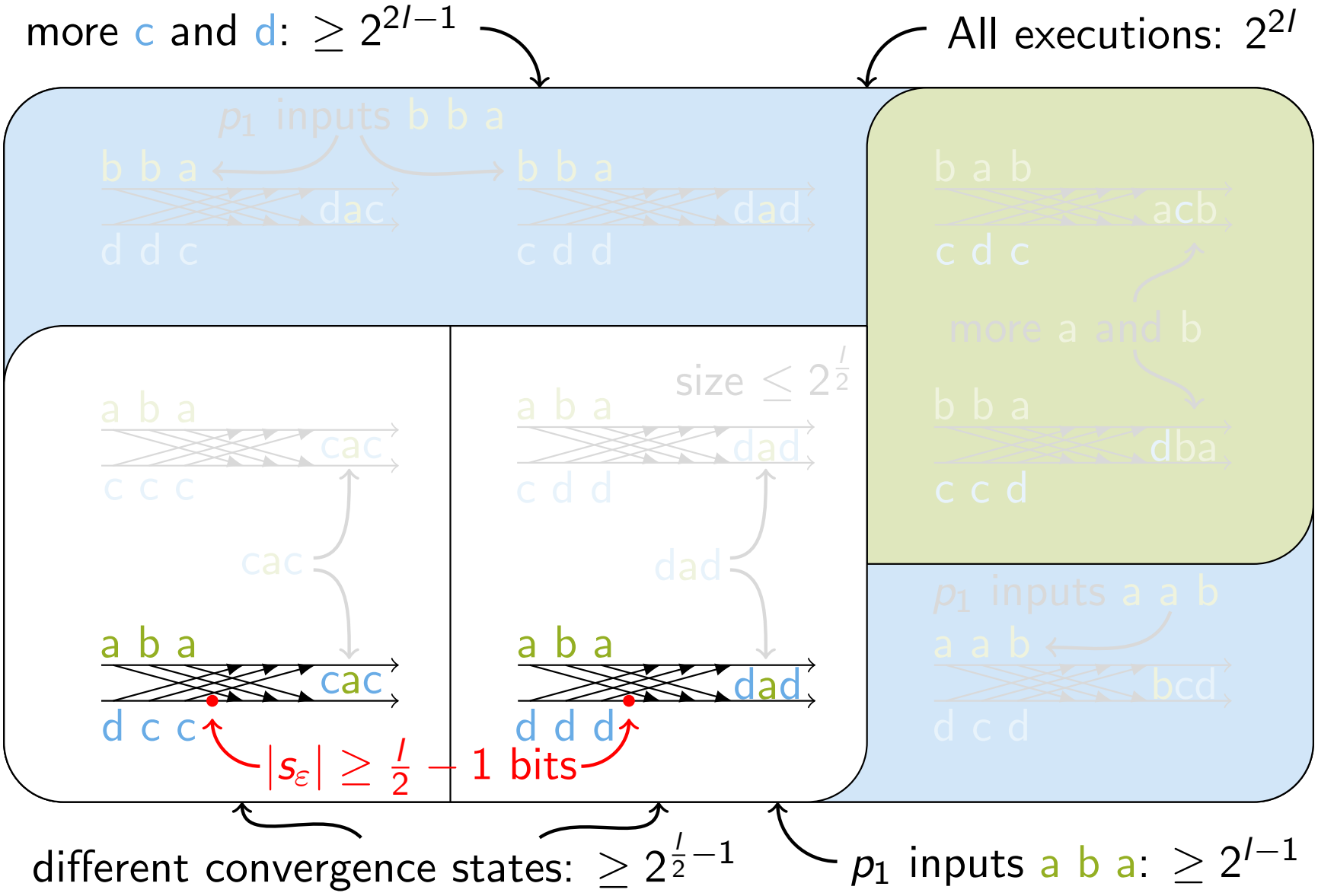
Let us count the executions



Let us count the executions



Let us count the executions

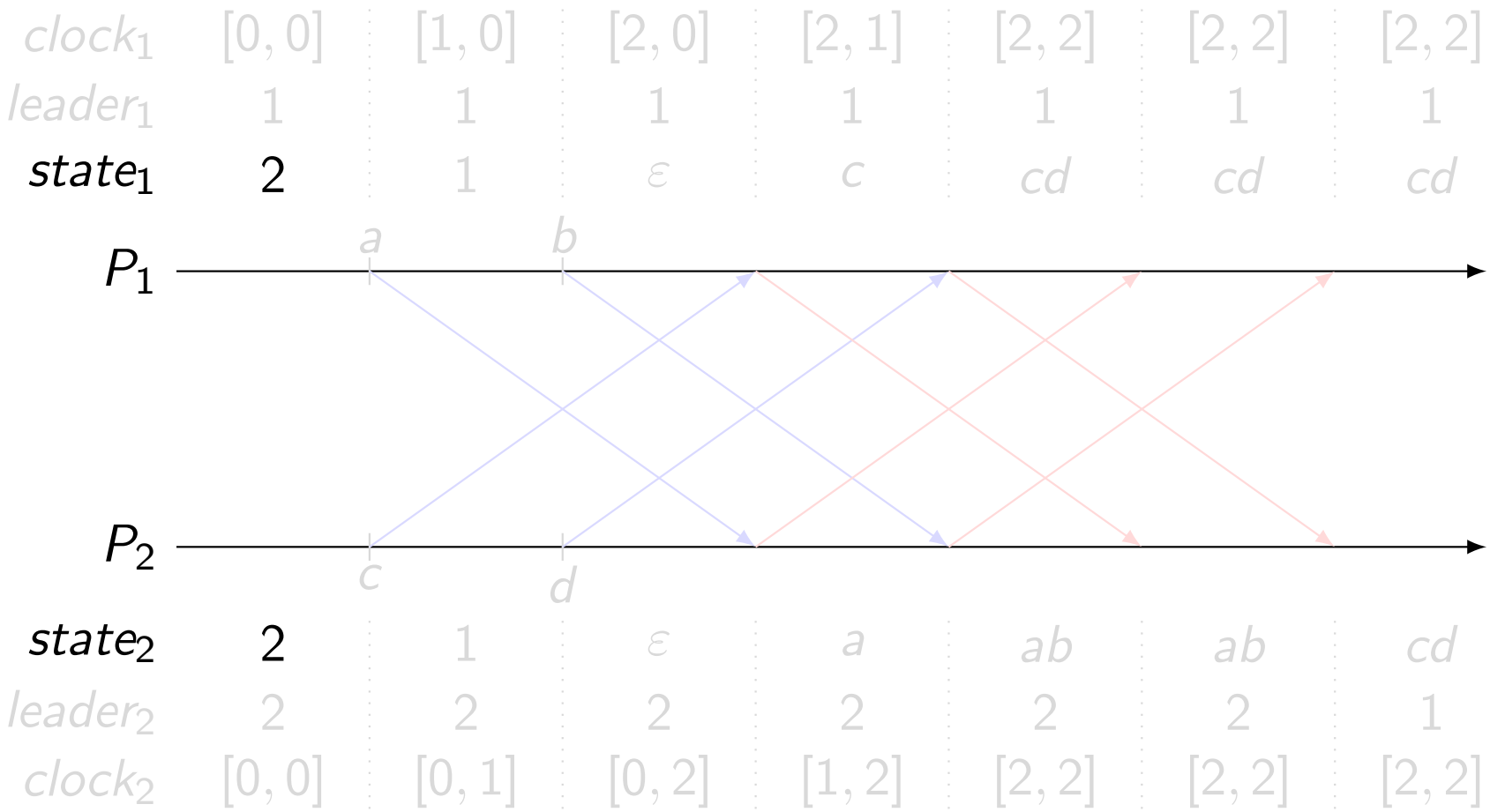


Complexity of the Wait-Free Implementation

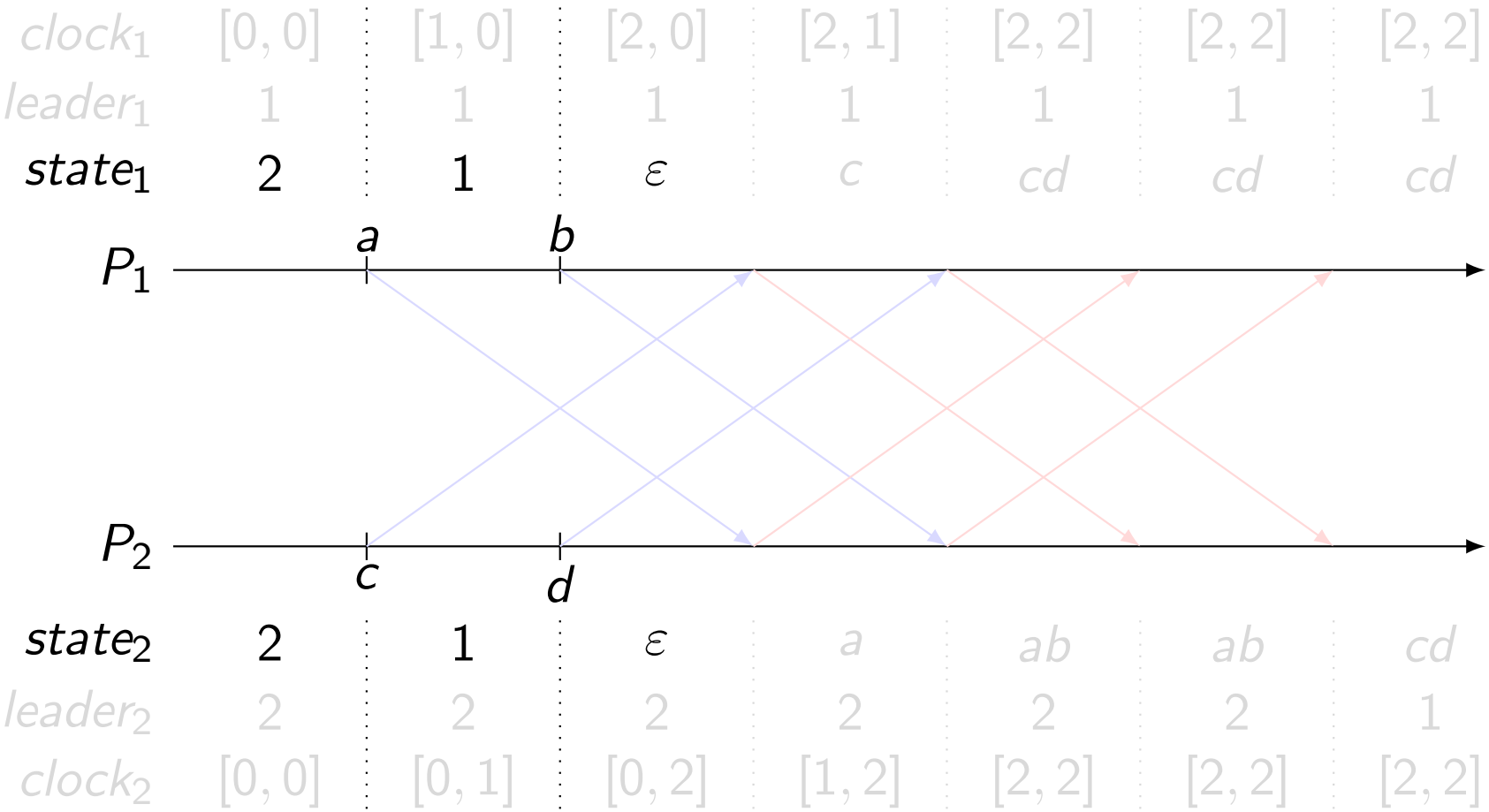
Lemma

The algorithm has a local memory complexity of $\mathcal{O}(n \log(nl))$ for the X execution.

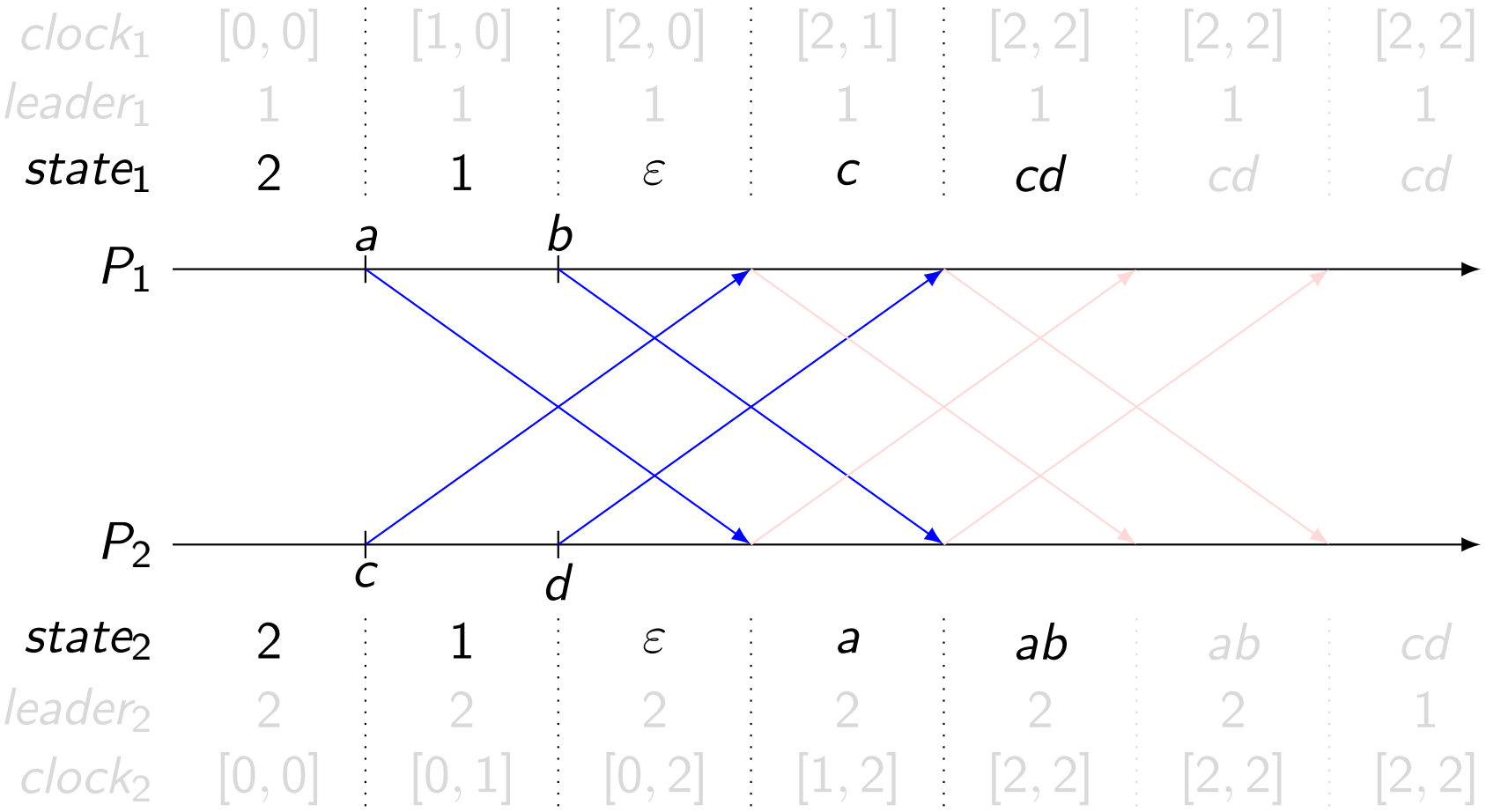
Implementation of the l-Countdown-Append in the Wait-Free Model



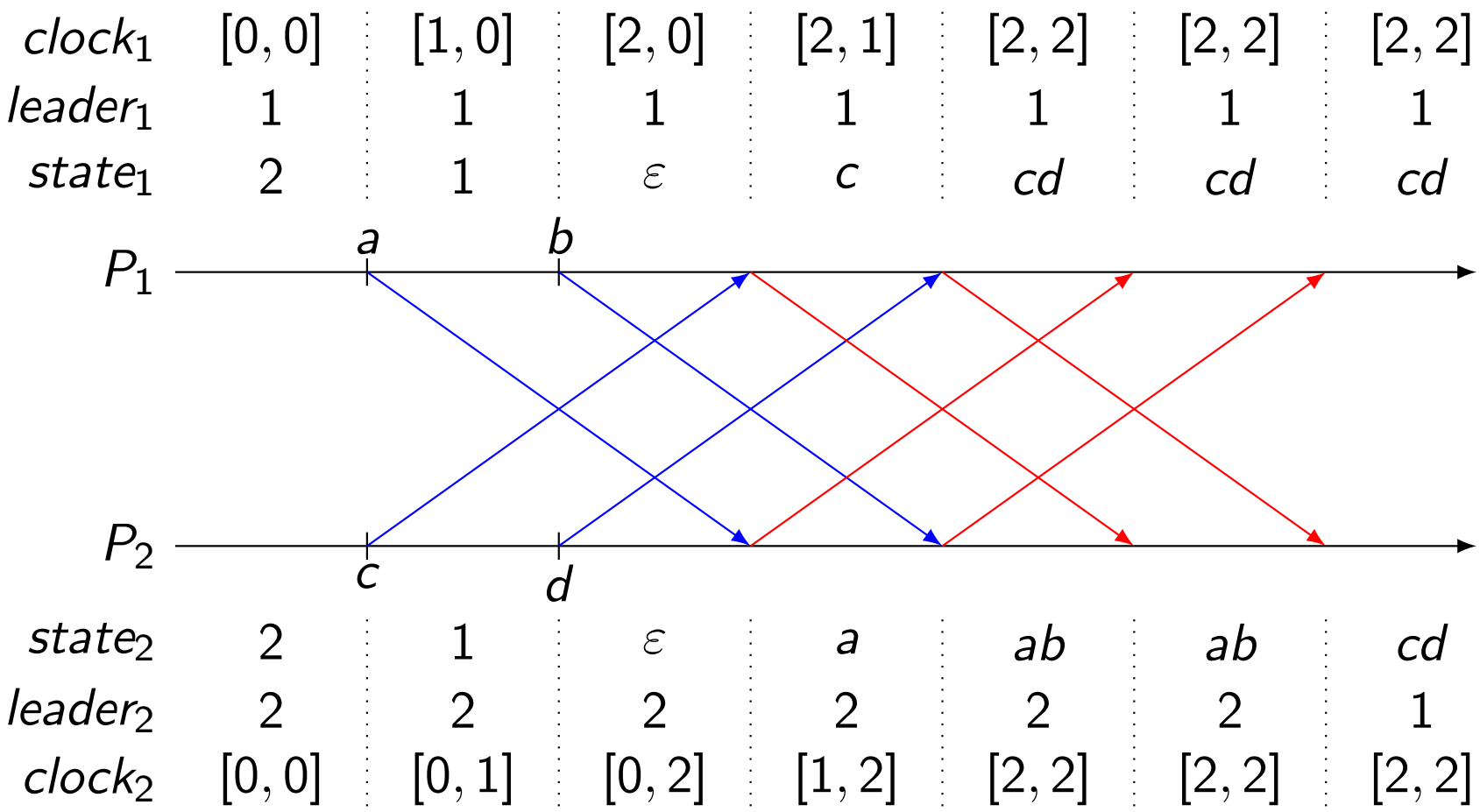
Implementation of the l-Countdown-Append in the Wait-Free Model



Implementation of the l-Countdown-Append in the Wait-Free Model



Implementation of the l-Countdown-Append in the Wait-Free Model



Conclusion

Results: Local Memory Complexity

1. Operational model: $\mathcal{O}(I)$
2. Wait-free model: $\mathcal{O}(n \log(I))$

The Operational Model does not allow the optimal implementation for Update Consistency.