Time Is Not a Healer, but It Sure Makes Hindsight 20:20

Giuliano Losa, Stellar Development Foundation
Eli Gafni, UCLA

We present a new, simple proof of the FLP impossibility theorem

The proof in a nutshell

1. Using a simulation, we reduce the problem to the synchronous model with message-omission failures of Santoro and Widmayer
2. Each round of the synchronous model, we identify a process that can impose a decision but fails to do so (not FLP bi-valency)

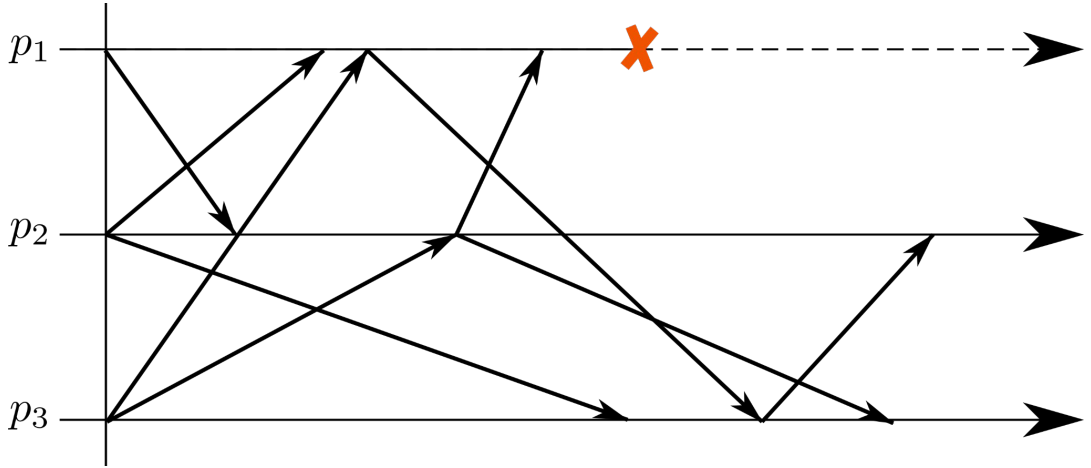
Why might you care?

1. Neat proof
2. Pedagogically interesting for its combination of a reduction argument and a simple indistinguishability argument
3. The proof is constructive: each round, it is easy to compute which messages to drop to prevent a decision

Additional contribution: we also show that the FLP model and the model of Santoro and Widmayer are equivalent (they simulate each other)

FLP ‘82: consensus is impossible in an asynchronous message-passing system in which one process may crash

- Process can do arbitrary deterministic, local computation
- Messages are never lost but their delay is unpredictable
- At most one process may crash



FLP ‘82: consensus is impossible in an asynchronous message-passing system in which one process may crash

In the (binary) consensus problem, every process gets a binary input and:

Liveness:

Every process must eventually produce a binary output

Agreement:

No two processes must produce different outputs

Validity:

If all processes start with the same input b, then no process outputs $\overline{b} \neq b$

Santoro and Widmayer ‘89: consensus is impossible in a synchronous message-passing system in which, each round, one process may suffer send-omission failures

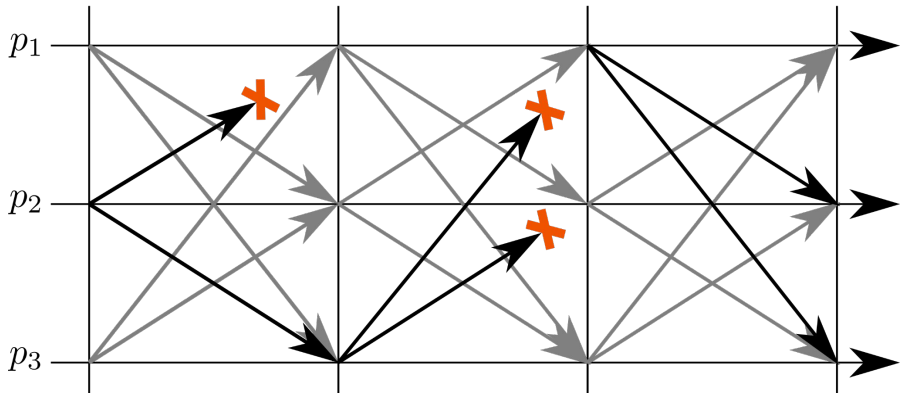
The fail-to-send model

Processes never crash!

We have synchronous, communication-closed rounds

No interleaving of messages

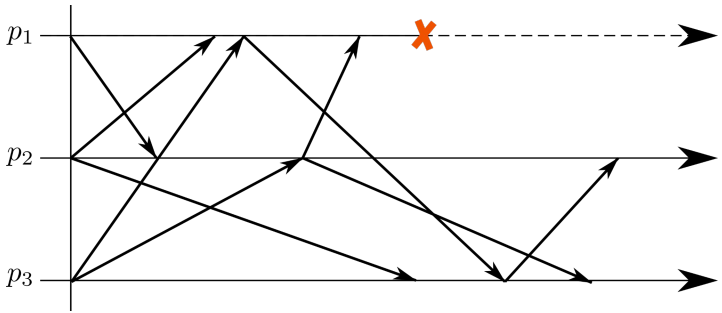
Each round, an adversary picks a process and drops some of its messages



FLP model

Asynchronous communication

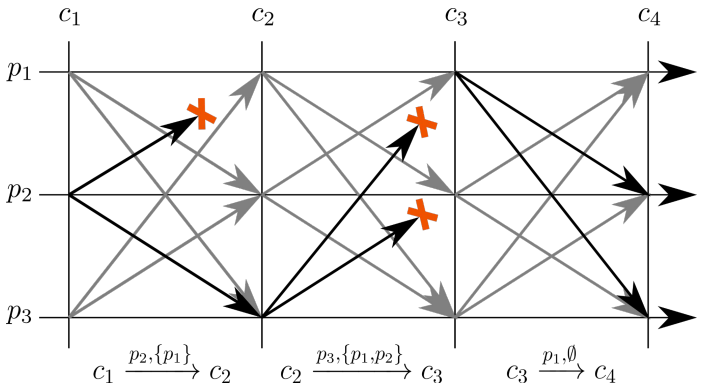
Only one, irrevocable process failure



fail-to-send model

Synchronous, round-by-round communication

Message-omission failures that can affect any one process per round

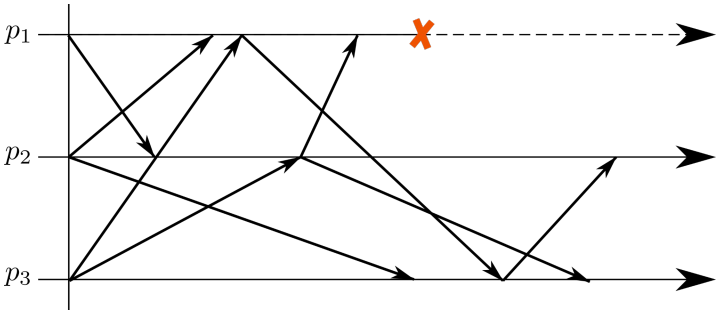


Both original impossibility proofs are quite similar… Can we prove one by reduction to the other?

FLP model

Asynchrony

Only one, irrevocable process failure

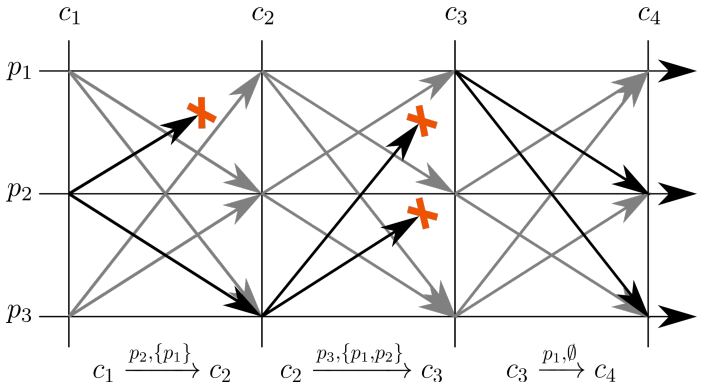


???

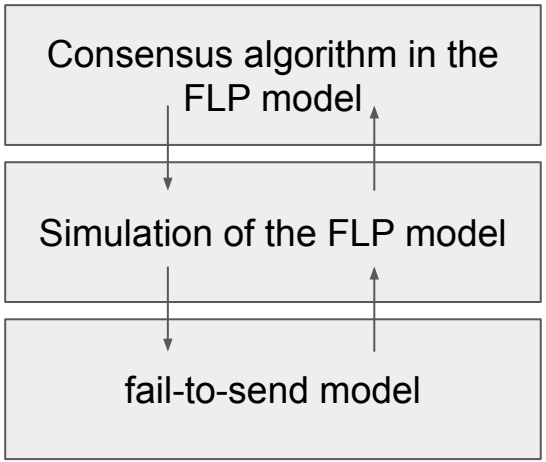
fail-to-send model

Synchrony

Message-omission failures that can affect any one process per round



The proof, step 1:
Simulation of the FLP model in the fail-to-send model



Simulation

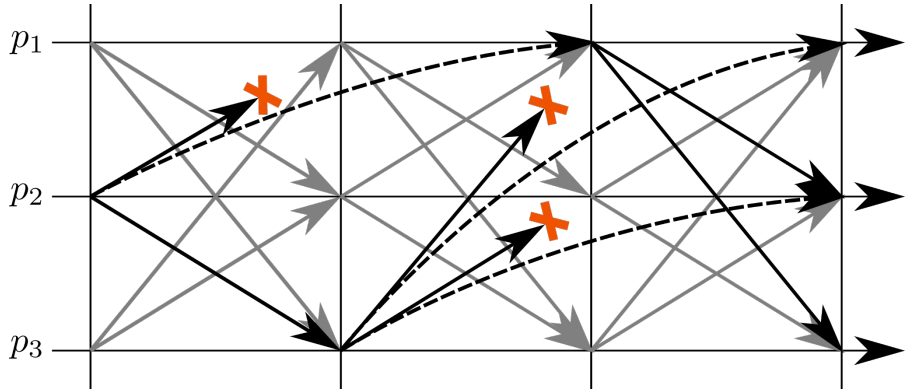
=

Implementation of the communication system

To simulate the FLP model in the fail-to-send model, we just keep re-sending messages to obtain eventual delivery

Each round, each process re-broadcasts every message it ever sent or received (piggybacking on new messages)

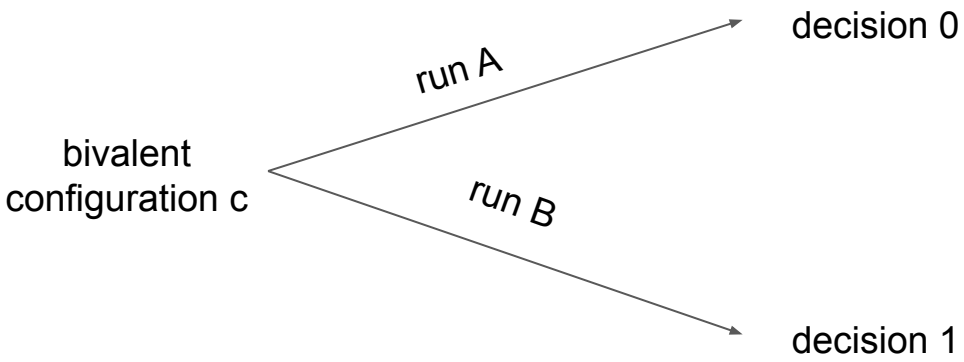
If a process fails to send any message forever, then we can consider it crashed



The proof, step 2:
Impossibility of consensus in the fail-to-send model

Like FLP, Santoro and Widmayer proved consensus impossible in the fail-to-send model using the notion of *bivalent configuration*

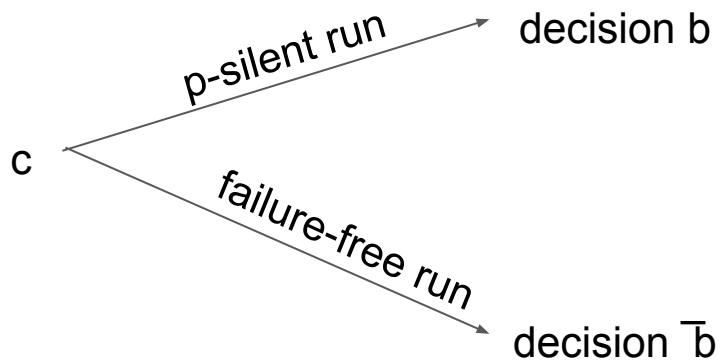
Assuming a consensus algorithm, both FLP and Santoro and Widmayer build an infinite bi-valent run; contradiction!



Key insight: build an infinite run of p-dependent configurations

A configuration c is *p-dependent* when:

- The p-silent run from c decides b
- The failure-free run from c decides $\overline{b} \neq b$



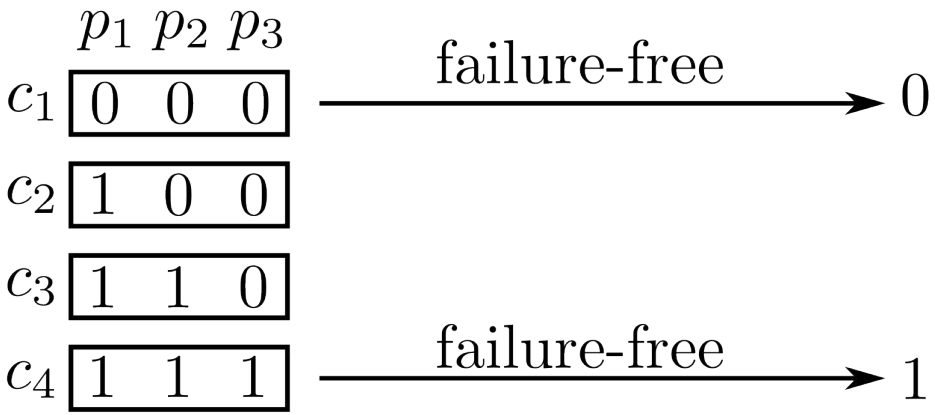
Lemma: a p-silent configuration is undecided

We build an infinite run of p-dependent configurations

Given a pseudo-consensus algorithm (with weaker liveness)

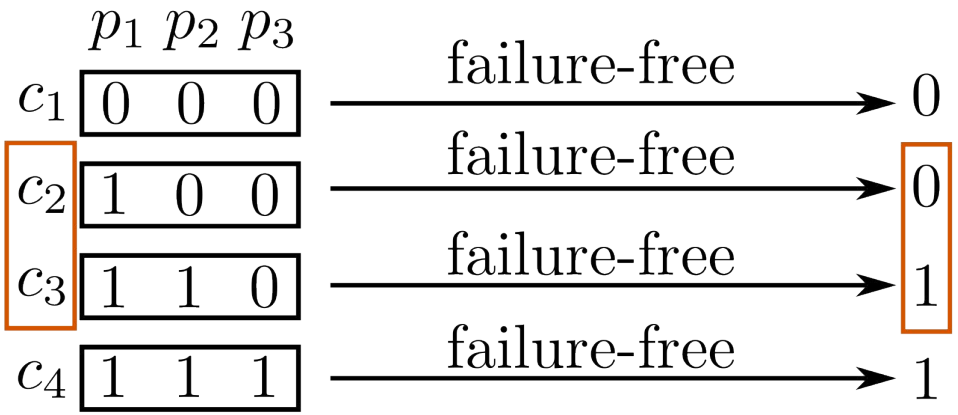
1. There is an initial p-dependent configuration
2. Given a p-dependent configuration, a p’-dependent configuration is reachable in one round.

There is a p-dependent initial configuration

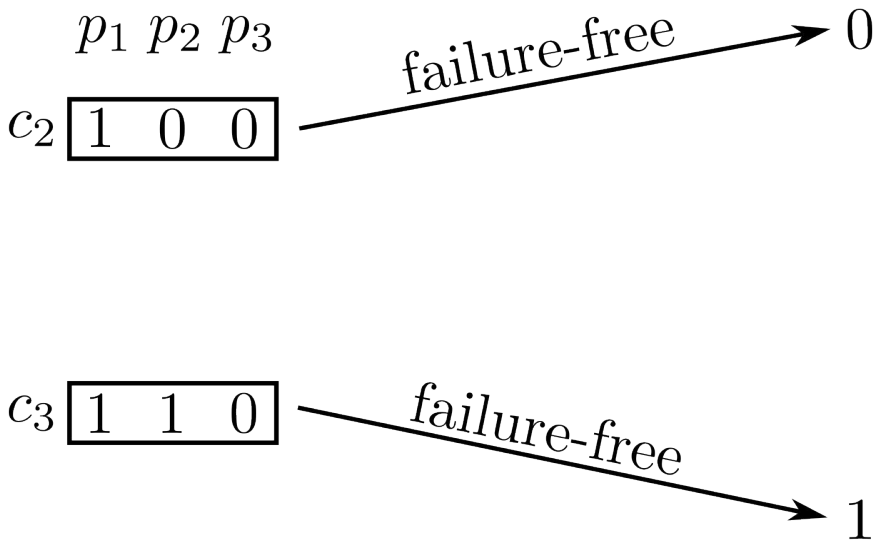


c_i and c_{i+1} are adjacent: only one process has a different state

There is a p-dependent initial configuration

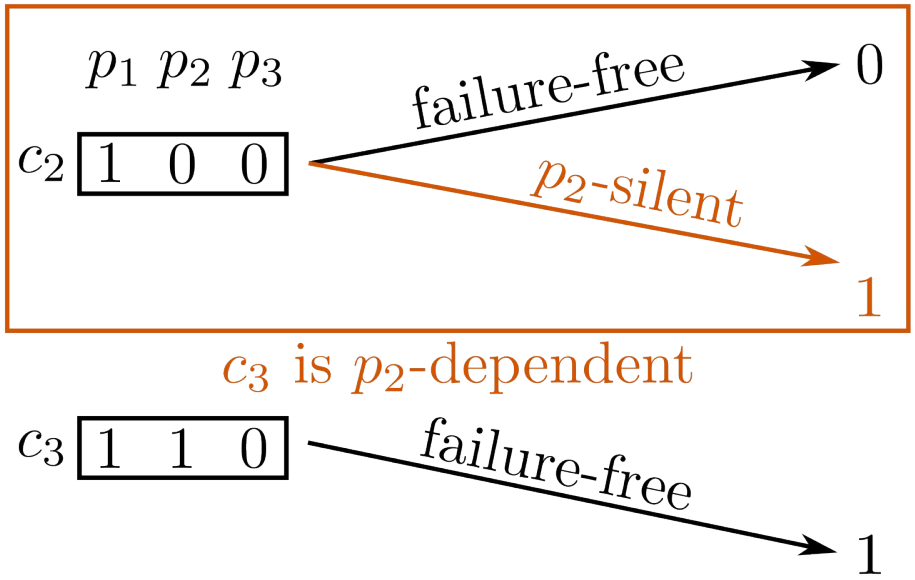


There is a p-dependent initial configuration



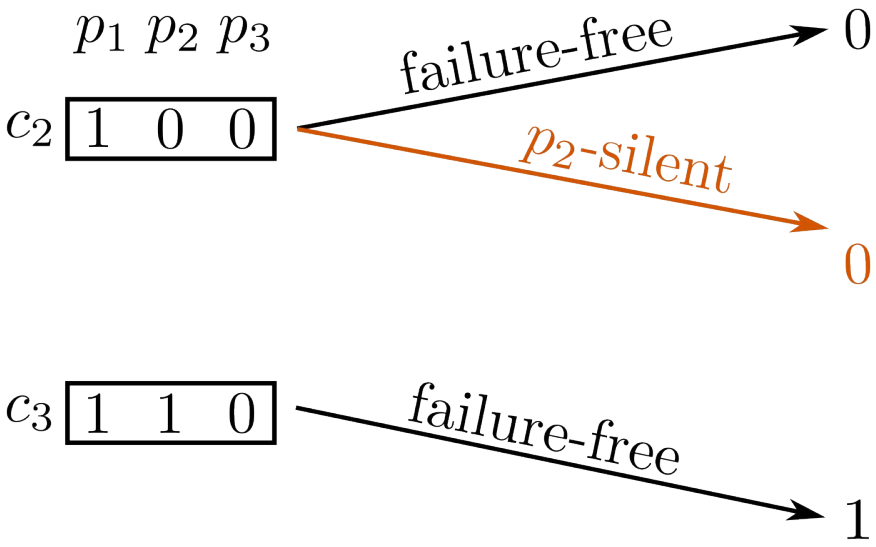
There is a p-dependent initial configuration

Case 1



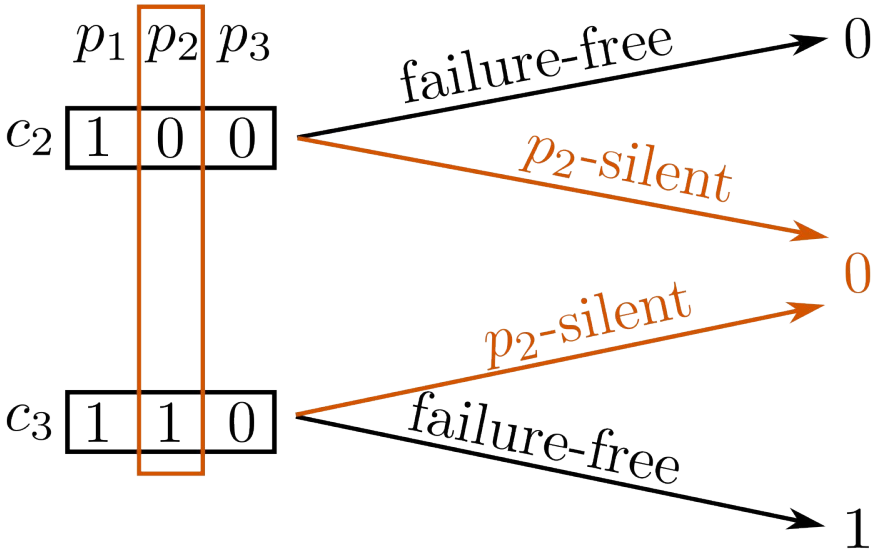
There is a p-dependent initial configuration

Case 2



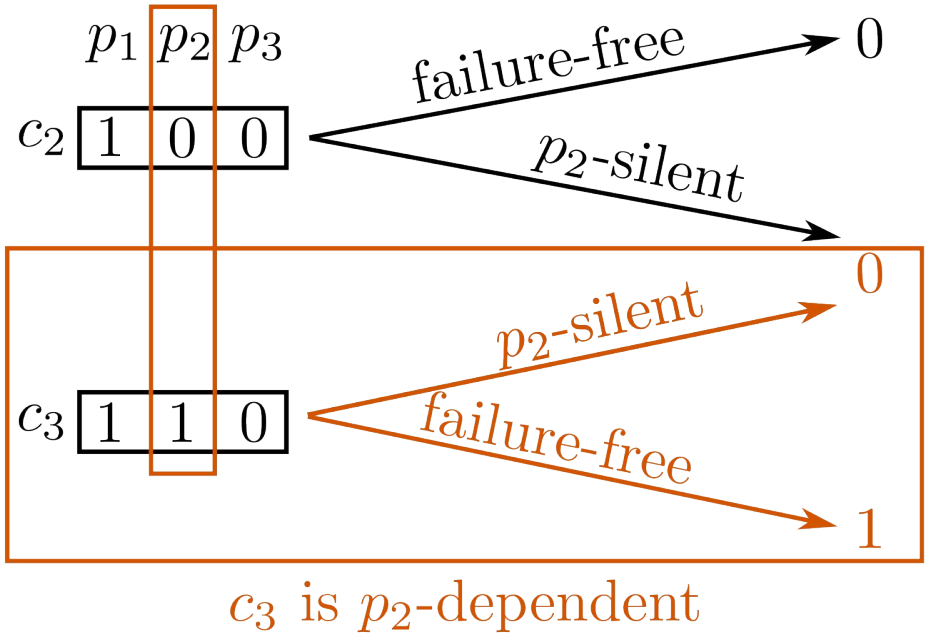
There is a p-dependent initial configuration

Case 2



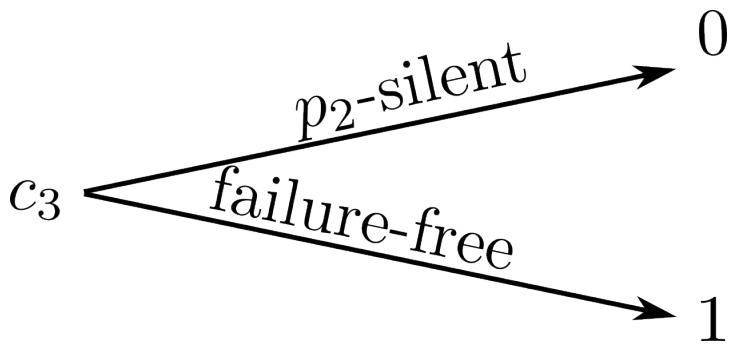
There is a p-dependent initial configuration

Case 2

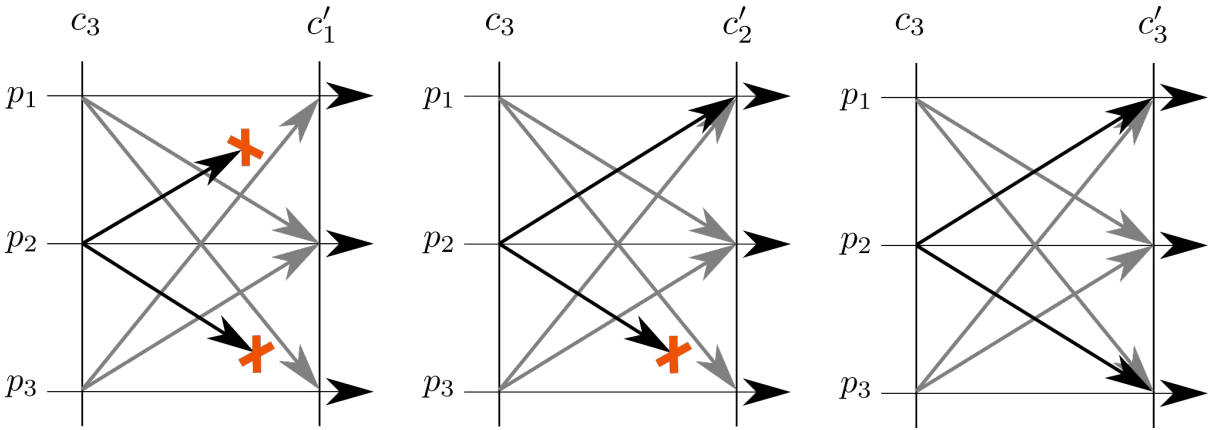


p-dependent leads to p’-dependent in one round

Take c_3 as in the previous slide, where c_3 is p_2-dependent:

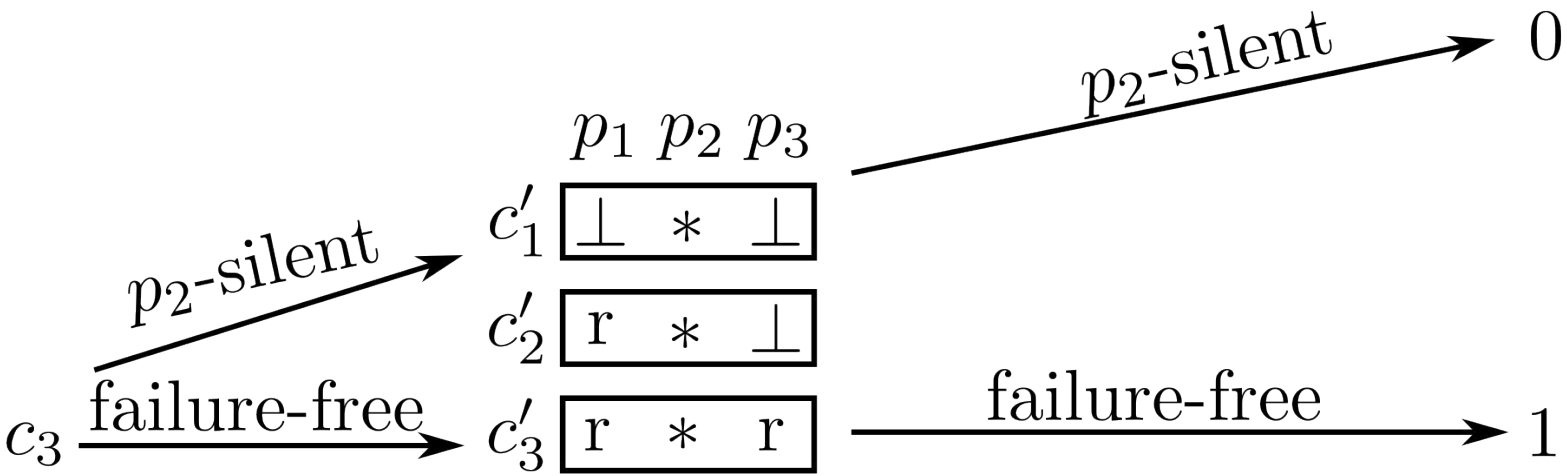


p-dependent leads to p’-dependent in one round



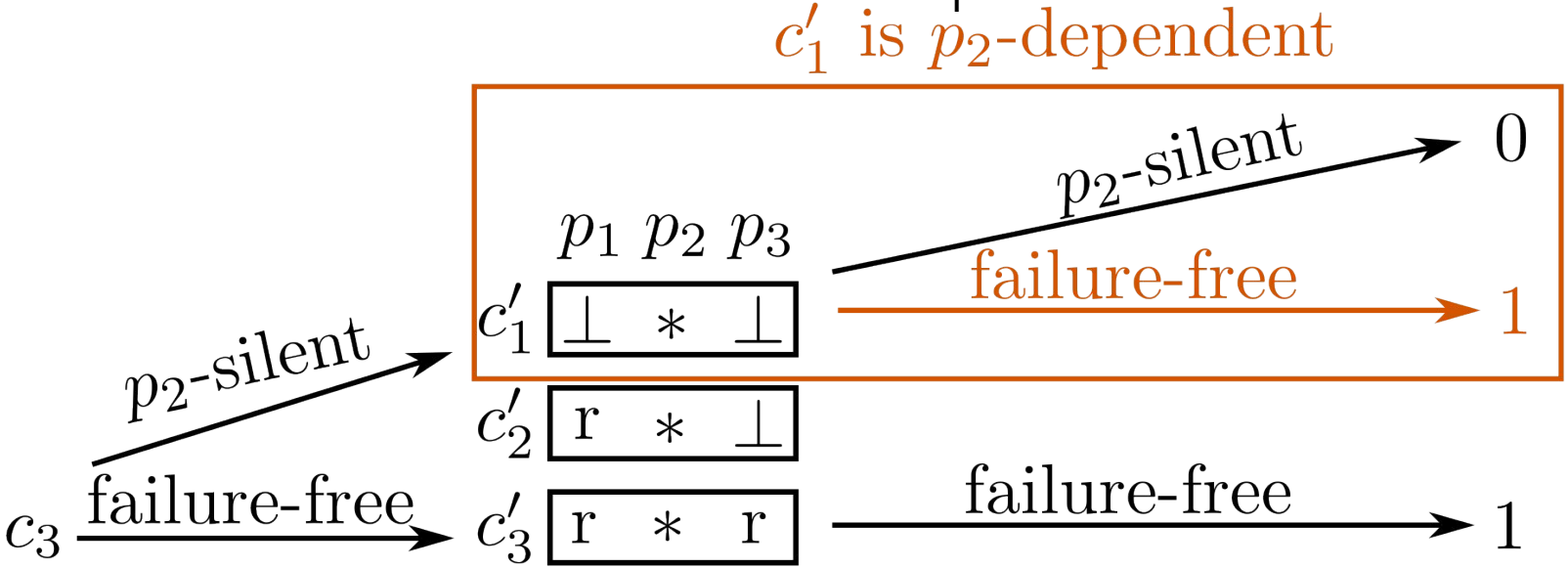
	p_1	p_2	p_3
c'_1	$\perp$	$*$	$\perp$
c'_2	r	$*$	$\perp$
c'_3	r	$*$	r

p-dependent leads to p’-dependent in one round

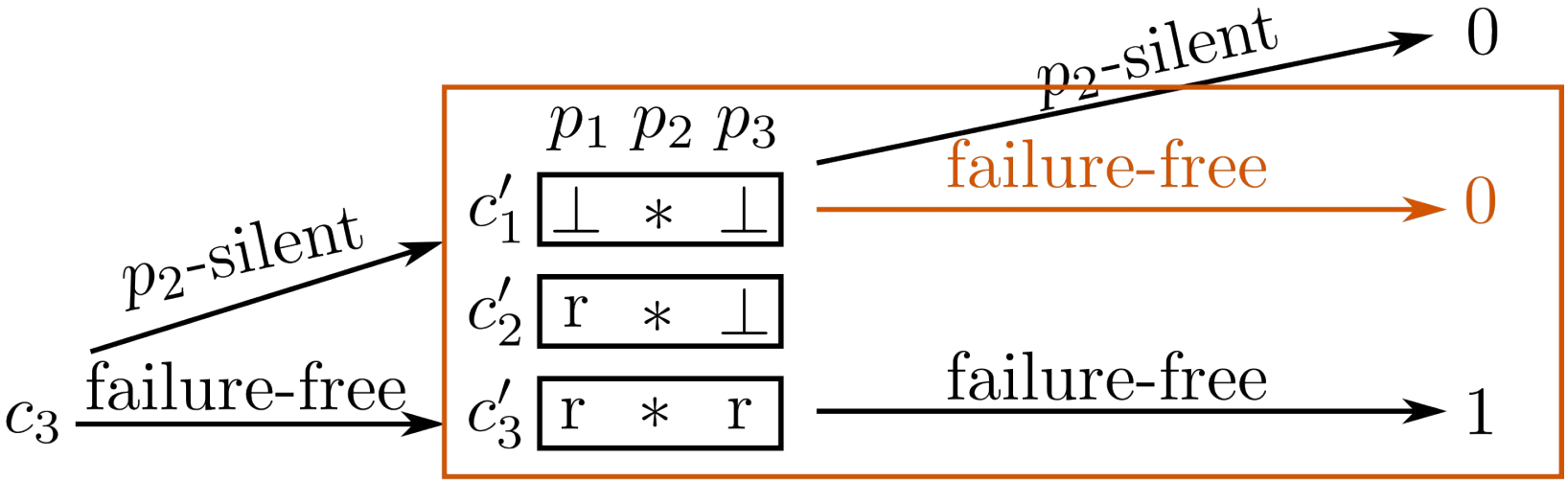


p-dependent leads to p'-dependent in one round

Case 1: failure-free decision from c_1' is 1



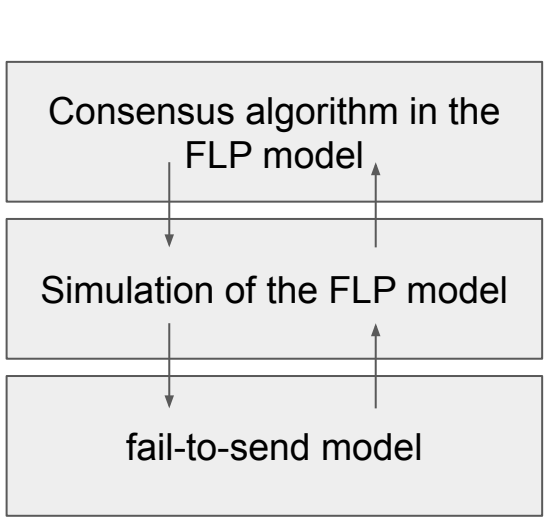
p-dependent leads to p'-dependent in one round
Case 2: failure-free decision from c_1' is 0



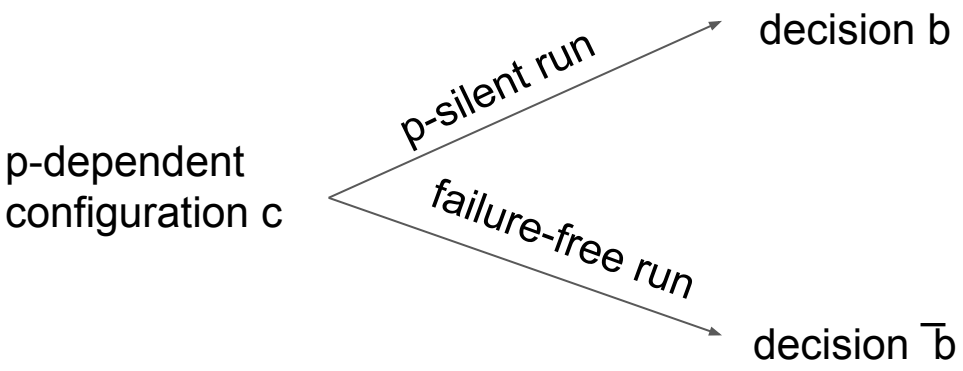
QED

Key ingredients:

- Reduction to impossibility in the synchronous, fail-to-send model
- Proof in the fail-to-send model using p-dependent configurations



~~bivalent~~

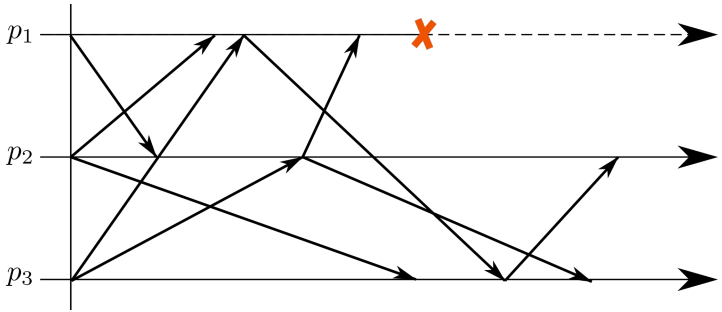


We show equivalence by simulating each model in the other

FLP model

Asynchrony

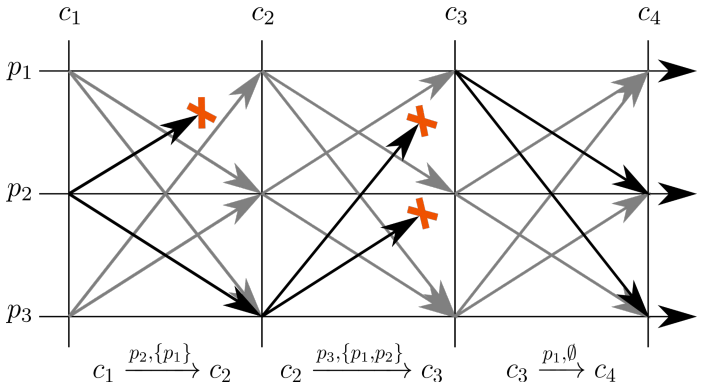
Only one, irrevocable process failure



fail-to-send model (Santoro and Widmayer)

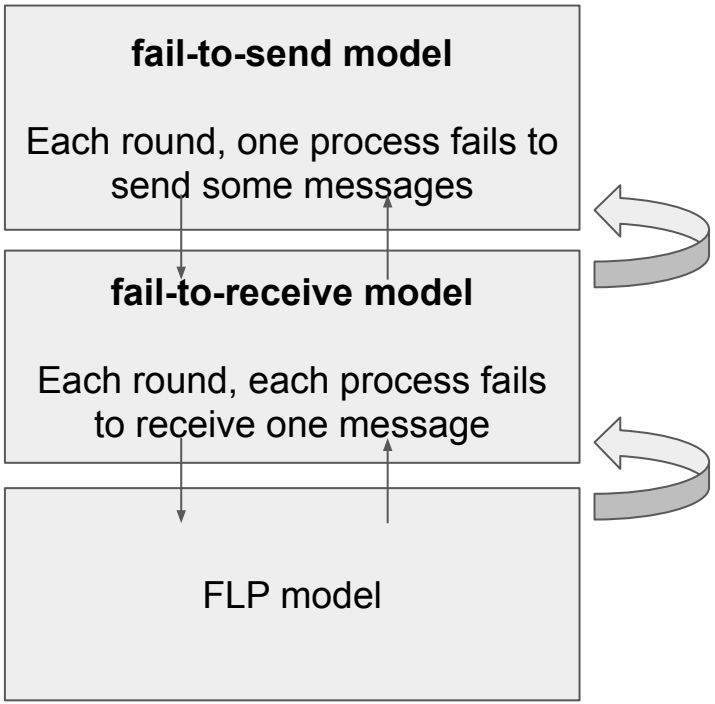
Synchrony

Temporary communication failures that can affect any one process each round



We can also simulate the SW model in the FLP model

This is more surprising: how do we simulate synchrony in an asynchronous model?



Use the get-core algorithm
(3 rounds to simulate one round)

Easy: wait for n-1 messages and move to the next round

Finally: why the title?

Impossibility of Distributed Consensus with One Faulty Process. Fischer, Lynch, and Paterson 1982 (Consensus is impossible in the FLP model)

Time Is Not a Healer. Santoro and Widmayer 1989 (Consensus is impossible in the fail-to-send model)

In hindsight, we see clearly that those two results are equivalent, thus:

Time Is Not a Healer, but It Sure Makes Hindsight 20:20

(In the USA, vision is measured on a scale from 0 to 20)