MANY FACES OF CONSENSUS

OVERVIEW

- What is consensus?
- Consensus in crash failure model
  - Scenarios
  - What is impossible?
  - What is possible?
- Consensus in byzantine model
  - What is impossible?

FORMALISING CONSENSUS

- Given a set of parties with their inputs
- They all must agree on a decision based on their inputs
- Binary consensus
  - 0 or 1

Can be extended to any number of decisions

FORMALISING CONSENSUS

- Given a set of parties with their inputs
- They all must agree on a decision based on their inputs
- Binary consensus
  - 0 or 1

CONSISTENCY

All parties agree on same value

VALIDITY

Agreed-upon value is some party's input

TERMINATION

Each party decided in finite number of steps

HESITATE AND YOU'RE LOST

Crash Failures

INTUITION DEVELOPMENT

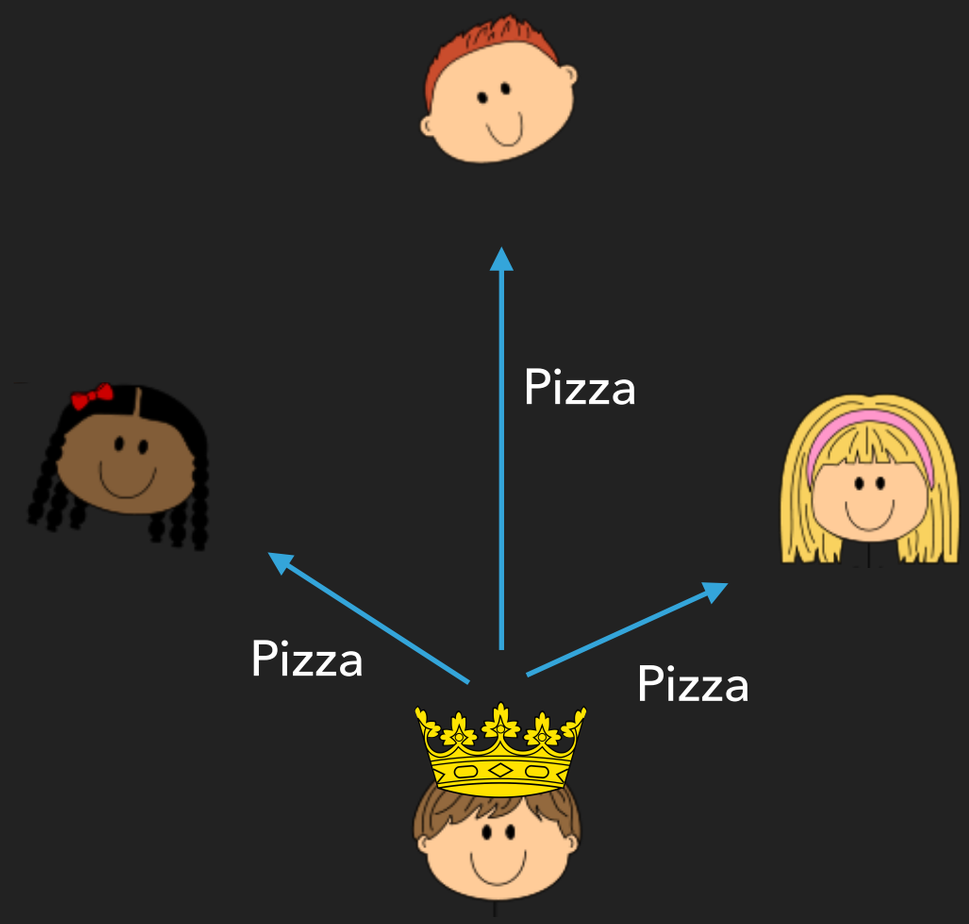
CASE I

To be or not to be : Pizza or Pasta

- Previous communication is allowed
- Unbounded but reliable communication
- No one dies

INTUITION DEVELOPMENT

CASE I



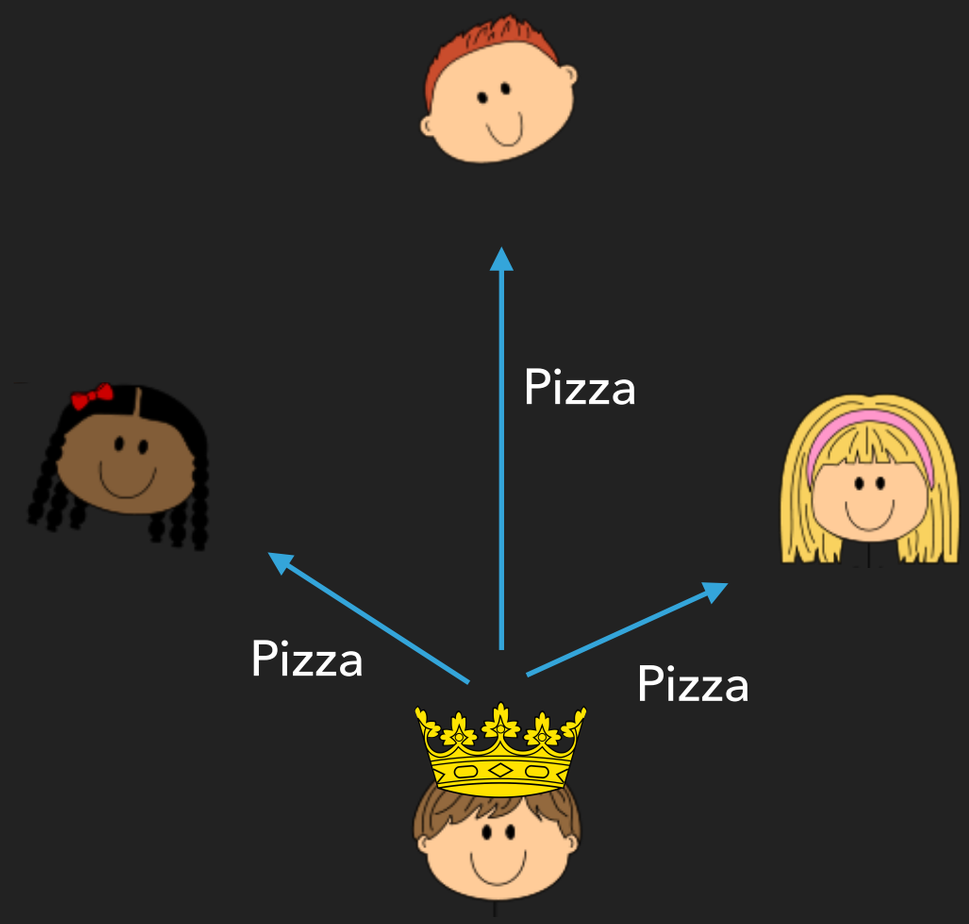
To be or not to be : Pizza or Pasta

- Previous communication is allowed
- Unbounded but reliable communication
- No one dies

YES

INTUITION DEVELOPMENT

CASE II

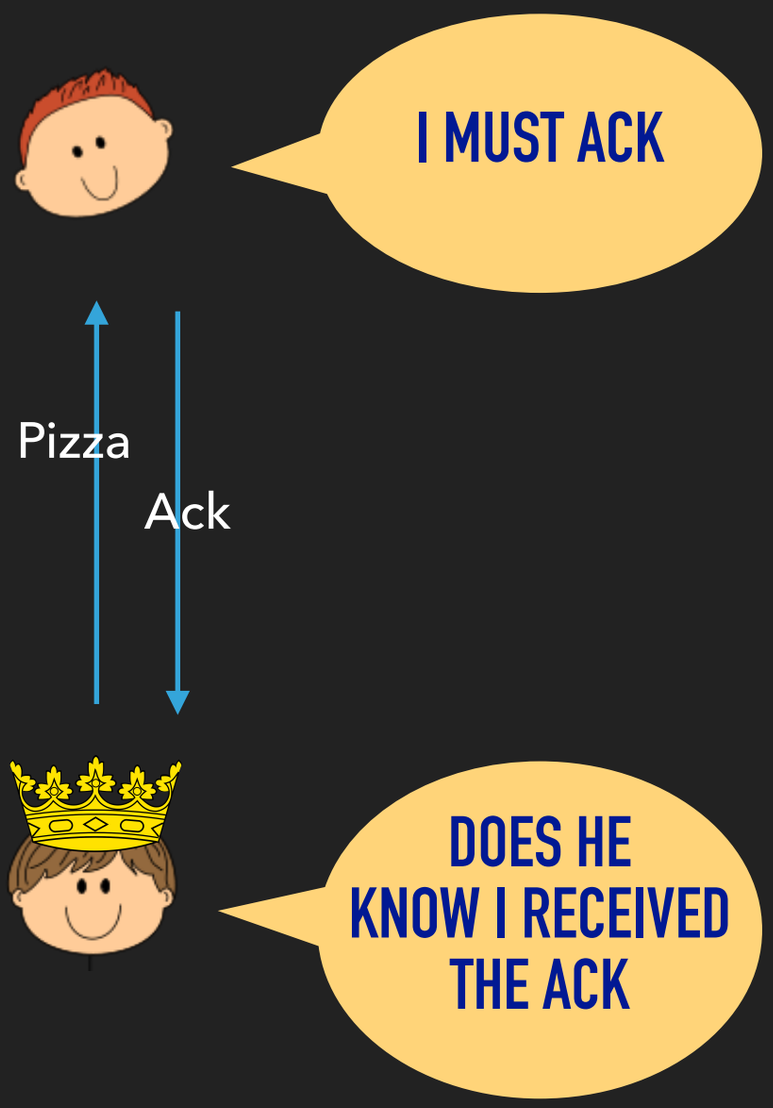


- Previous communication is allowed
- Unbounded but unreliable communication
- No one dies

To be or not to be : Pizza or Pasta

INTUITION DEVELOPMENT

CASE II

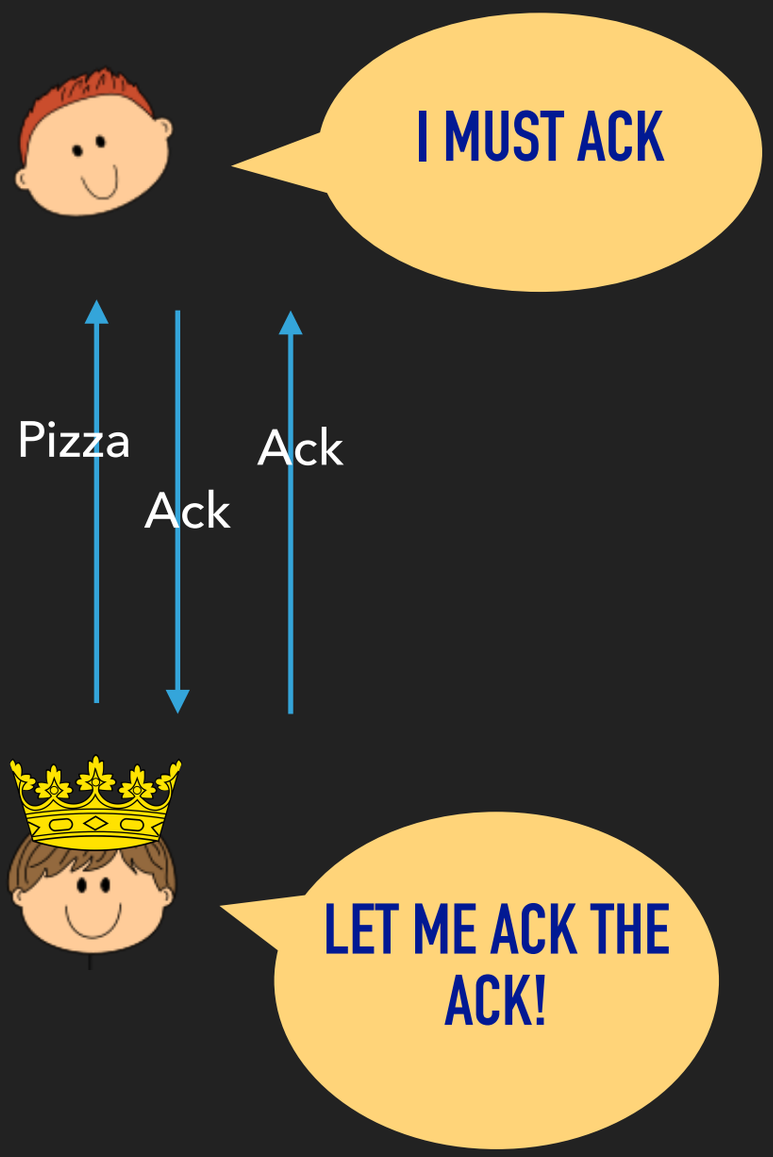


To be or not to be : Pizza or Pasta

- Previous communication is allowed
- Unbounded but unreliable communication
- No one dies

INTUITION DEVELOPMENT

CASE II

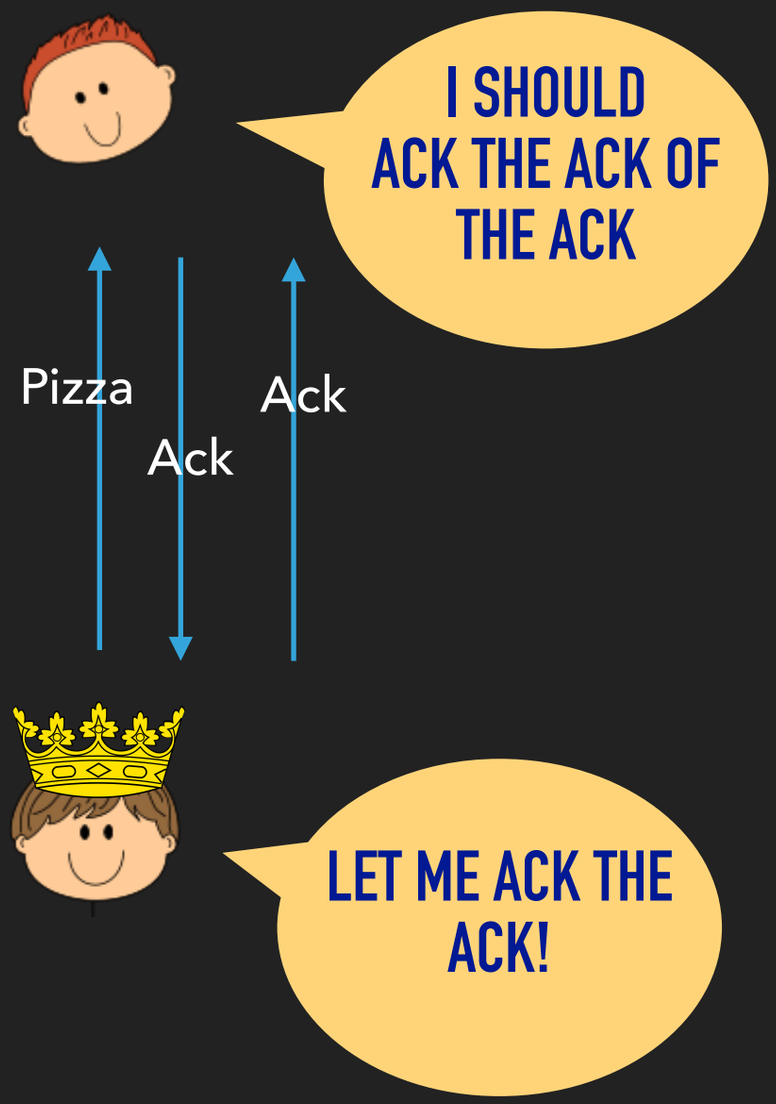


To be or not to be : Pizza or Pasta

- Previous communication is allowed
- Unbounded but unreliable communication
- No one dies

INTUITION DEVELOPMENT

CASE II

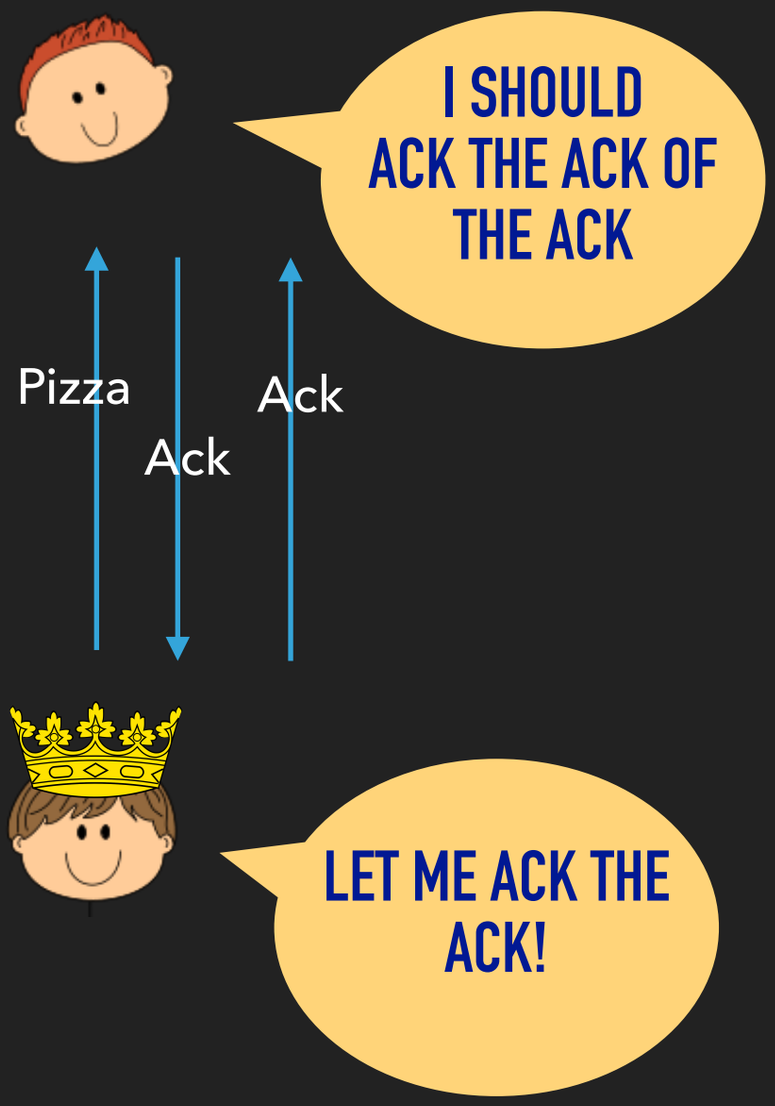


To be or not to be : Pizza or Pasta

- Previous communication is allowed
- Unbounded but unreliable communication
- No one dies

INTUITION DEVELOPMENT

CASE II



- Previous communication is allowed
- Unbounded but unreliable communication
- No one dies

MAYBE NOT

QUACK . . . QUACK . . . QUACK . . . QUACK . . . QUACK . . . QUACK

INTUITION DEVELOPMENT

CASE III

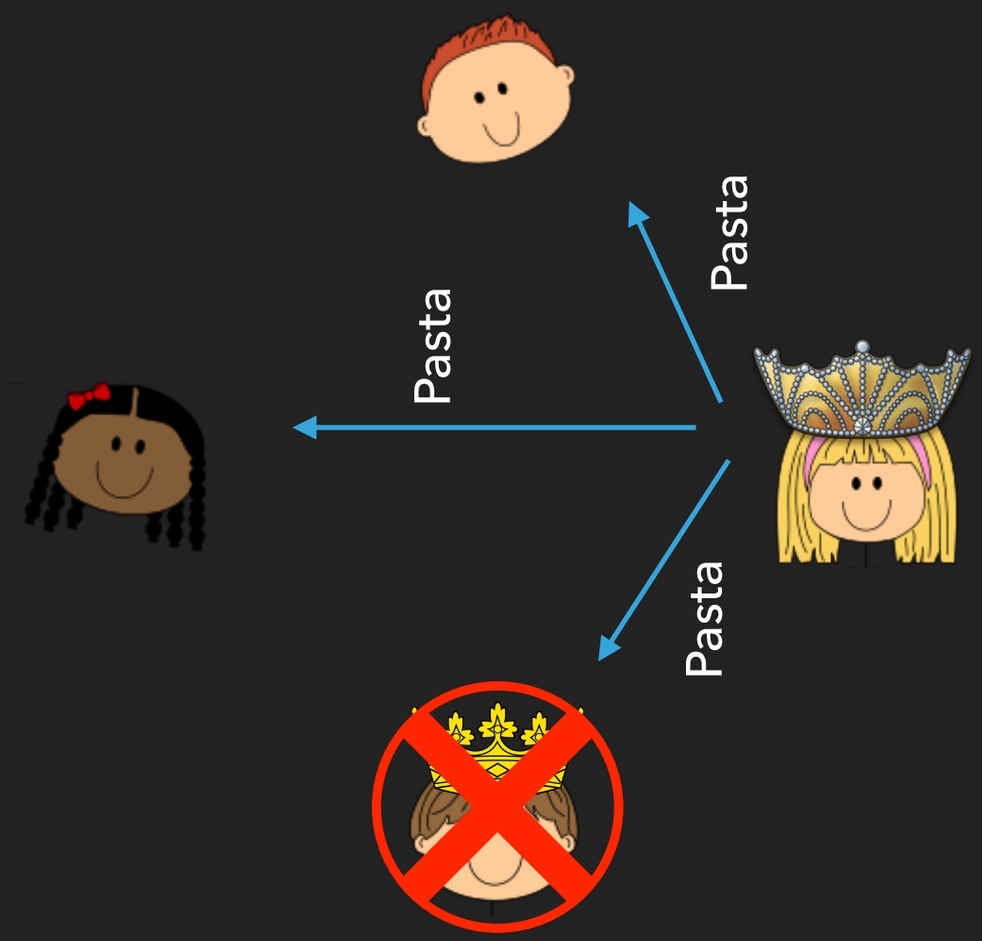
To be or not to be : Pizza or Pasta

- Previous communication is allowed
- Unbounded but reliable communication
- One can die
- Others can not differentiate b/w whether crown is dead or network is slow

MAYBE NOT

INTUITION DEVELOPMENT

CASE IV



To be or not to be : Pizza or Pasta

- Previous communication is allowed
- Bounded but reliable communication

Use timeouts

- One can die

YES

CONSENSUS DEPENDS HEAVILY ON SYSTEM MODEL

Takeaway

CONSENSUS* IS IMPOSSIBLE IN A COMPLETELY ASYNCHRONOUS MESSAGE PASSING SYSTEMS WHERE EVEN A SINGLE PROCESSOR FAILS

[Fischer, Lynch, Paterson '85]

MODELS OF COMMUNICATION

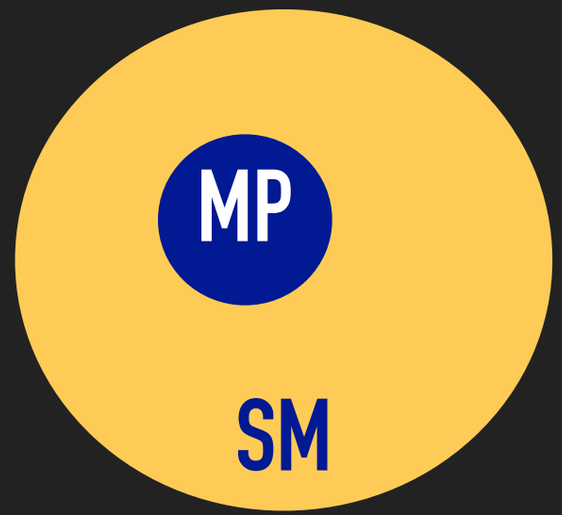
MESSAGE PASSING

FLP

SHARED MEMORY

OUR PROOF

- Proof in asynchronous shared-memory [Herlihy '88]
- More powerful than message-passing
  - Inherent broadcasting capabilities
  - On crashes, value still in memory



DETOUR

MP = SM (t < 1/2)

[Attiya, Noy, Dolev '90]

PROOF STRATEGY

Assume otherwise …

Reason about the properties of any such protocol

Derive a contradiction

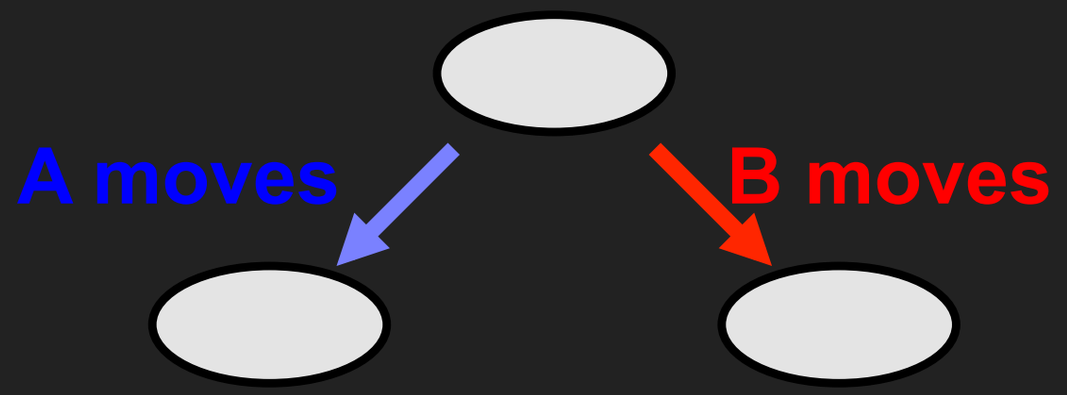
Quod

Erat

Demonstrandum

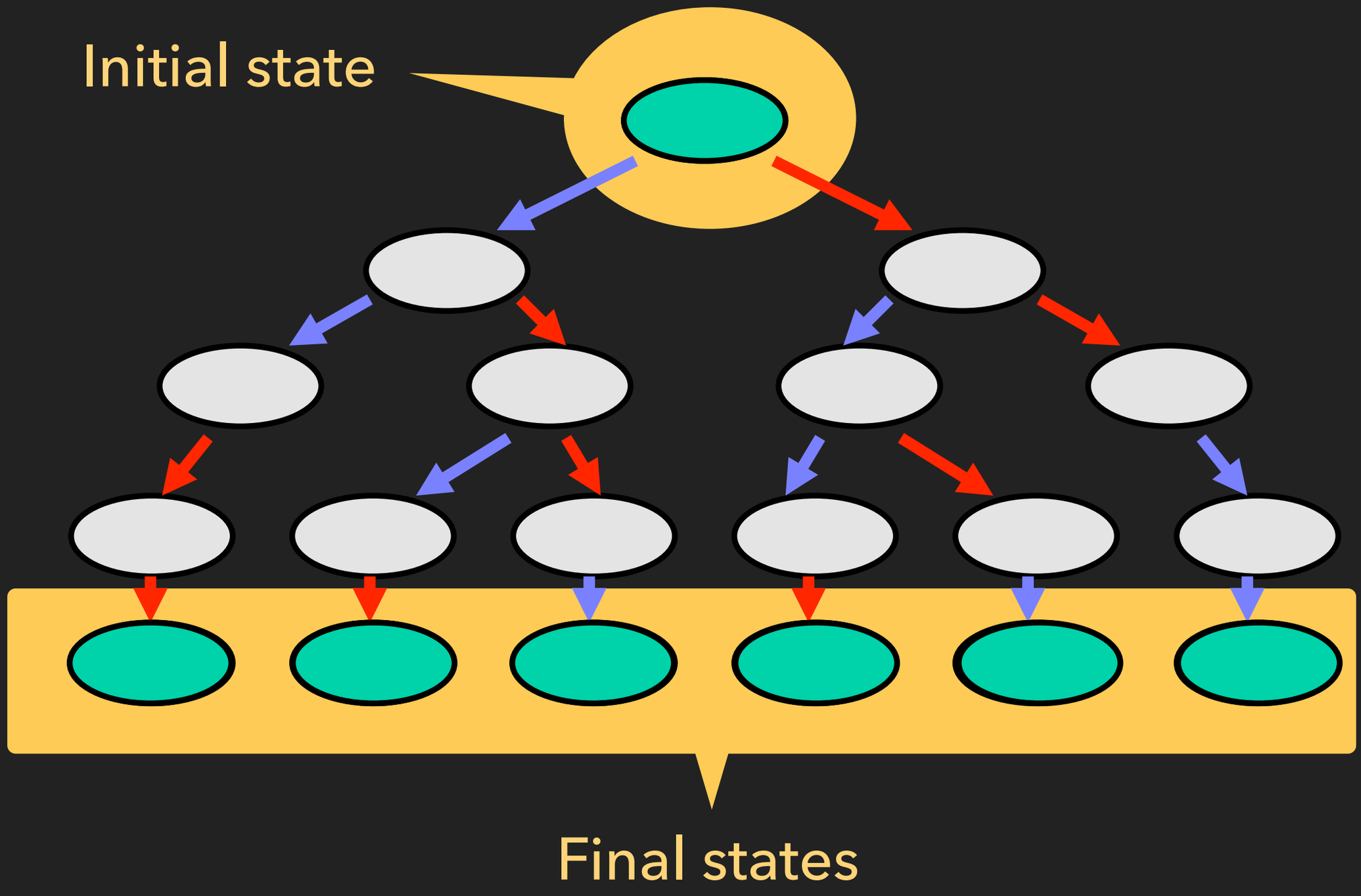
Enough to consider binary consensus and n = 2

PROOF

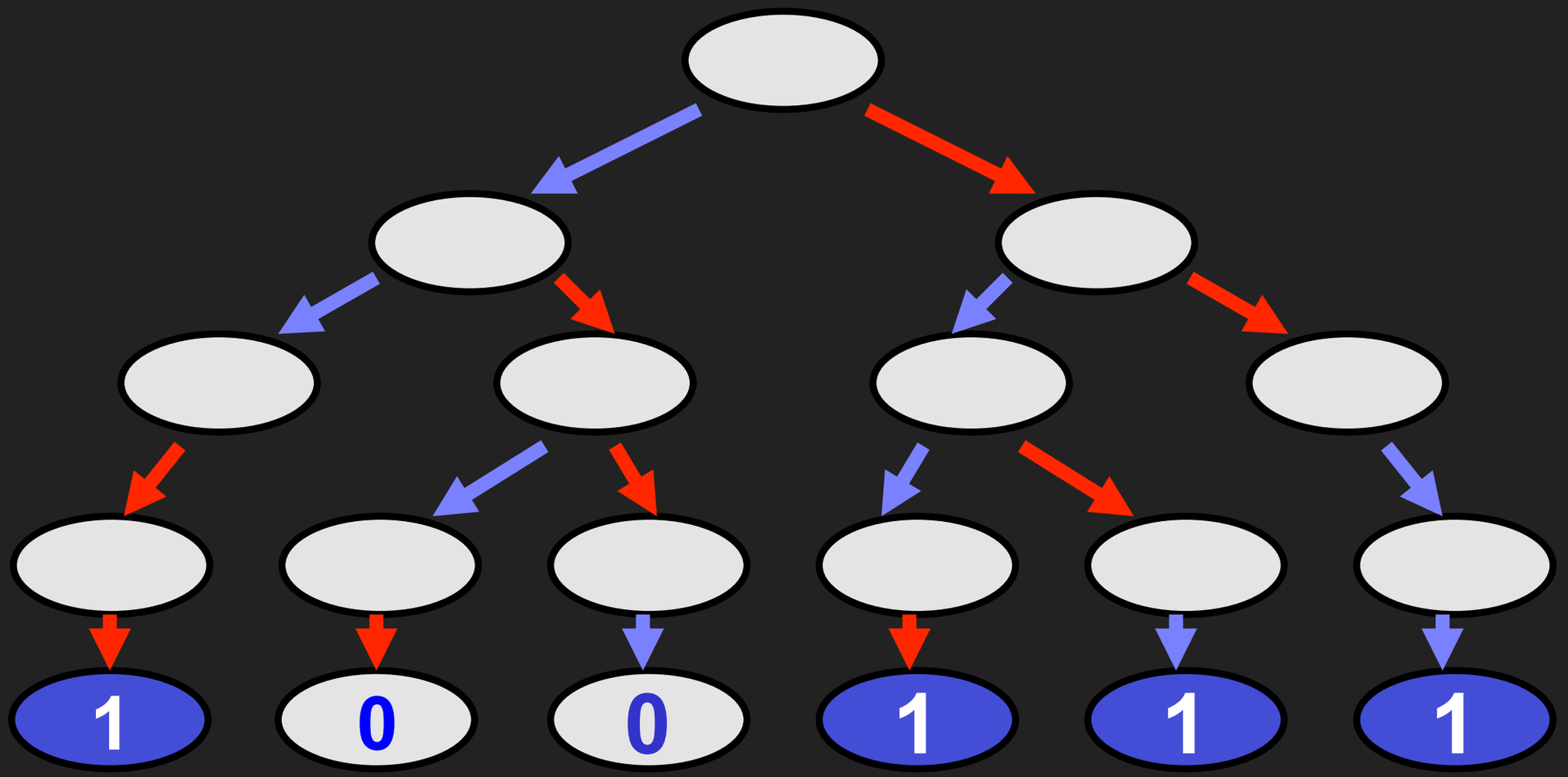


- Either A or B “moves”
- Moving means
  - Register read
  - Register write

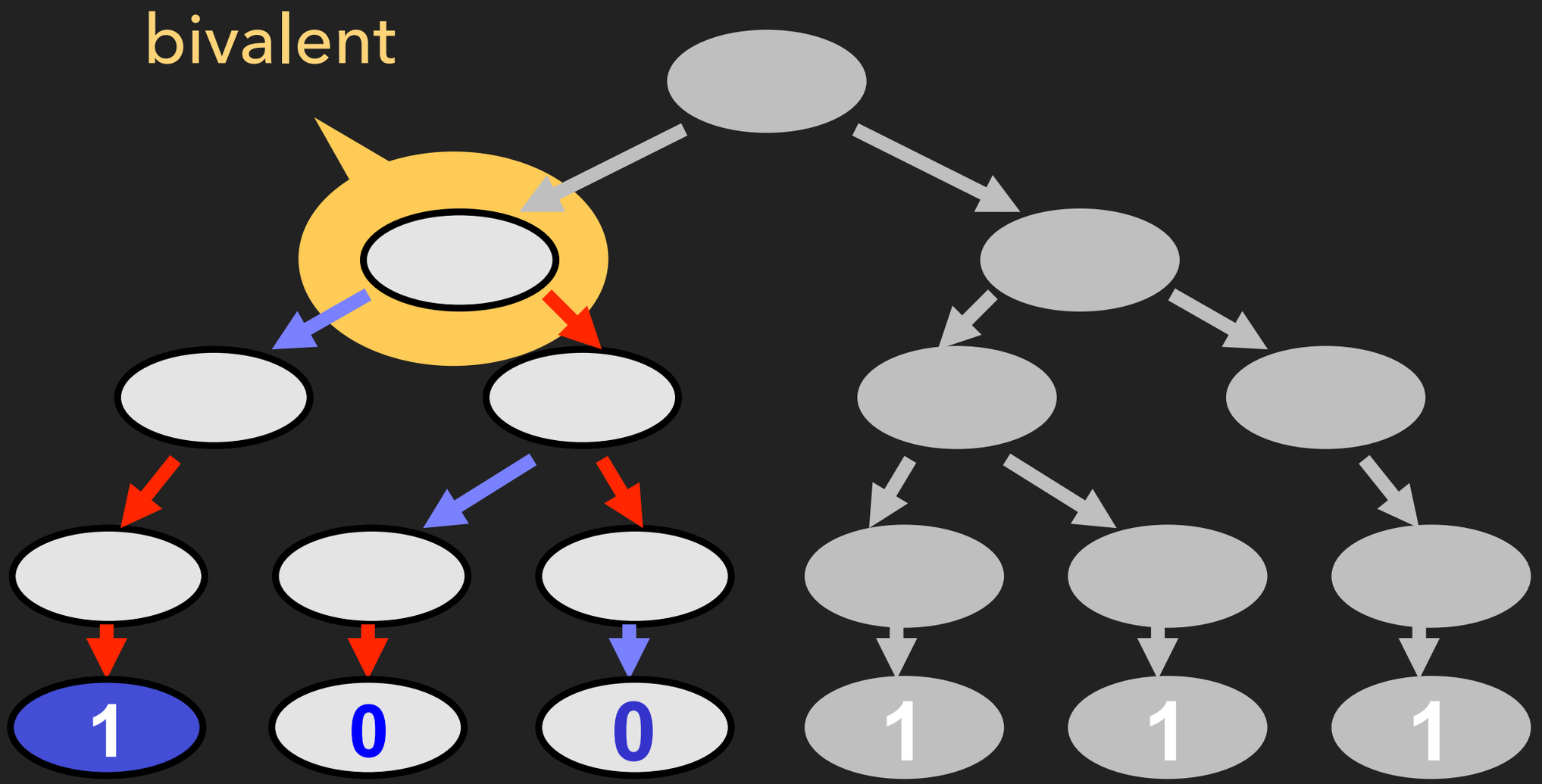
THE TWO-MOVE TREE



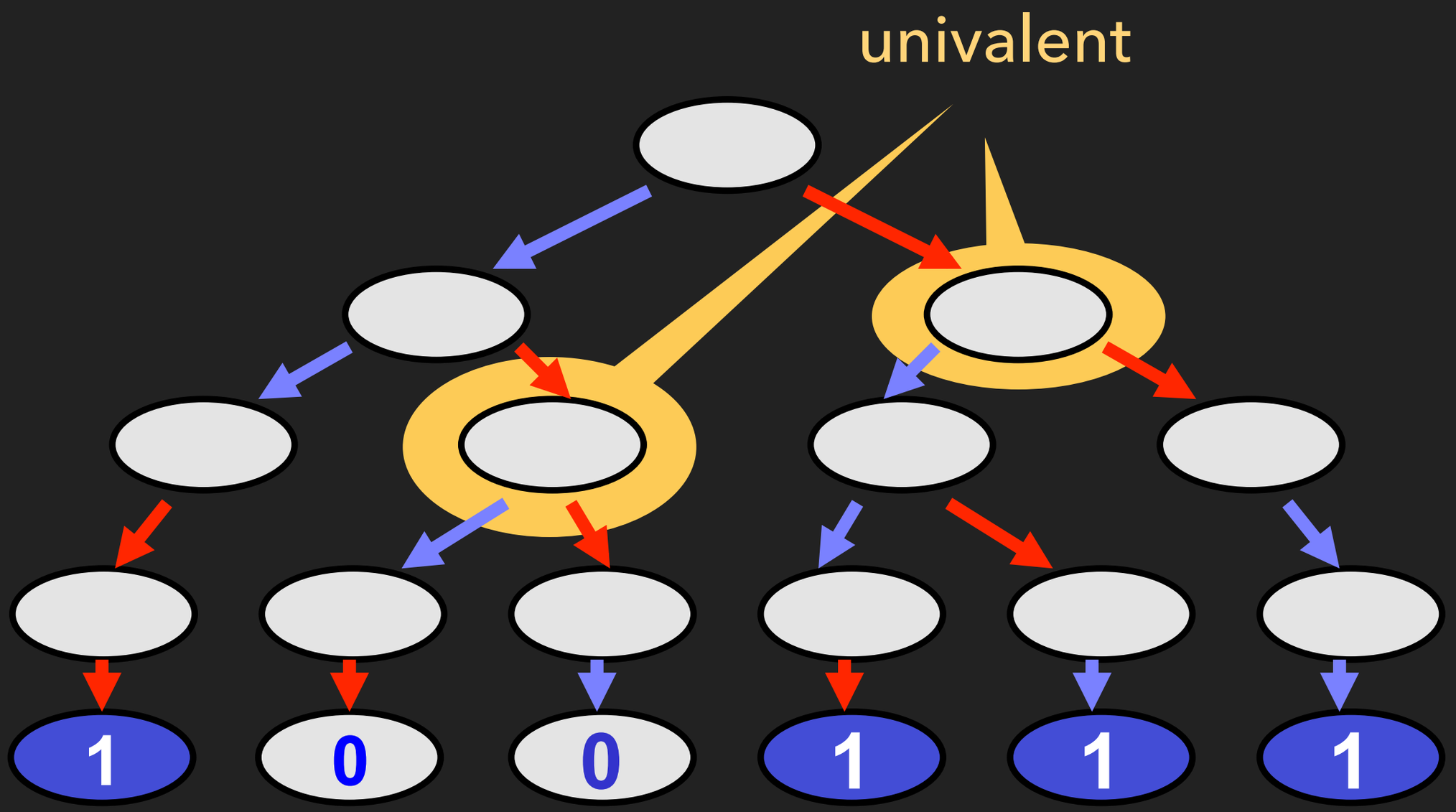
DECISION VALUES



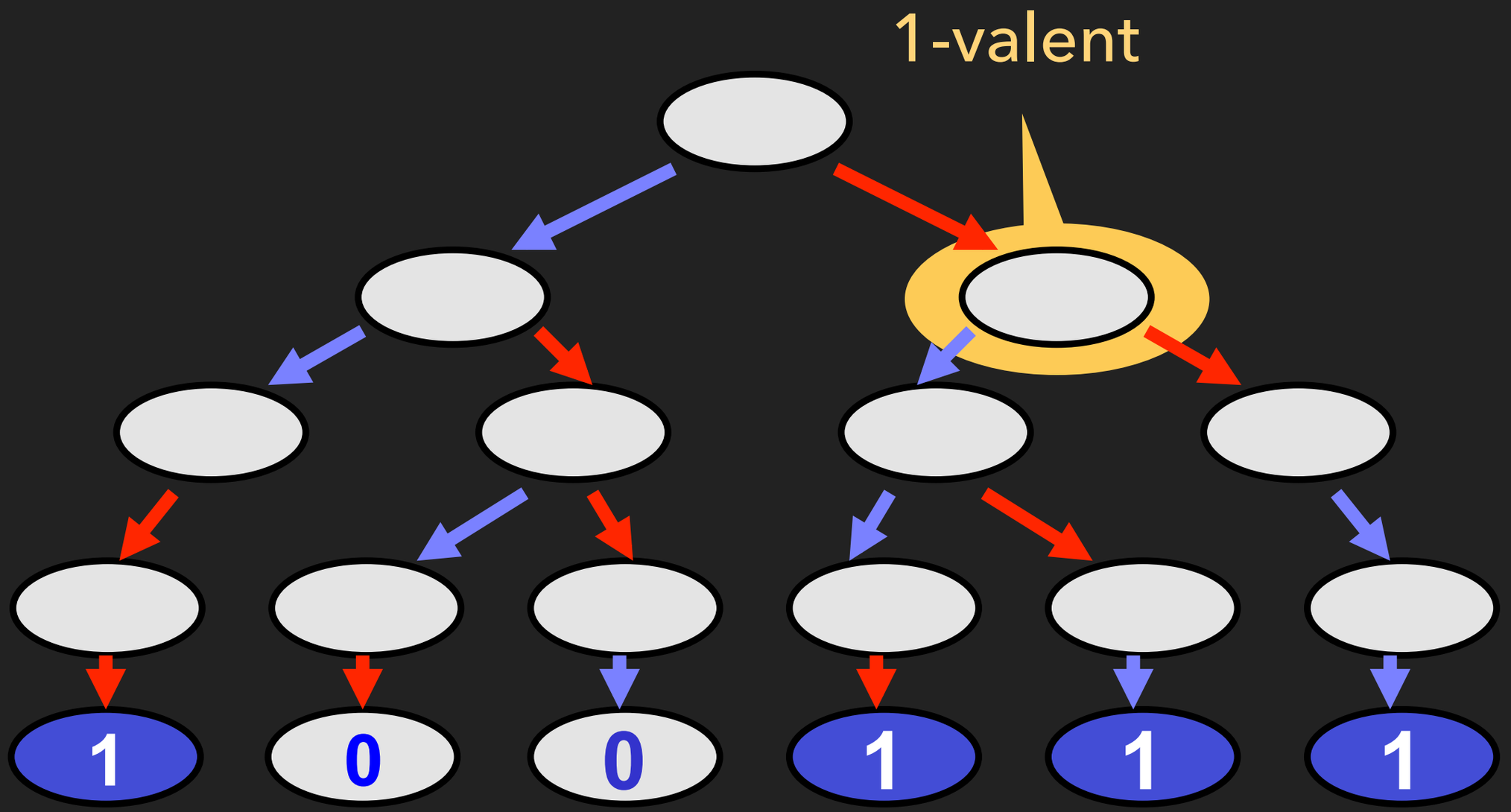
BIVALENT : BOTH POSSIBLE



DECISION VALUES



DECISION VALUES



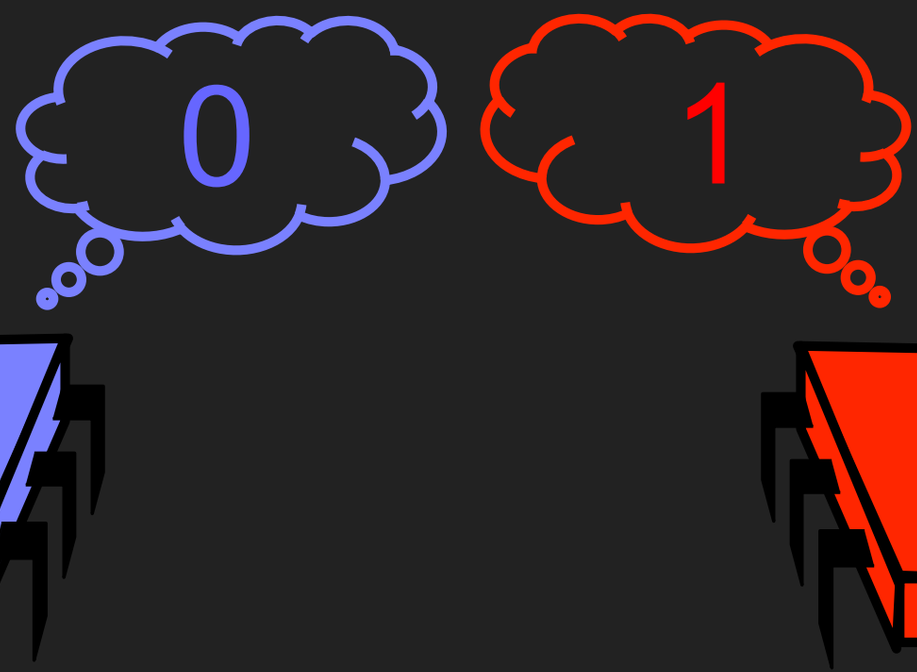
x-valent : x is the only decision value

SUMMARY

- Any consensus protocol is a tree
- Bivalent system states
  - outcome not fixed
- Univalent states
  - Outcome is fixed
  - May not be “known” yet
- 1-Valent and 0-Valent states

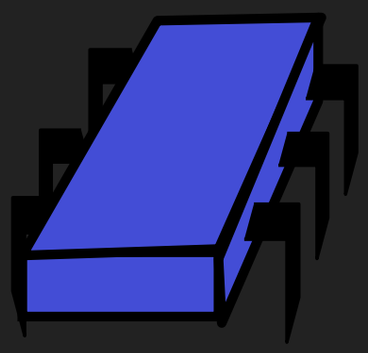
THERE EXISTS AN INITIAL BIVALENT STATE

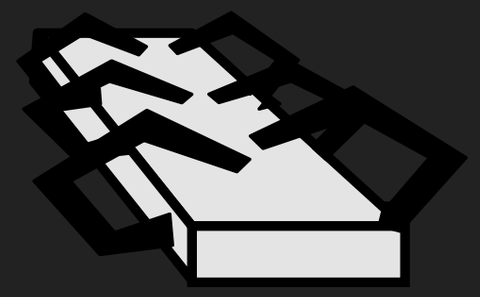
What if inputs differ?



THERE EXISTS AN INITIAL BIVALENT STATE

Must decide on 0





In this solo execution by A

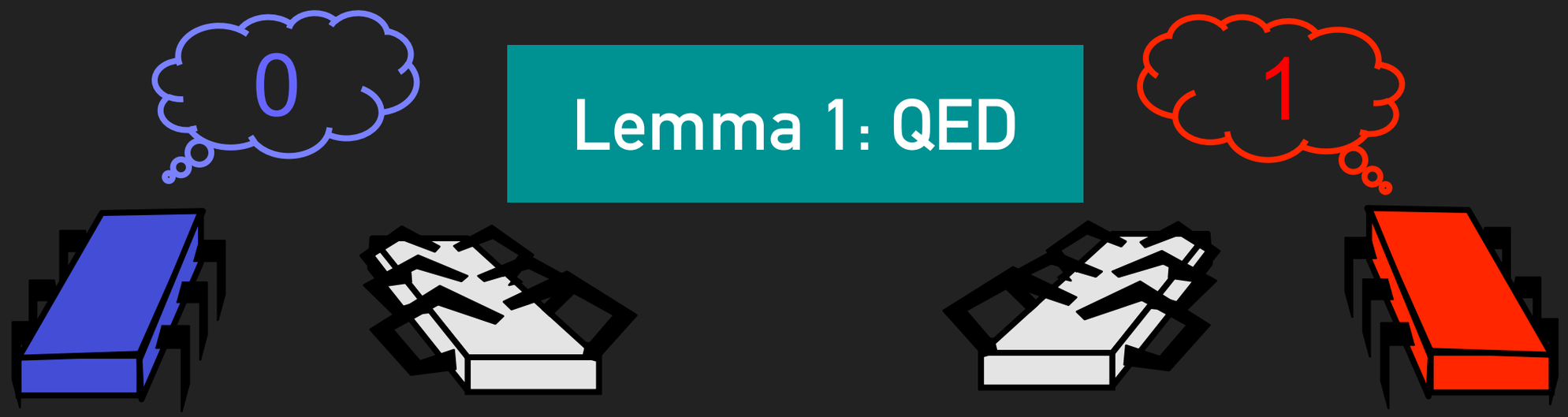
THERE EXISTS AN INITIAL BIVALENT STATE

Must decide on 1

In this solo execution by B

THERE EXISTS AN INITIAL BIVALENT STATE

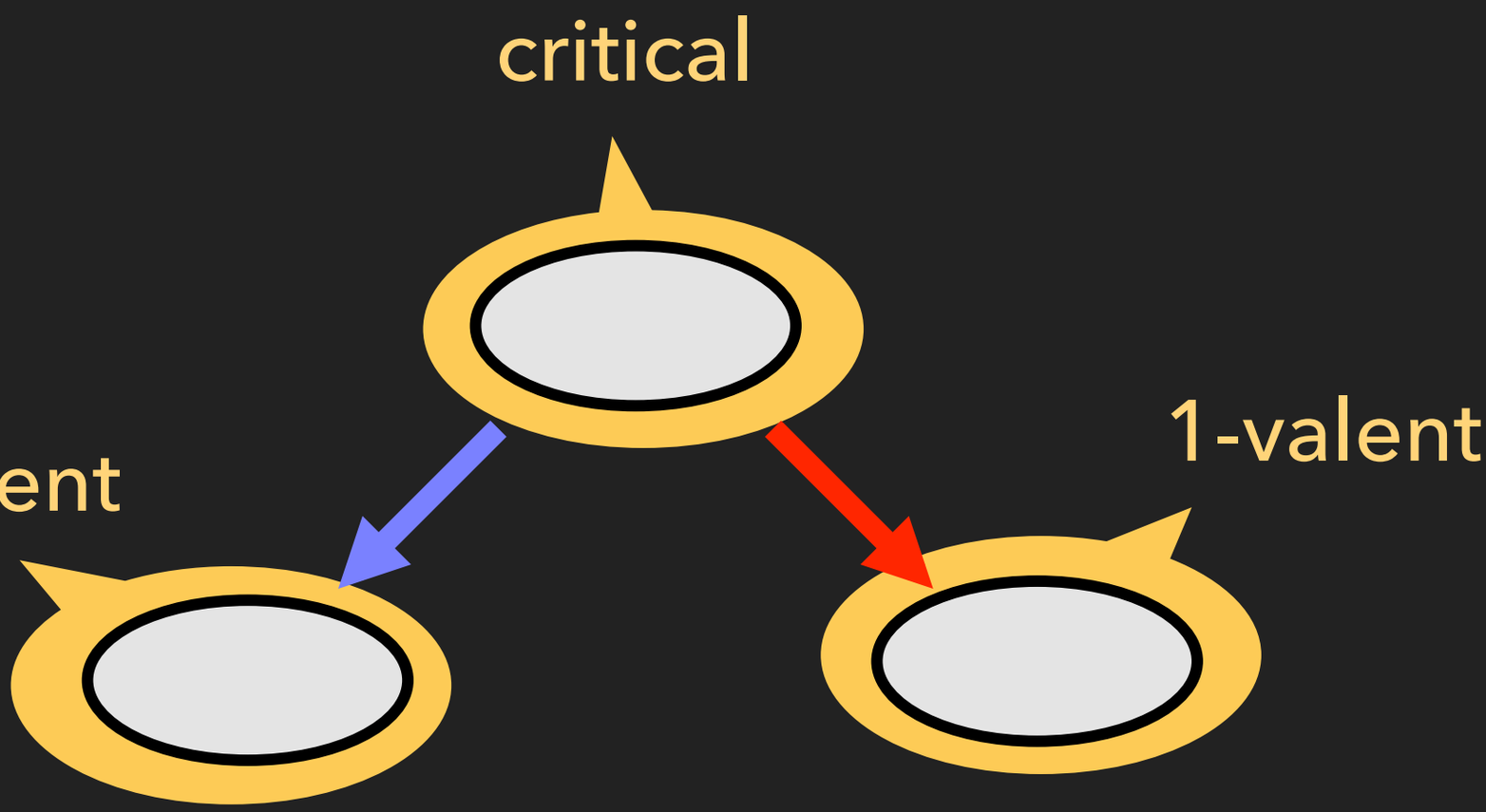
Mixed initial state bivalent



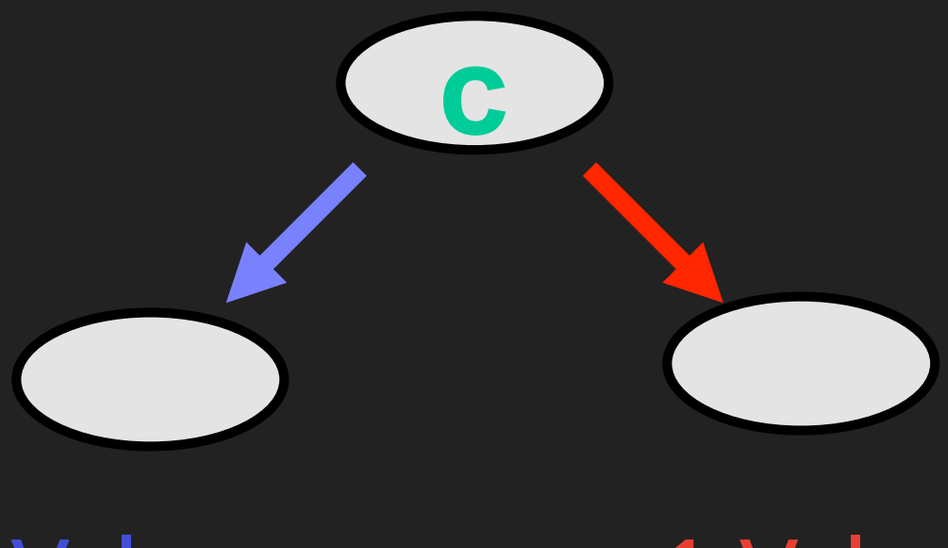
Solo execution by A
must decide 0

Solo execution by B
must decide 1

CRITICAL STATES



FROM A CRITICAL STATE



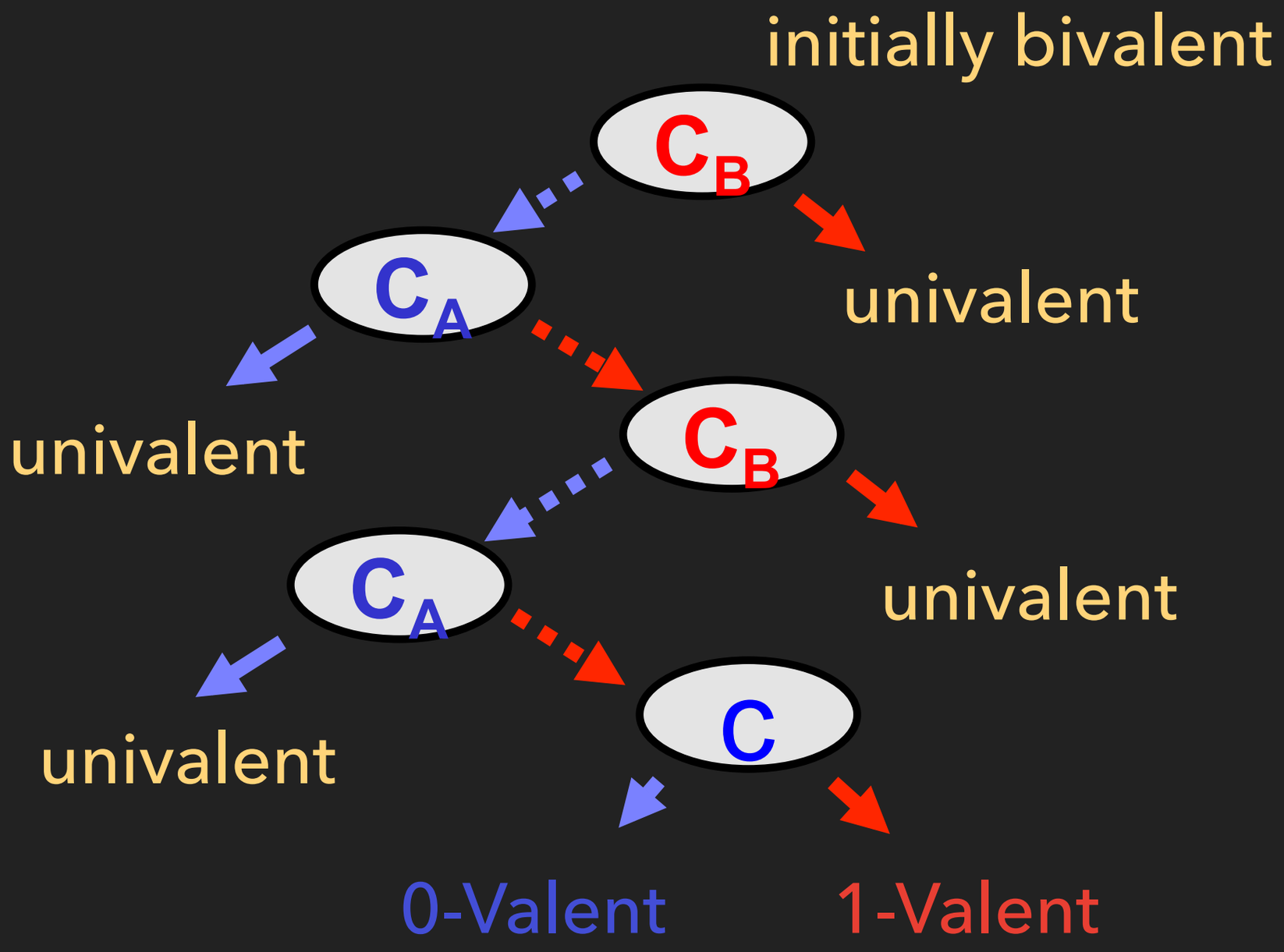
0-Valent

1-Valent

If A goes first protocol decides 0

If B goes first protocol decides 1

WHY MUST THERE BE A CRITICAL STATE ?



CRITICAL STATES

- Starting from a bivalent state
- The protocol can reach a critical state
  - otherwise we can stay bivalent forever
- And the protocol does not terminate in finite steps

CLOSING DEAL. . .

- Starting from a critical state
- Each processor fixes the outcome by
  - Reading or writing
  - Same or different registers
- Leading to a 0 or 1 decision
- And a contradiction

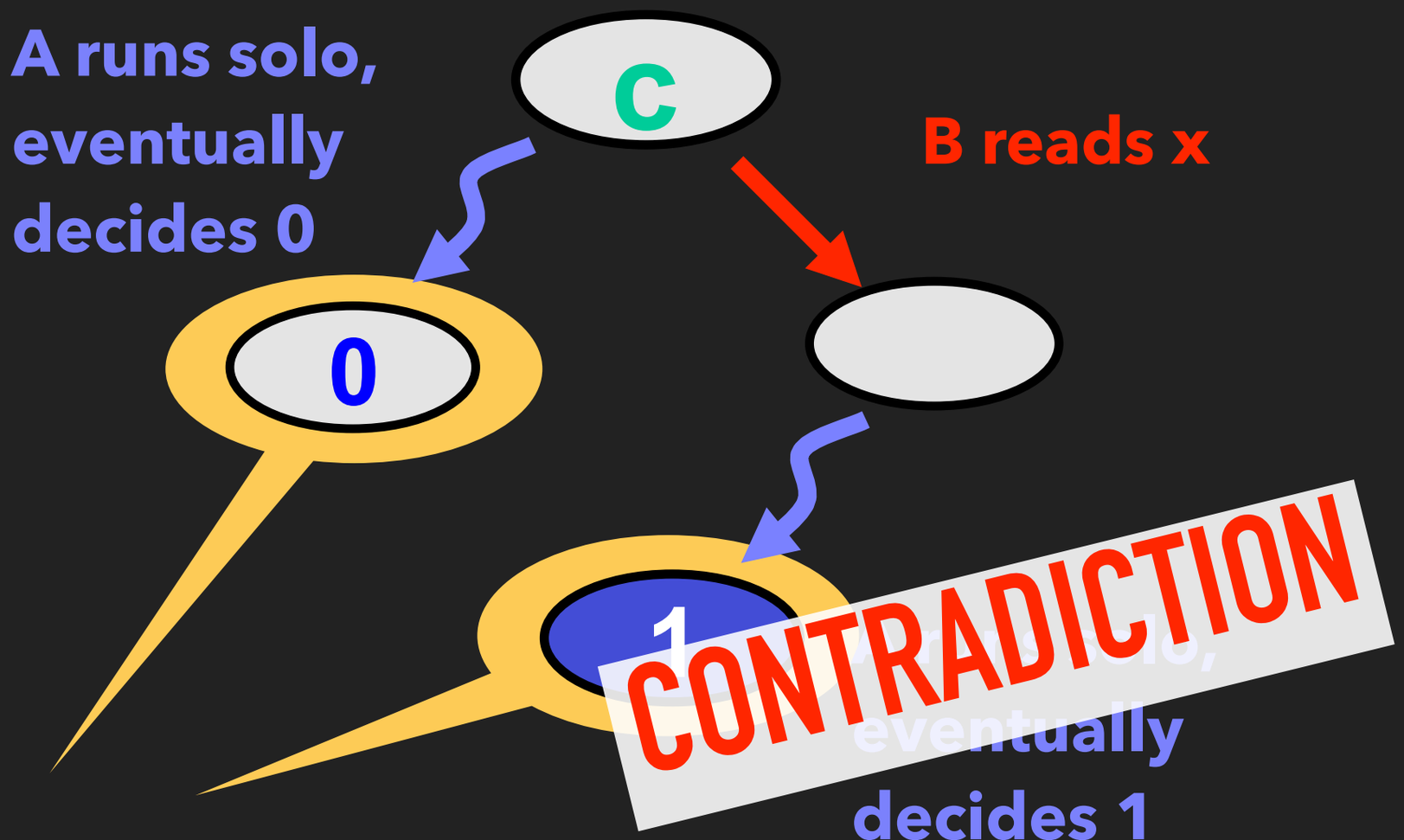
POSSIBLE INTERACTIONS

A reads x

A reads y

	x.read	y.read	x.write	y.write
x.read	?	?	?	?
y.read	?	?	?	?
x.write	?	?	?	?
y.write	?	?	?	?

SOME THREAD READS

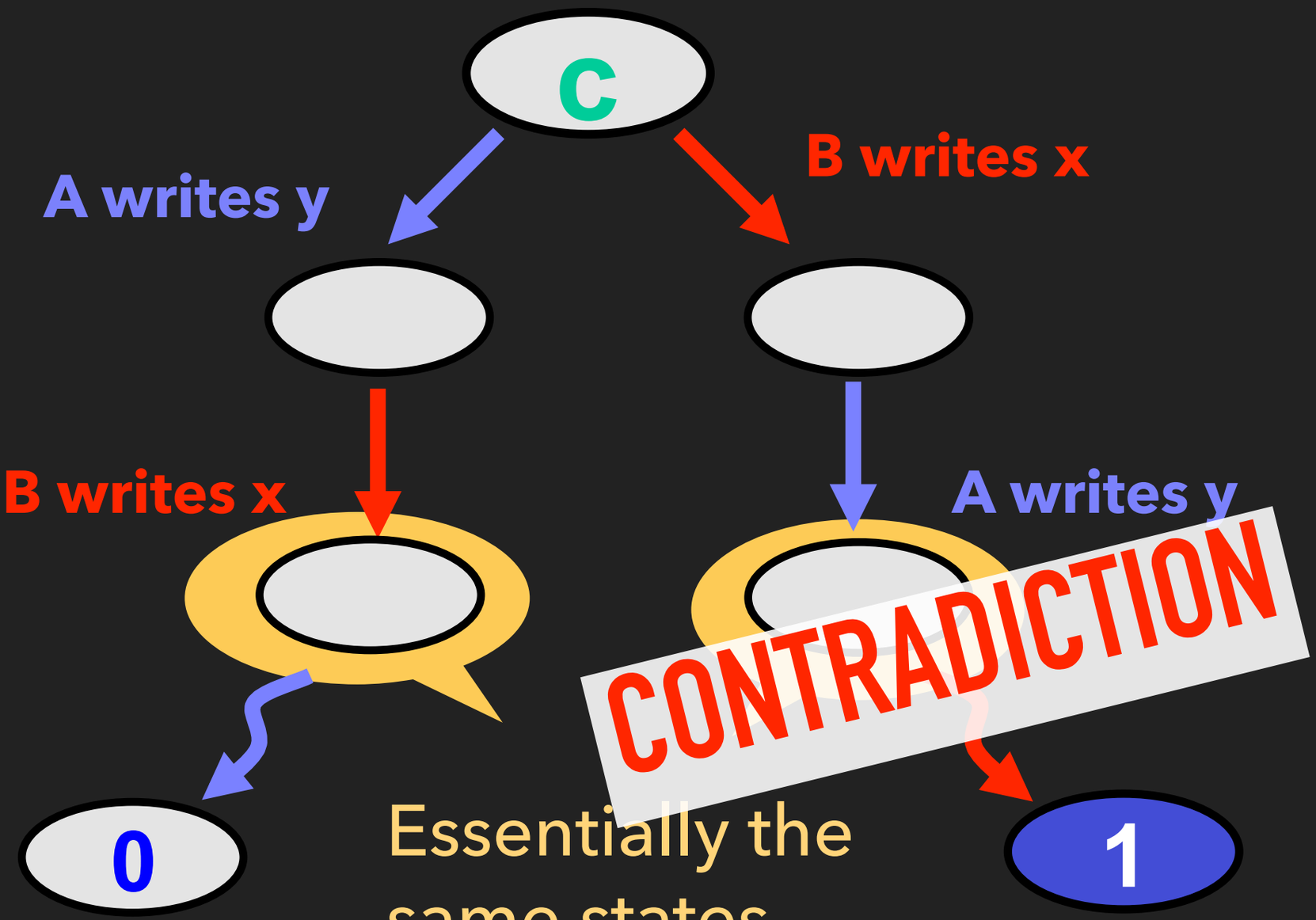


States look same to A

POSSIBLE INTERACTIONS

	x.read	y.read	x.write	y.write
x.read	no	no	no	no
y.read	no	no	no	no
x.write	no	no	?	?
y.write	no	no	?	?

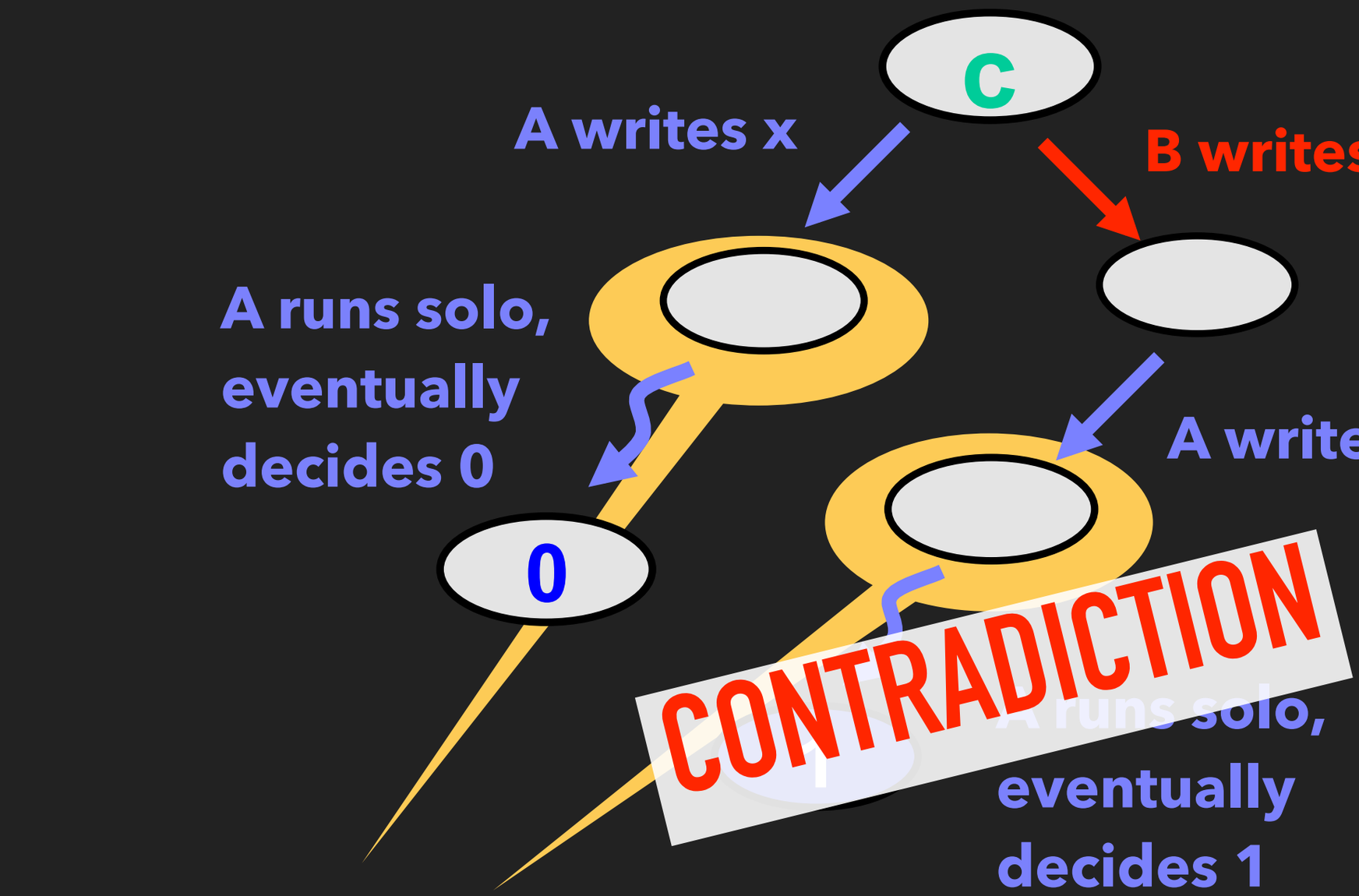
WRITING DISTINCT REGISTERS



POSSIBLE INTERACTIONS

	x.read	y.read	x.write	y.write
x.read	no	no	no	no
y.read	no	no	no	no
x.write	no	no	?	no
y.write	no	no	no	?

WRITING SAME REGISTERS



State looks the same to A

POSSIBLE INTERACTIONS

	x.read	y.read	x.write	y.write
x.read	no	no	no	no
y.read	no	no	no	no
x.write	no	no	no	no
y.write	no	no	no	no

Q.E.D.

CONSENSUS* IS IMPOSSIBLE IN A COMPLETELY ASYNCHRONOUS MESSAGE PASSING SYSTEMS WHERE EVEN A SINGLE PROCESSOR FAILS

[Fischer, Lynch, Paterson '85]

CONSENSUS* IS IMPOSSIBLE IN A COMPLETELY ASYNCHRONOUS MESSAGE PASSING SYSTEMS WHERE EVEN A SINGLE PROCESSOR FAILS

[Fischer, Lynch, Paterson '85]

WHAT ARE COMPLETELY ASYNCHRONOUS SYSTEMS

PROCESSOR ASYNCHRONY

MESSAGE ORDER ASYNCHRONY

COMMUNICATION ASYNCHRONY

Are all three types of asynchrony needed simultaneously to obtain impossibility result?

NO

[Dolev, Dwork, Stockmeyer '87]

DOLEV, DWORK, STOCKMEYER '87

Minimal synchronism needed for distributed consensus

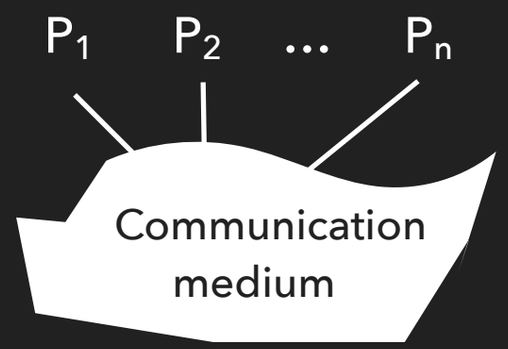
P_1 P_2 ... P_n

Communication medium

DOLEV, DWORK, STOCKMEYER '87

Processors can be synchronous or asynchronous

Communication delay can be bounded or unbounded



Messages can be ordered or unordered

Transmission mechanism can be point-to-point or broadcast

WORLD OF (IM)POSSIBILITIES

Processors	*Message Order*		*Communication*
	Unordered	Ordered	
	FLP	FLP	Unbounded
Asynchronous			
Synchronous			Bounded
			Unbounded
	Point-to-point	Broadcast	Point-to-point
	Transmission		

16 possibilites

DOLEV, DWORK, STOCKMEYER '87

- 3 cases where N-resilient protocols exist

Case 1	Processors are synchronous and communication is bounded
Case 2	Messages are ordered and transmission mechanism is broadcast
Case 3	Processors are synchronous and messages are ordered

WORLD OF (IM)POSSIBILITIES

Processors	*Message Order*		*Communication*
	Unordered	Ordered	
Asynchronous	FLP	FLP	Unbounded
Synchronous	YES	YES	Bounded
	YES	YES	Unbounded
	Point-to-point	Broadcast	Point-to-point
	Transmission		

USE TIMEOUTS

Processors are synchronous and communication is bounded

WORLD OF (IM)POSSIBILITIES

DECIDE ON THE FIRST RECEIVED MESSAGE

Processors	Message Order	Communication	Transmission
Asynchronous	Unordered	Unbounded	Point-to-point
			Broadcast
			Point-to-point
		Bounded	Point-to-point
			Broadcast
			Point-to-point
Synchronous	Unordered	Unbounded	Point-to-point
			Broadcast
			Point-to-point
	Ordered	Unbounded	Point-to-point
			Broadcast
			Point-to-point
		Bounded	Point-to-point
			Broadcast
			Point-to-point
		Unbounded	Point-to-point
			Broadcast
			Point-to-point

Unordered		Ordered	
FLP	FLP	YES	
		YES	
YES	YES	YES	YES
		YES	

Messages are ordered and transmission mechanism is broadcast

WORLD OF (IM)POSSIBILITIES

EXPONENTIAL NUMBER OF MESSAGES

Processors	*Message Order*	*Communication*
Asynchronous	FLP	Unbounded
Synchronous	FLP	Unbounded
Asynchronous	YES	Unbounded
Synchronous	YES	Unbounded
Asynchronous	YES	Bounded
Synchronous	YES	Bounded
Asynchronous	YES	Unbounded
Synchronous	YES	Unbounded

Transmission
Point-to-point
Broadcast
Point-to-point

Processors are synchronous and messages are ordered

MORE RESULTS: [DOLEV, DWORK, STOCKMEYER '87]

- 3 cases where N-resilient protocols exist
- No t-resilient protocol if system weakened for t = 1,2
  - Favourable -> Unfavourable

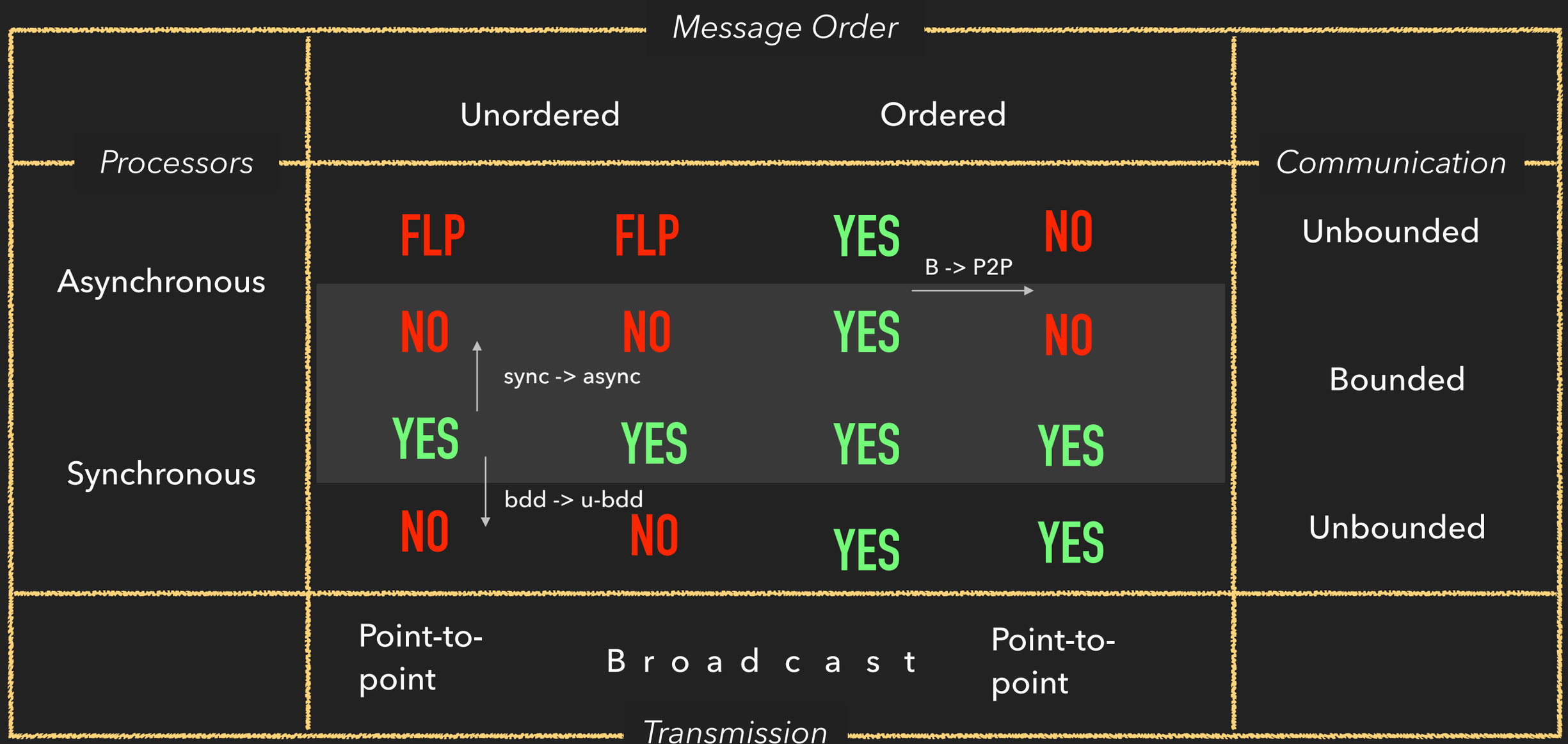
Processors : synchronous / asynchronous

Communication : bounded / unbounded

Messages : ordered / unordered

Transmission : broadcast / point-to-point

WORLD OF (IM)POSSIBILITIES



LAST LESSON BEFORE FIGHTING THE BEAST : PROBABILITY ROCKS

- Probabilistic consensus possible in presence of faults
  - do not terminate wlp
  - if terminate, right result
  - decisions not consistent wlp

PLOTTING A BYZANTINE AGREEMENT

BYZANTINE GENERALS PROBLEM

[Lamport, Shostak, Pease '82]

BYZANTINE GENERALS PROBLEM

Attack

[A, R, A, R, A]

Retreat

[A, R, A, R, A]

Attack

[A, R, A, R, A]

Retreat

[A, R, A, R, A]

Attack

[A, R, A, R, A]

Choose the majority vote

Reach consensus

BYZANTINE GENERALS PROBLEM

Attack

~~[A, R, A, R, A]~~

Do not care what bz decides

Retreat

[A, R, A, R, A]

Attack

[A, R, A, R, A]

Retreat

[A, R, A, R, A]

Attack

[A, R, A, R, A]

BYZANTINE GENERALS PROBLEM

Attack

[A, R, A, R, A]

Do not care what bz decides

Retreat

[R, R, A, R, A] Retreat

Consensus violated

Attack

[R, R, A, R, A] Retreat

Retreat

[A, R, A, R, A] Agree

Attack

[A, R, A, R, A] Agree

BYZANTINE GENERALS PROBLEM

- Is there a protocol such that

CONSISTENCY

All honest parties agree on same value

VALIDITY

Agreed-upon value is input to some honest party

TERMINATION

Each honest party decides in finite number of steps

WORLD OF (IM)POSSIBILITIES

Processors	*Message Order*				*Communication*
	Unordered		Ordered		
Asynchronous	FLP	FLP	YES	NO	Unbounded
Synchronous	NO	NO	YES	NO	Bounded
	YES	YES	YES	YES	
	NO	NO	YES	YES	Unbounded
	Point-to-point	Broadcast		Point-to-point	
	Transmission				

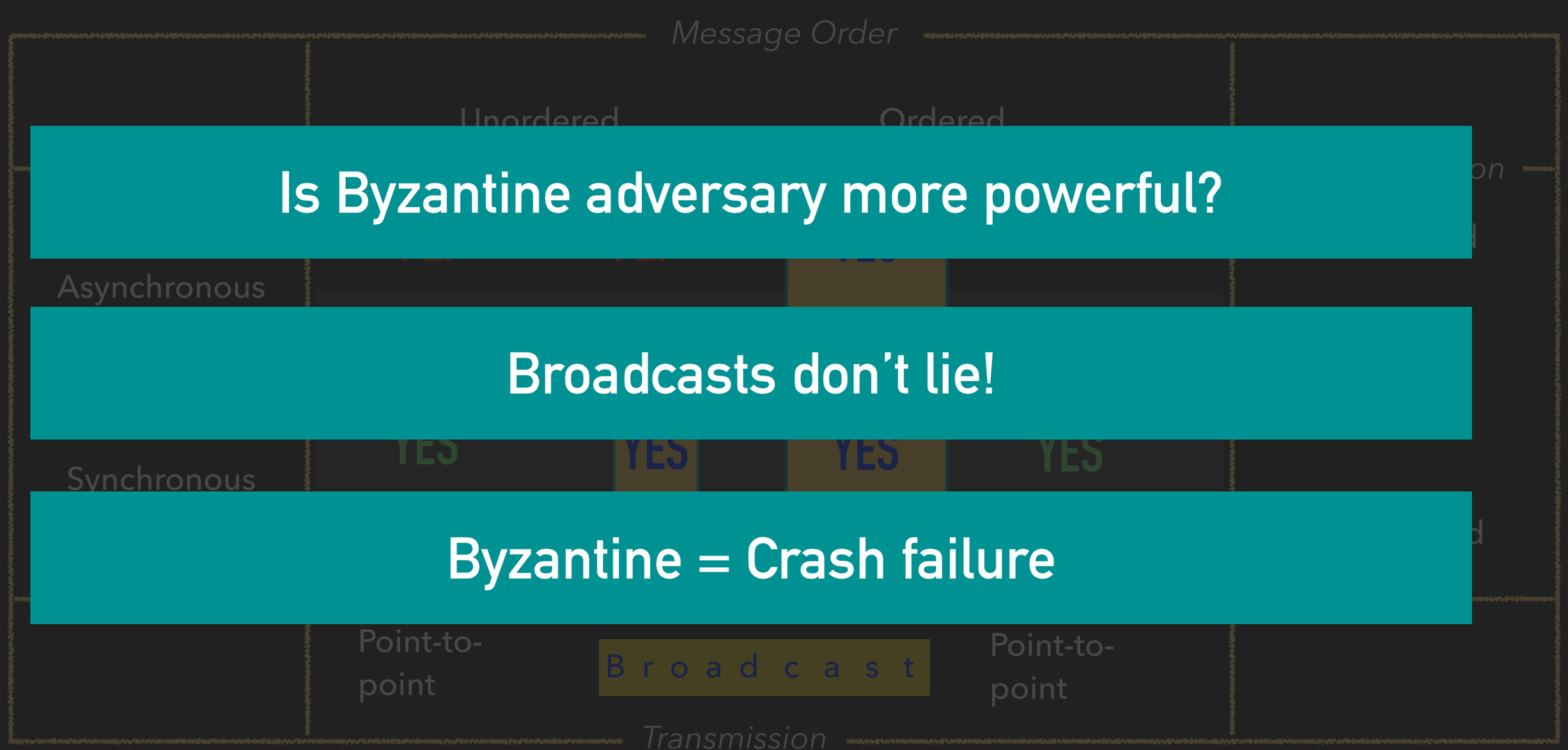
Impossible in CF model => Impossible in Byzantine model

WORLD OF (IM)POSSIBILITIES

Processors	Message Order: Unordered	Message Order: Unordered	Message Order: Ordered	Message Order: Ordered	*Communication*
Asynchronous	FLP	FLP	YES	NO	Unbounded
	NO	NO	YES	NO	Bounded
Synchronous	YES	YES	YES	YES	Unbounded
	NO	NO	YES	YES	
	Point-to-point	Broadcast		Point-to-point	

Transmission

WORLD OF (IM)POSSIBILITIES



Is Byzantine adversary more powerful?

Broadcasts don't lie!

Byzantine = Crash failure

WORLD OF (IM)POSSIBILITIES

Processors	*Message Order*				*Communication*
	Unordered		Ordered		
Asynchronous	FLP	FLP	YES	NO	Unbounded
	NO	NO	YES	NO	Bounded
Synchronous	YES	YES	YES	YES	Bounded
	NO	NO	YES	YES	Unbounded
	Point-to-point	Broadcast		Point-to-point	
	Transmission				

BYZANTINE CONSENSUS IS IMPOSSIBLE WHEN NO AUTHENTICATION IS POSSIBLE IF T >= 1/3 N

[Lamport, Shostak, Pease '82]

IMPOSSIBILITY PROOF [Fischer, Lynch, Meritt '86]

Assume otherwise …

Reason about the properties of any such protocol

Derive a contradiction

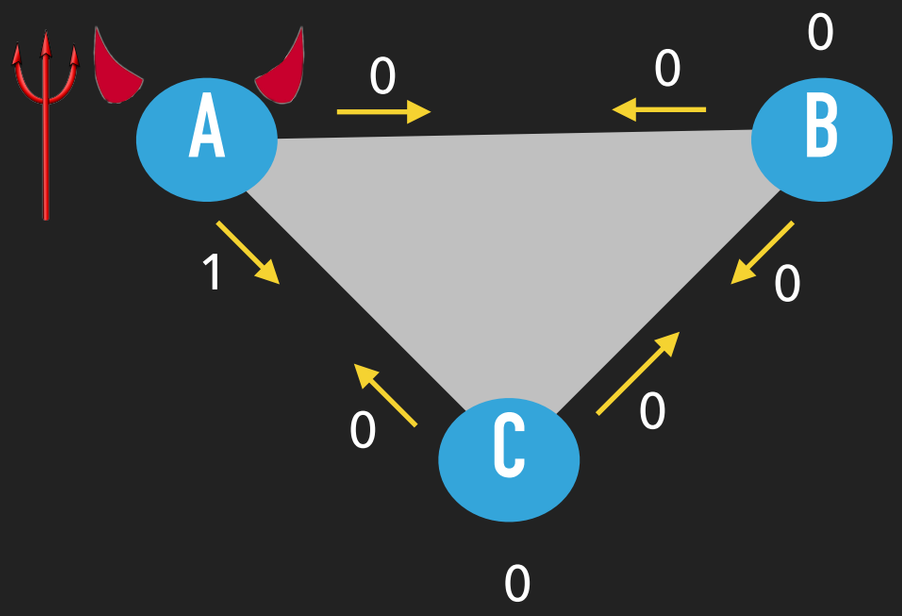
Quod

Erat

Demonstrandum

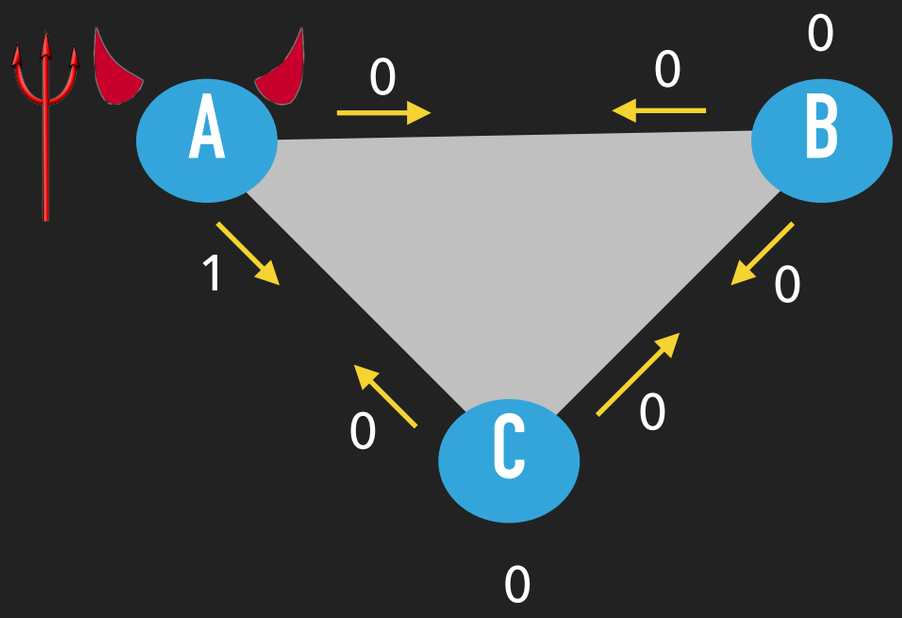
Show for n = 3 and t = 1

IMPOSSIBILITY PROOF (HIGH LEVEL)

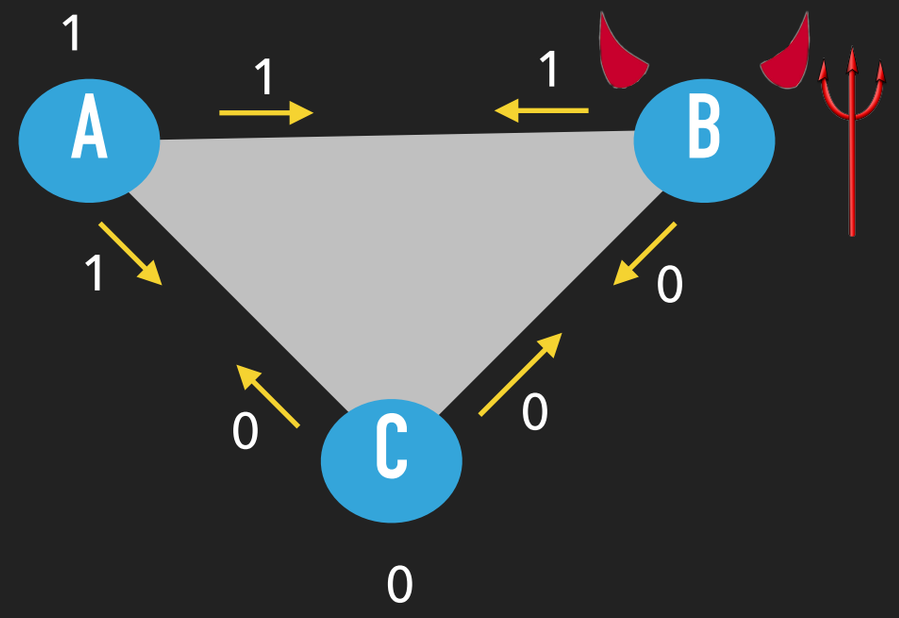


validity: Decide 0

IMPOSSIBILITY PROOF (HIGH LEVEL)

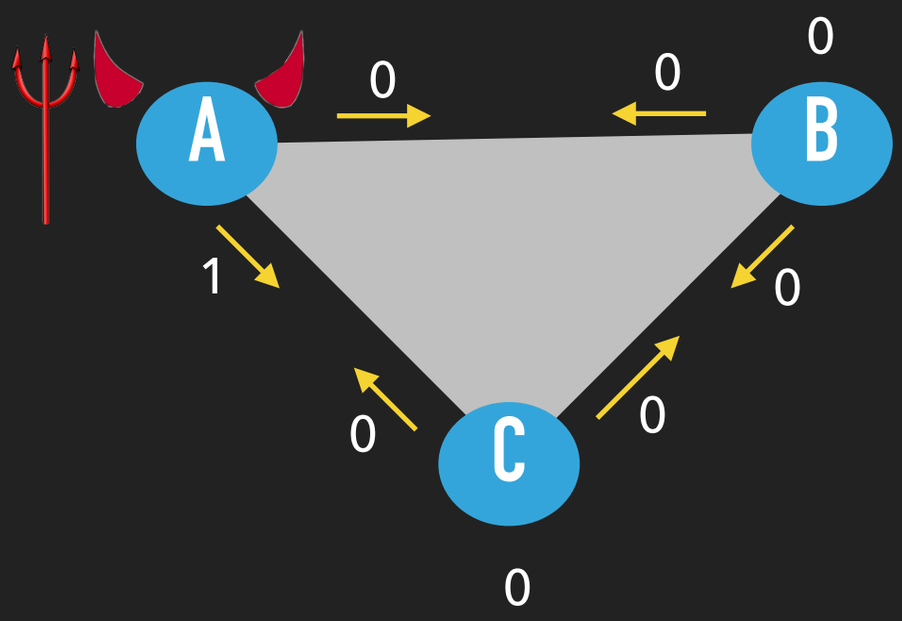


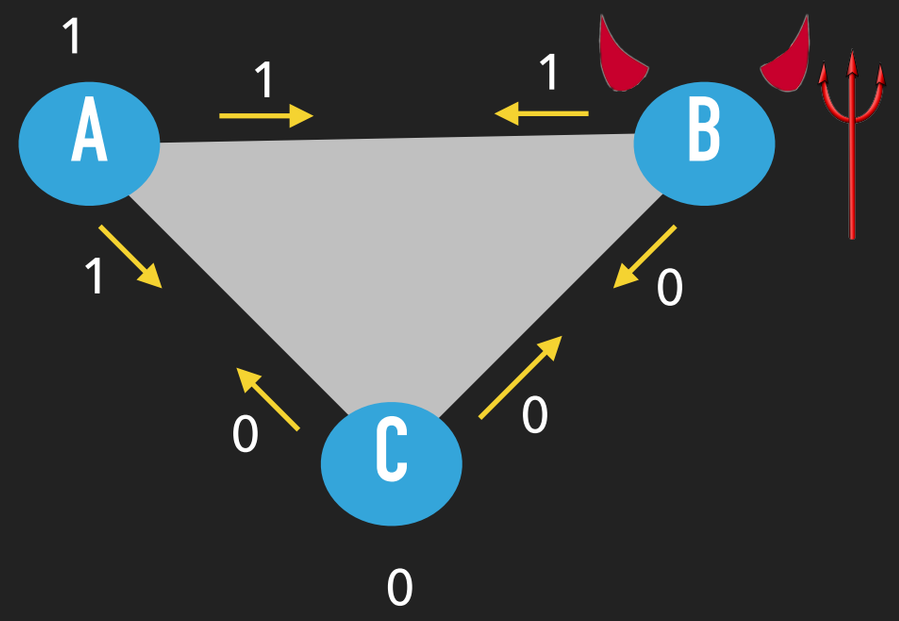
validity: Decide 0



Decide 0 : View of C is indistinguishable from Case 1

IMPOSSIBILITY PROOF (HIGH LEVEL)





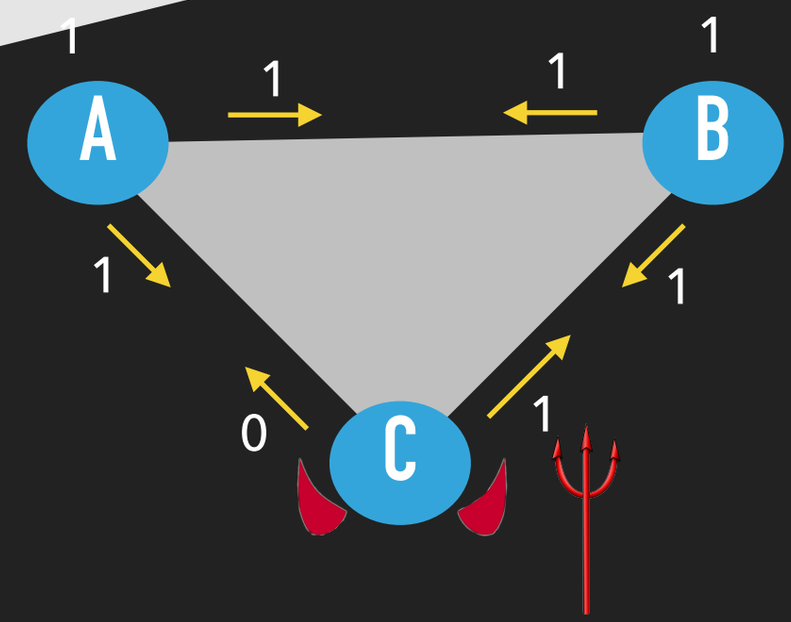
validity: Decide 0

validity: Decide 1

A decides 0: View indistinguishable from Case 2

CONTRADICTION

Decide 0 : View of C is indistinguishable from Case 1

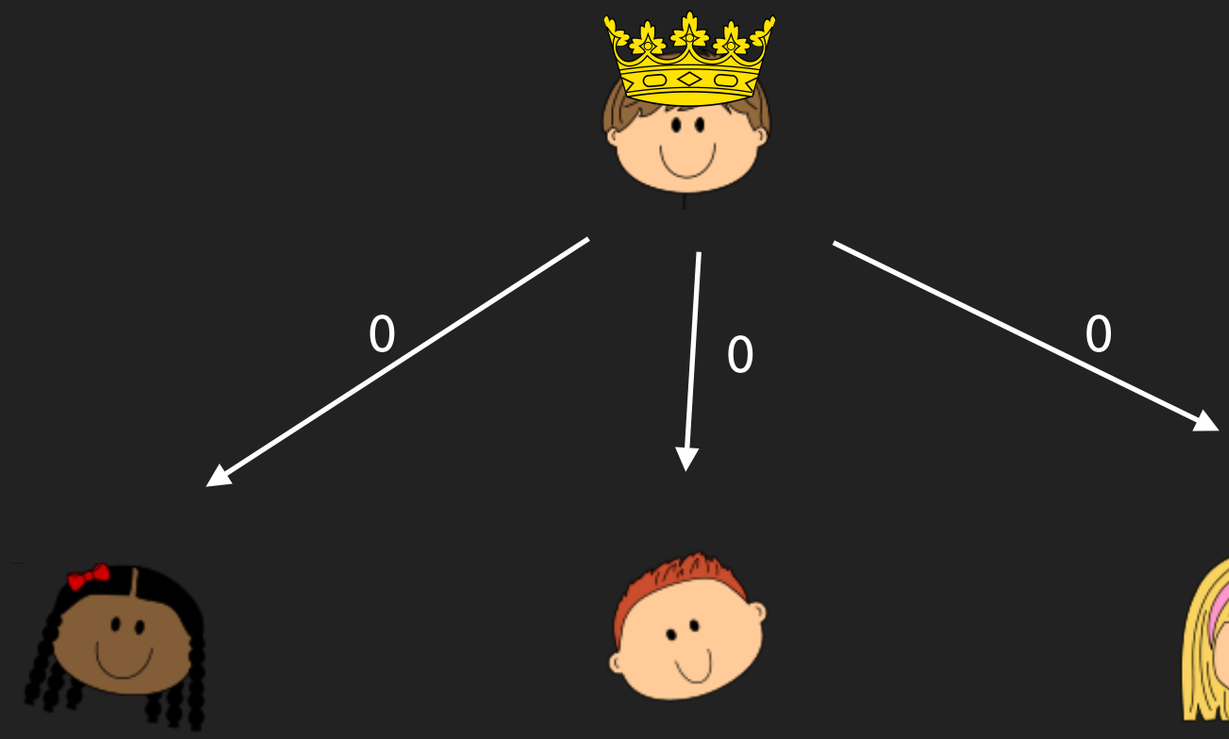


WORLD OF (IM)POSSIBILITIES

Processors	*Message Order*				*Communication*
	Unordered		Ordered		
Asynchronous	FLP	FLP	YES	NO	Unbounded
	NO	NO	YES	NO	Bounded
Synchronous	YES*	YES	YES	YES*	Bounded
	NO	NO	YES	YES*	Unbounded
	Point-to-point	Broadcast		Point-to-point	
	Transmission				

ALGORITHM IDEA WHEN T < 1/3 N

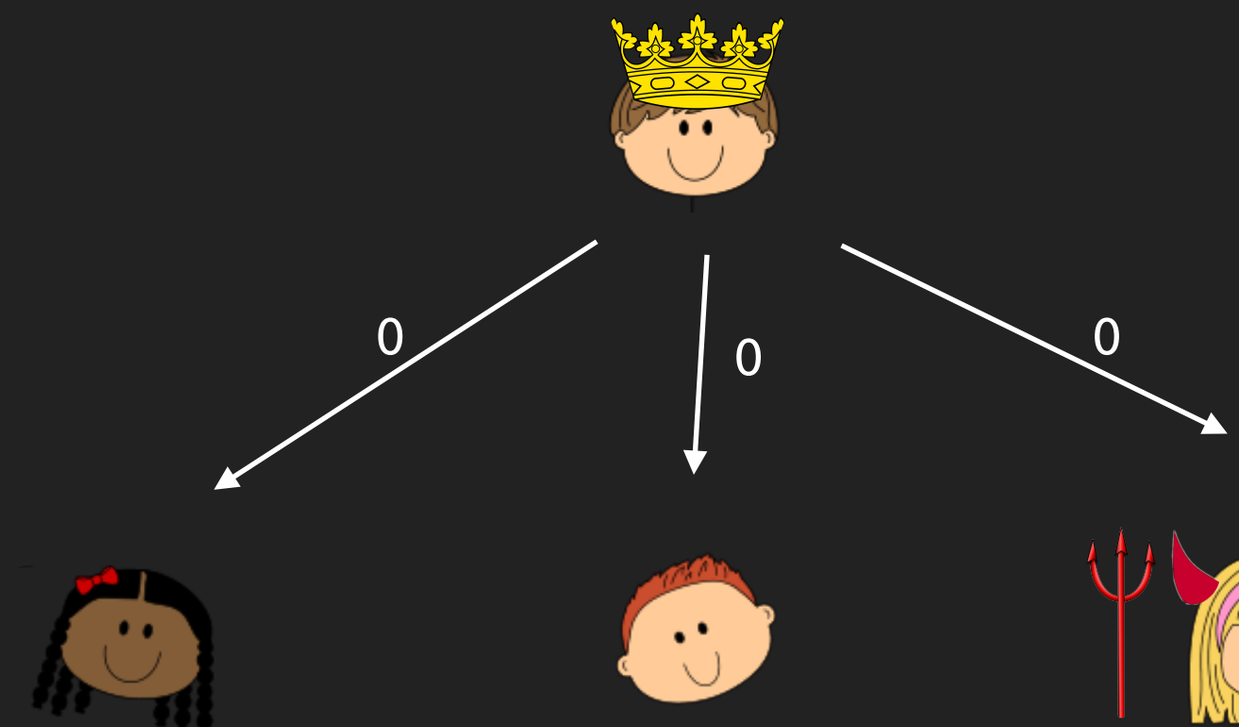
- Case n = 4 , t = 1
- Choose a leader
- Listen to what he says
- Talk to others
  - See what they say about the leader said
- Decide on the majority value



$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

0 0 0

ALGORITHM IDEA WHEN T < 1/3 N



Case 1

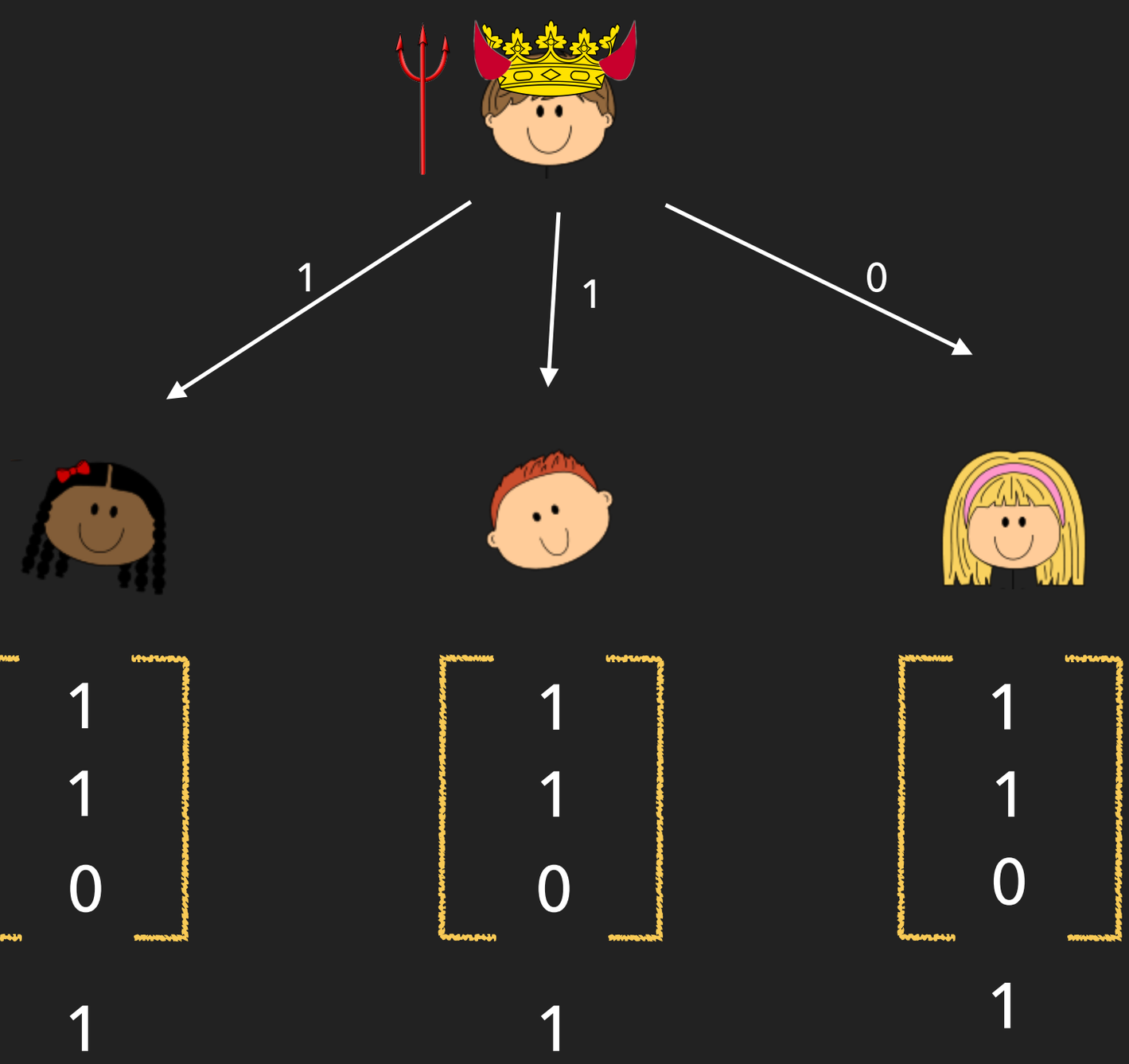
CONSISTENCY

VALIDITY

$$\begin{bmatrix} 0 \\ 0 \\ ? \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ ? \end{bmatrix} \quad \begin{bmatrix} ? \\ ? \\ 0 \end{bmatrix}$$

0 0

ALGORITHM IDEA WHEN T < 1/3 N



Case 2

CONSISTENCY

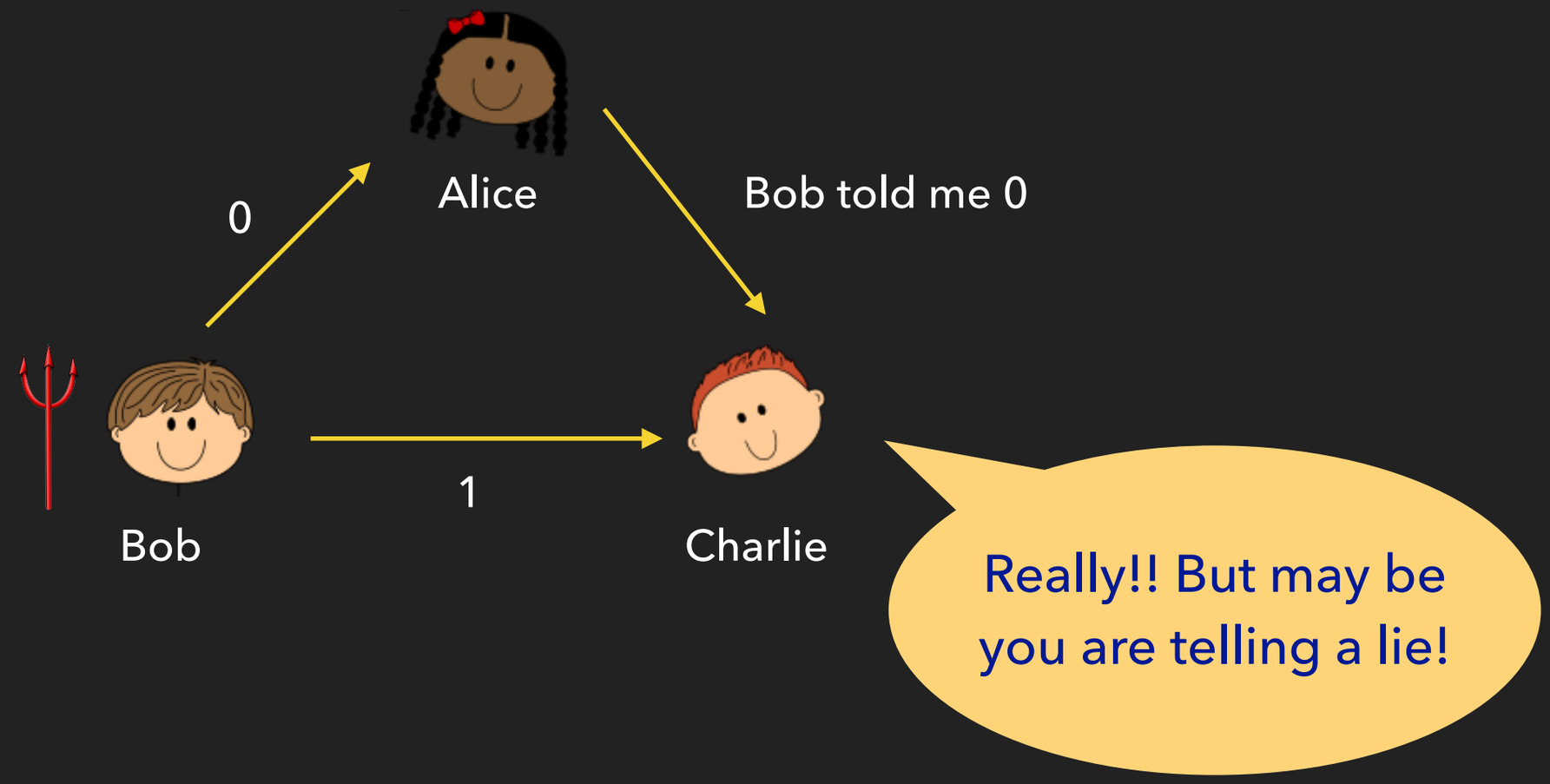
VALIDITY

BYZANTINE CONSENSUS IS IMPOSSIBLE WHEN NO AUTHENTICATION IS POSSIBLE IF T >= 1/3 N

[Lamport, Shostak, Pease '82]

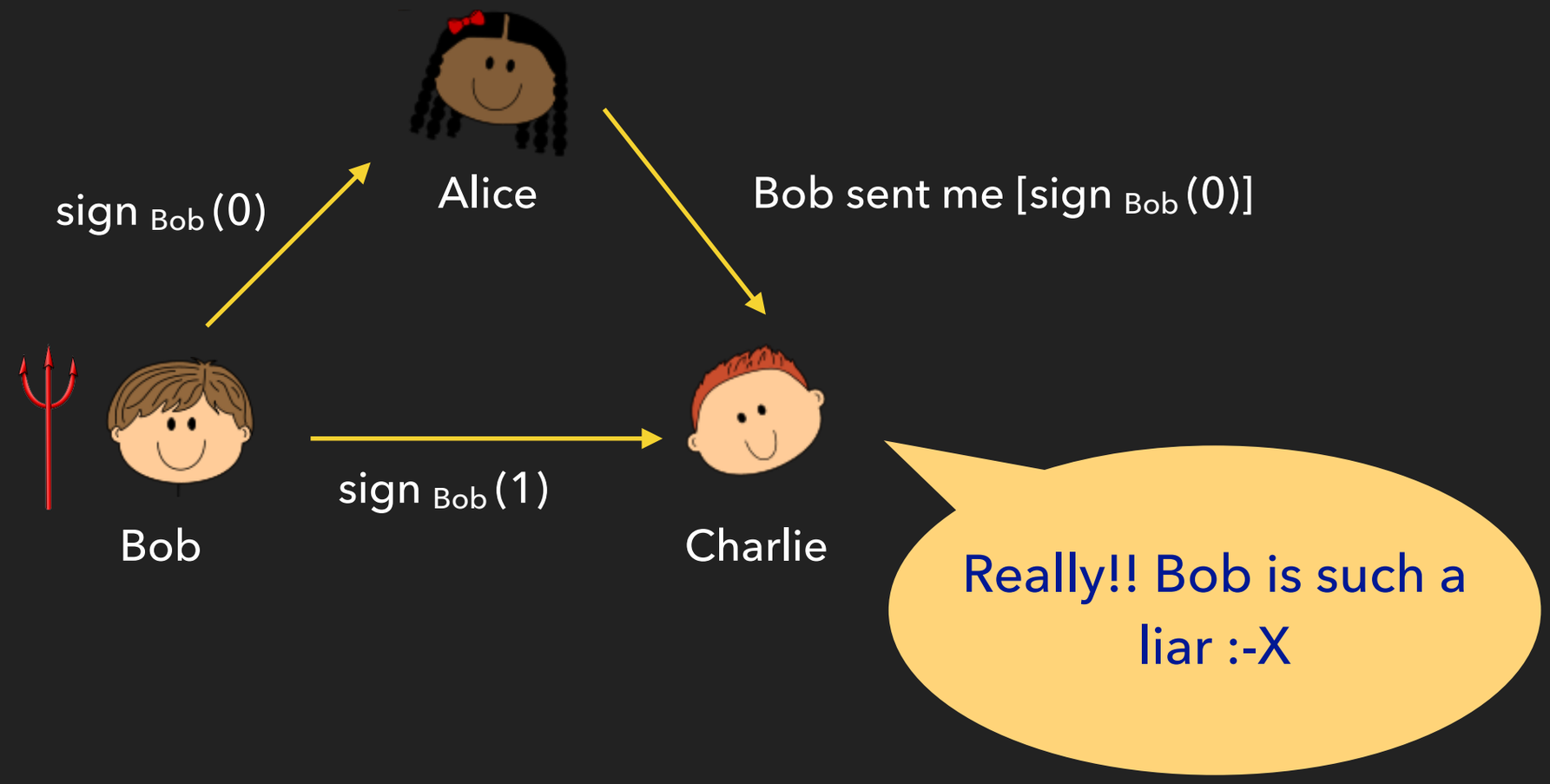
HE SAID, SHE SAID

- Malicious can send conflicting messages to other two



HE SAID, SHE SAID

- Malicious can send conflicting messages to other two



SIGN ON THE DOTTED LINE [Lamport, Shostak, Pease '82]

- Use authenticated messages
- Give byzantine consensus protocol
  - Remember the model (sync processors + bdd delay)
- Use cryptographic techniques to sign

THE ADVANCES IN CONSENSUS : SANDS OF TIME

Bz consensus is impossible unless t < n/3 if no crypto

Protocol using crypto

Sync + p-to-p + bdd delay

Probabilistic consensus protocol

CF consensus is impossible for t = 1

Complete asynchronous

Minimum synchronisation for CF consensus

Different models

CF consensus is impossible for t = 1 in RW SM

Asynchronous processors

MP = SM for t < 1/2

Proof of stake

Practical models

TAKEAWAYS

Crash failure consensus is hard

Byzantine is even harder

Crypto is cool

that's why we have been doing it

Distributed + crypto is even cooler

that's why we will be doing it

Questions?

REFERENCES

- Crash failure impossibility proof from lecture slides of Maurice Herlihy and Nir Shavit
- Many Faces of Consensus in Distributed Systems by John Turek and Dennis Shasha

OVERVIEW