

Equivalence and Separation of Asynchronous and Heard-Of Message-Passing Models

Joint work with Hagit Attiya (Technion), Armando Castañeda (UNAM),
and Thomas Nowak (Université Paris-Saclay)

Dhrubajyoti Ghosh
Université Paris-Saclay, France

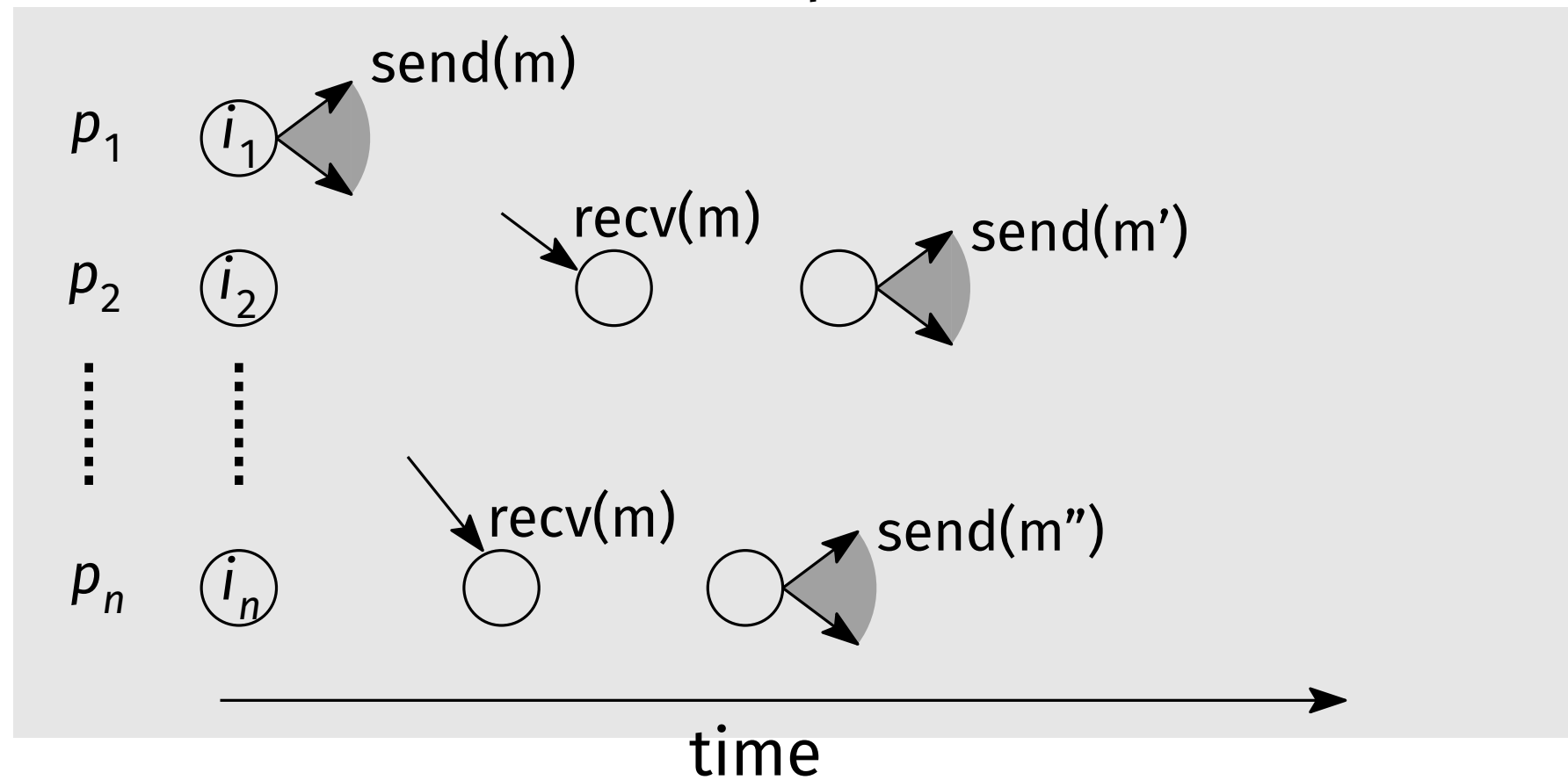
Preliminaries

- ☞ Complete message-passing network with n processes
- ☞ Every process starts with a given input as part of initial state

Preliminaries

- ☞ Complete message-passing network with n processes
- ☞ Every process starts with a given input as part of initial state

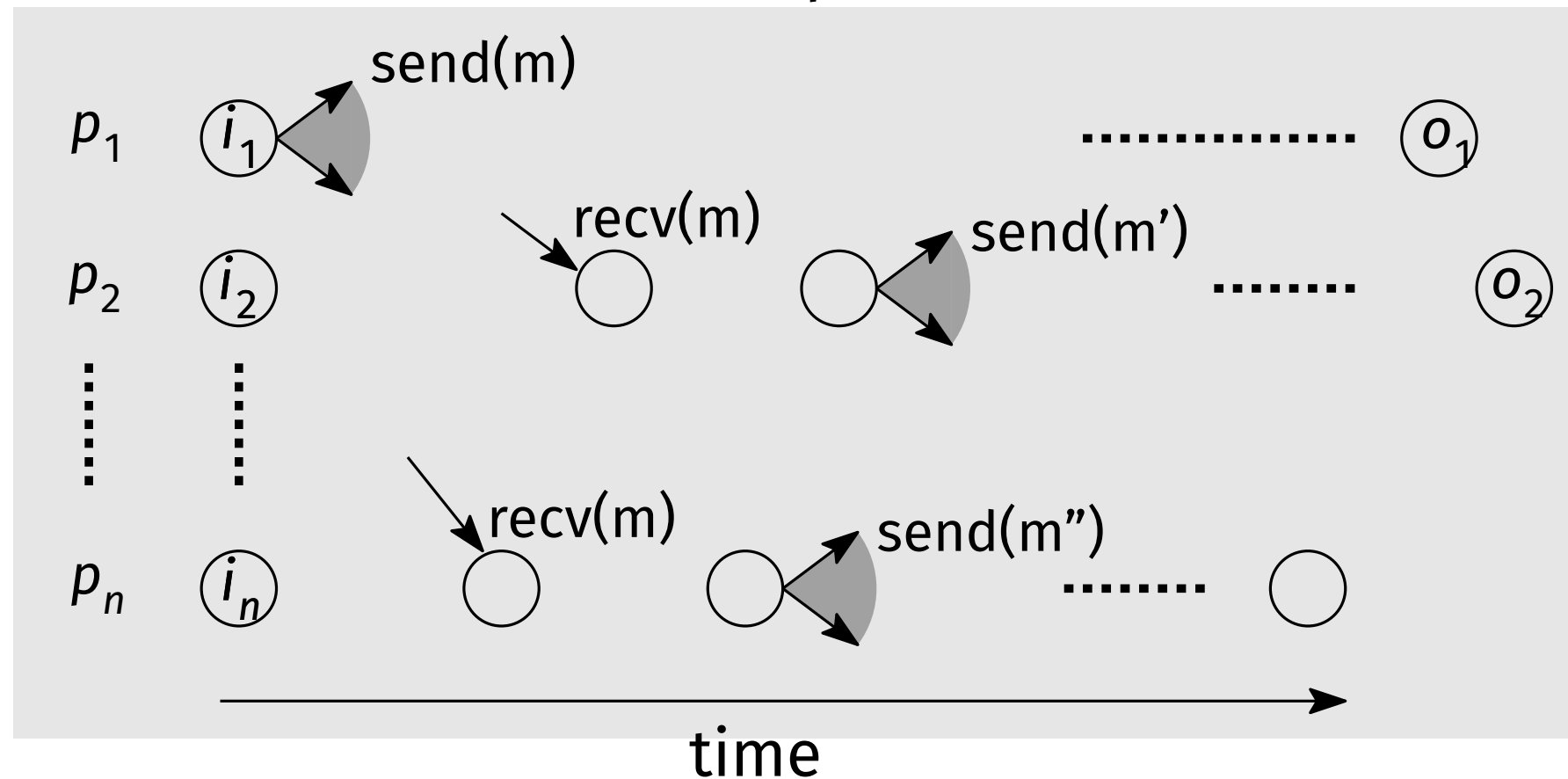
A run of a protocol:



Preliminaries

- ☞ Complete message-passing network with n processes
- ☞ Every process starts with a given input as part of initial state

A run of a protocol:

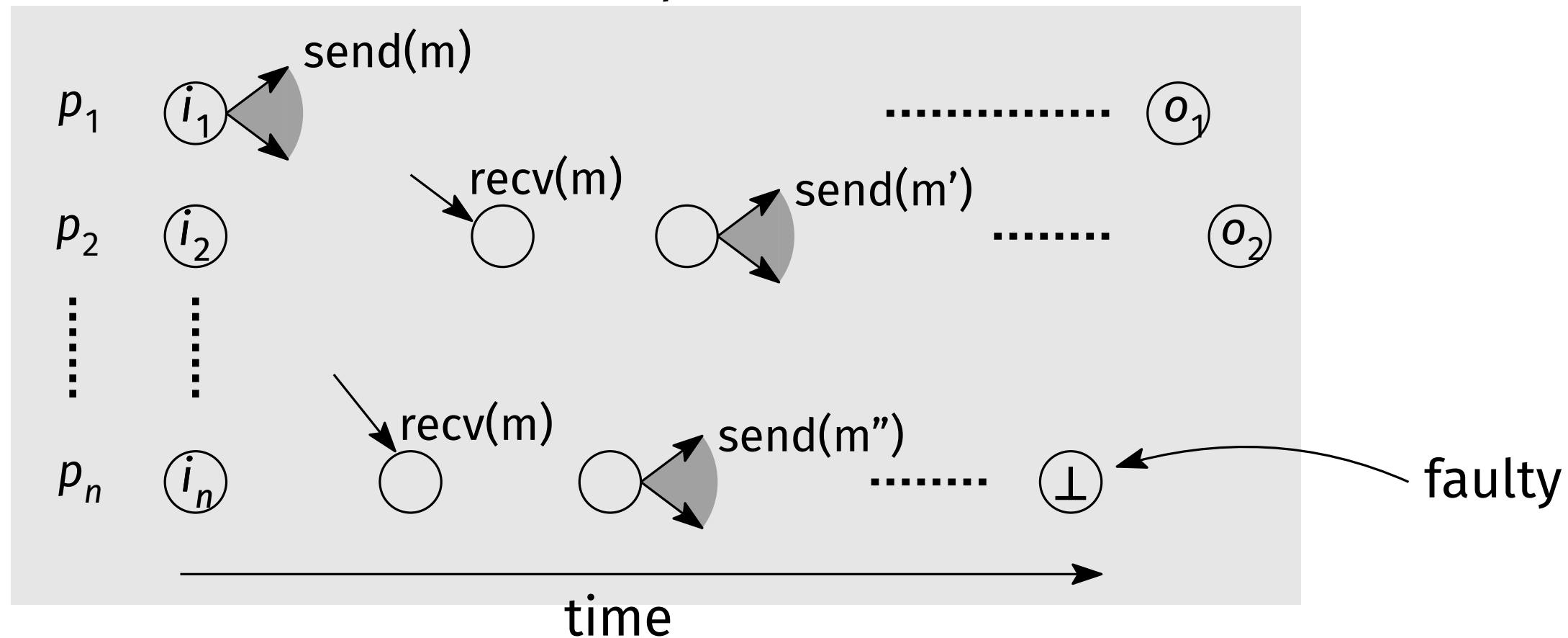


- ☞ It can decide on an output value *at most once*.

Preliminaries

- ☞ Complete message-passing network with n processes
- ☞ Every process starts with a given input as part of initial state

A run of a protocol:



- ☞ It can decide on an output value *at most once*.
- ☞ Every model designates certain processes as *faulty*. Faulty processes need not decide (output denoted by \perp).

Preliminaries

Decision Tasks

Tasks specify which combinations of inputs and outputs are allowed in a run.

Definition (Task)

A task T is a triple $(\mathcal{I}, \mathcal{O}, \Delta)$ such that

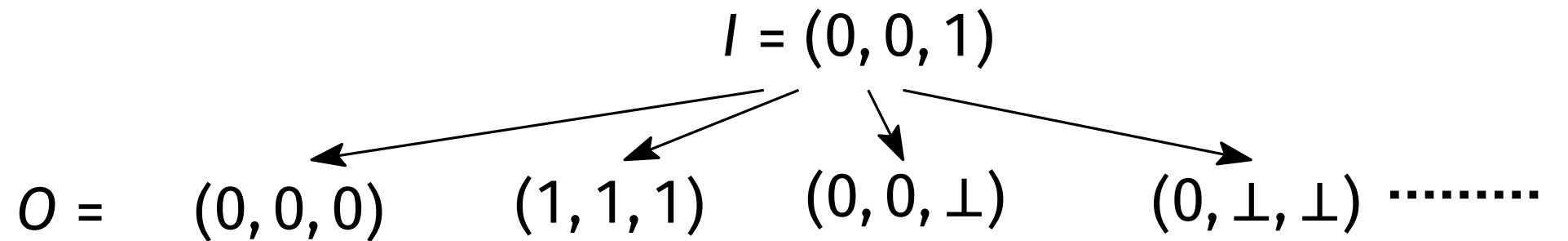
- 👉 \mathcal{I} is the set of all possible input vectors
- 👉 \mathcal{O} is the set of all possible output vectors
- 👉 $\Delta : \mathcal{I} \rightarrow 2^{\mathcal{O}}$ specifies the allowed output vectors for each input vector

Also, \mathcal{O} is closed under replacing any element of an output vector by \perp .

Binary consensus task:

Input vector: any combination of 0 and 1

Output vector: all non- \perp outputs must be equal (+validity).

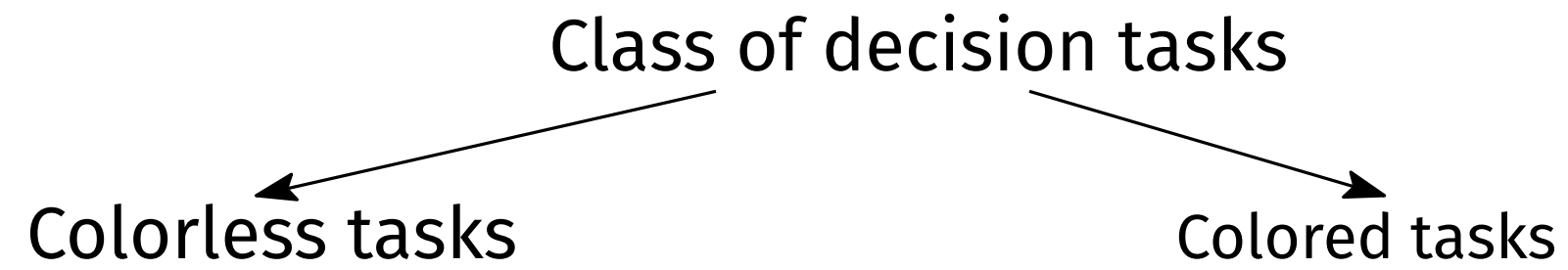


Definition (Task Solvability)

Protocol \mathcal{P} solves task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ in model M if for each input vector I , the resulting output vector O satisfies $(I, O) \in \Delta$.

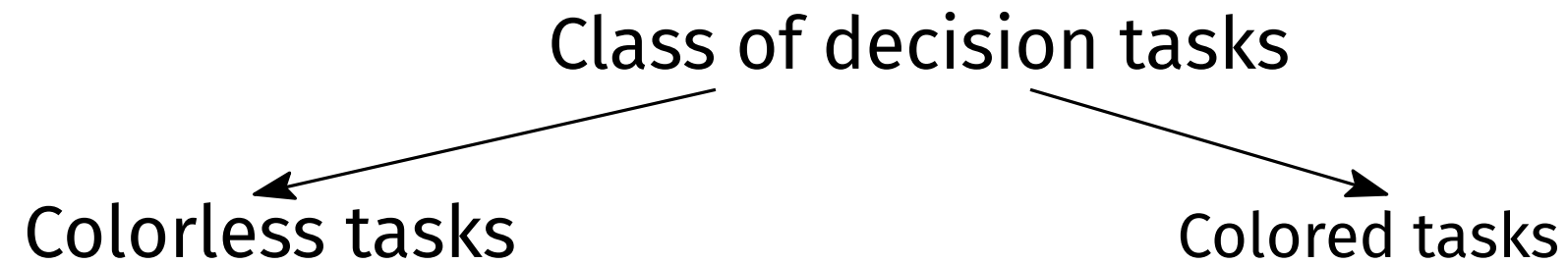
Preliminaries

Tasks



Preliminaries

Tasks



Colorless tasks

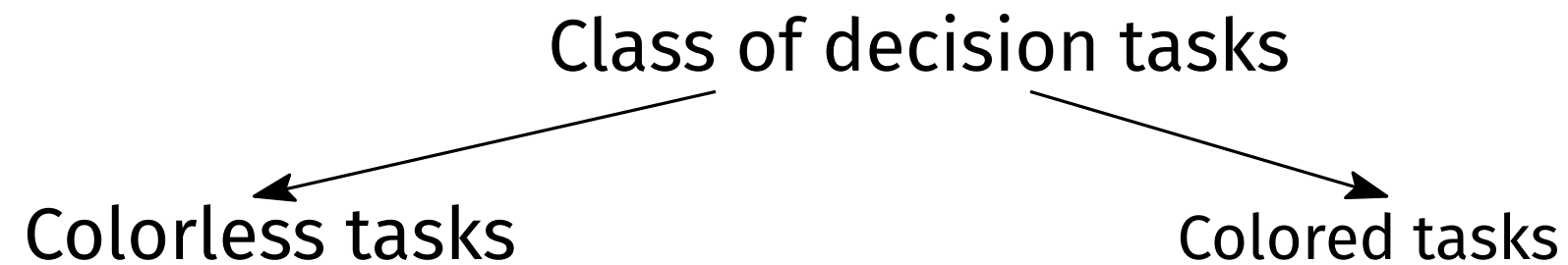
Processes can copy each other's inputs and outputs (if they know about them)

E.g. Binary Consensus.

$O = (1, 1, \perp, 1, 1)$ Valid output vector $O' = (1, 1, 1, 1, 1)$ Also valid

Preliminaries

Tasks



Colorless tasks

Processes can copy each other's inputs and outputs (if they know about them)

E.g. Binary Consensus.

$$O = (1, 1, \perp, 1, 1) \text{ Valid output vector} \quad O' = (1, 1, 1, 1, 1) \text{ Also valid}$$

Colored tasks

Any task that is not colorless.

E.g. Renaming task, where all non- \perp outputs must be distinct.

$$O = (2, 3, \perp, 5, 7) \text{ Valid output vector} \quad O' = (2, 3, 7, 5, 7) \text{ Invalid}$$

Preliminaries

Models of Computation

Asynchronous Message Passing model¹ with f faults (AMP_f):

- Sends are broadcasts
- Messages take finite but unbounded time to reach recipients
- At most f processes may crash

¹ Fischer, Lynch, Paterson. Impossibility of distributed consensus with one faulty process. J. ACM, 32(2):374–382, apr 1985.

Preliminaries

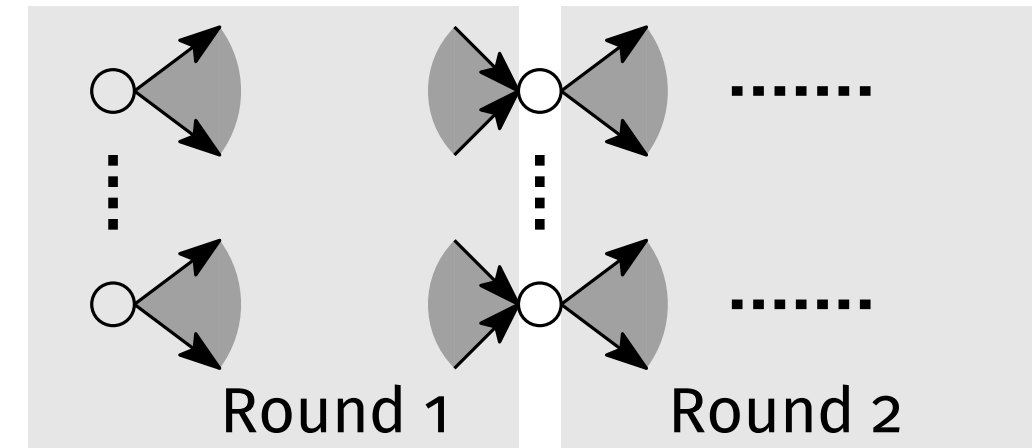
Models of Computation

Asynchronous Message Passing model¹ with f faults (AMP_f):

- ➡ Sends are broadcasts
- ➡ Messages take finite but unbounded time to reach recipients
- ➡ At most f processes may crash

Heard-Of model² with parameter f (HO_f):

➡ Round-based computation



¹ Fischer, Lynch, Paterson. Impossibility of distributed consensus with one faulty process. J. ACM, 32(2):374–382, apr 1985.

² B. Charron-Bost, A. Schiper. The Heard-Of model: computing in distributed systems with benign faults. Distributed Computing, 22(1):49–71, April 2009.

Preliminaries

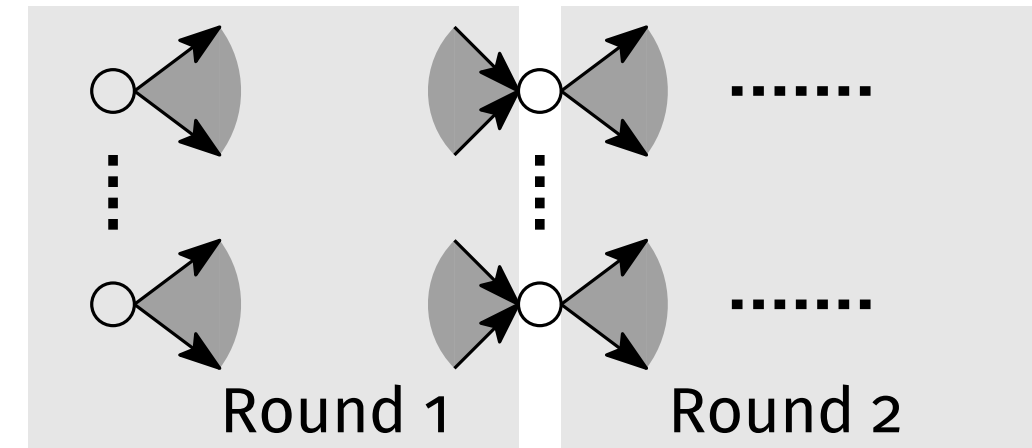
Models of Computation

Asynchronous Message Passing model¹ with f faults (AMP_f):

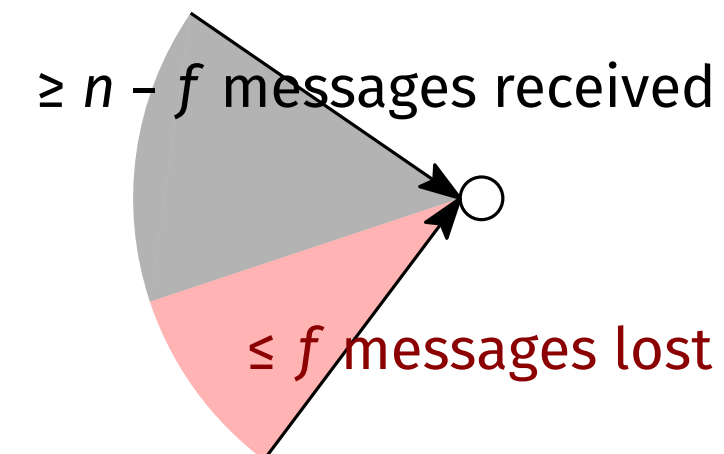
- Sends are broadcasts
- Messages take finite but unbounded time to reach recipients
- At most f processes may crash

Heard-Of model² with parameter f (HO_f):

➤ Round-based computation



➤ Each process may lose up to f messages from others (it always receives its own)



¹ Fischer, Lynch, Paterson. Impossibility of distributed consensus with one faulty process. J. ACM, 32(2):374–382, apr 1985.

² B. Charron-Bost, A. Schiper. The Heard-Of model: computing in distributed systems with benign faults. Distributed Computing, 22(1):49–71, April 2009.

Preliminaries

Models of Computation

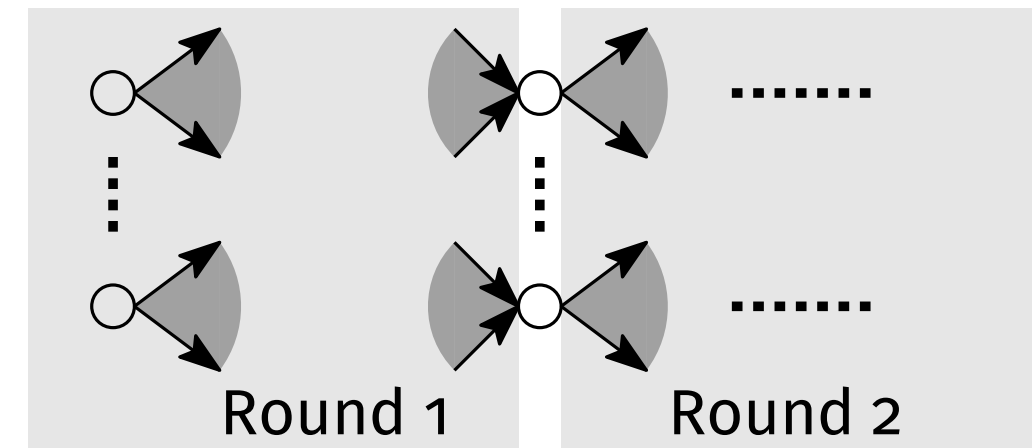
Asynchronous Message Passing model¹ with f faults (AMP_f):

- ☞ Sends are broadcasts
- ☞ Messages take finite but unbounded time to reach recipients
- ☞ At most f processes may crash

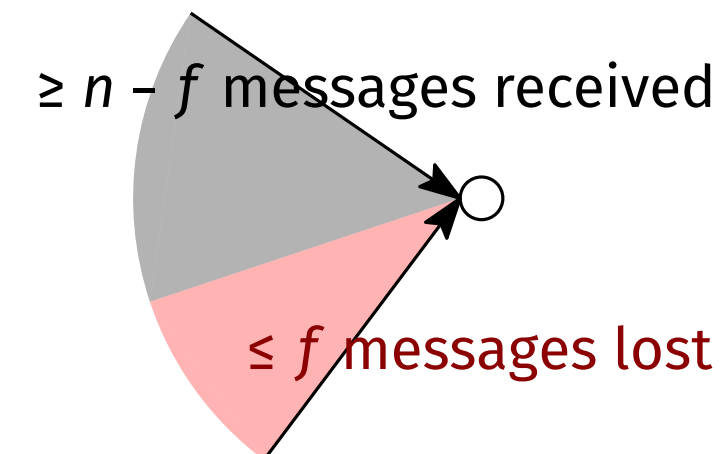
Remark. A process is faulty in AMP_f iff it is crashed.

Heard-Of model² with parameter f (HO_f):

☞ Round-based computation



☞ Each process may lose up to f messages from others (it always receives its own)



Remark. No process is defined to be faulty.

¹ Fischer, Lynch, Paterson. Impossibility of distributed consensus with one faulty process. J. ACM, 32(2):374–382, apr 1985.

² B. Charron-Bost, A. Schiper. The Heard-Of model: computing in distributed systems with benign faults. Distributed Computing, 22(1):49–71, April 2009.

Problem Studied

Statement

Can constructing a virtual round structure in AMP_f always help in solving tasks?

or, alternatively

Do AMP_f and HO_f solve the same set of decision tasks?

Problem Studied

Statement

Can constructing a virtual round structure in AMP_f always help in solving tasks?

or, alternatively

Do AMP_f and HO_f solve the same set of decision tasks?

	Colorless tasks	Colored tasks
$f \leq 1 (n \geq 3)$	Yes (Gafni, Losa 2023) ¹	
$1 < f < n/2$		

Question. Do AMP_f and HO_f coincide?

¹ Eli Gafni and Giuliano Losa. Time Is Not a Healer, but It Sure Makes Hindsight 20:20, Stabilization, Safety, and Security of Distributed Systems, 2023

Problem Studied

Statement

Can constructing a virtual round structure in AMP_f always help in solving tasks?

or, alternatively

Do AMP_f and HO_f solve the same set of decision tasks?

	Colorless tasks	Colored tasks
$f \leq 1 (n \geq 3)$	Yes (Gafni, Losa 2023) ¹	Yes ²
$1 < f < n/2$	Yes ²	No ²

Question. Do AMP_f and HO_f coincide?

¹ Eli Gafni and Giuliano Losa. Time Is Not a Healer, but It Sure Makes Hindsight 20:20, Stabilization, Safety, and Security of Distributed Systems, 2023

² Attiya, Castañeda, Ghosh, Nowak. Equivalence and Separation between Heard-Of and Asynchronous Message-Passing Models, *arXiv*, 2026

Proof strategy for equivalence

Obtained via simulations:

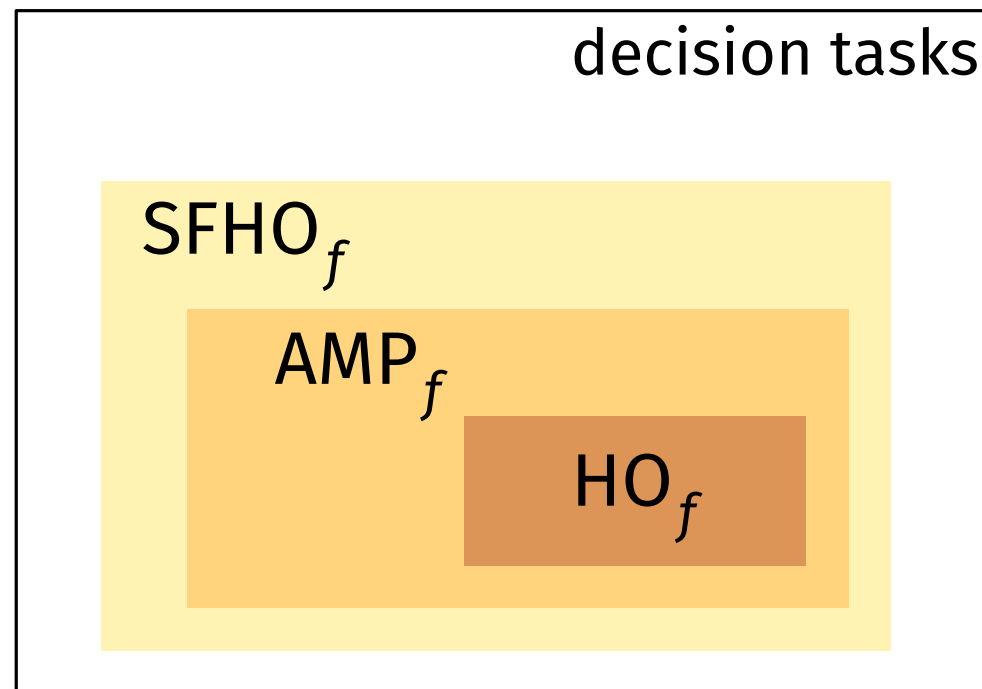
Theorem

Any task solvable in HO_f is solvable in AMP_f for $f < n$.

Theorem

Any task solvable in AMP_f is solvable in $SFHO_f$ for $f < n/2$.

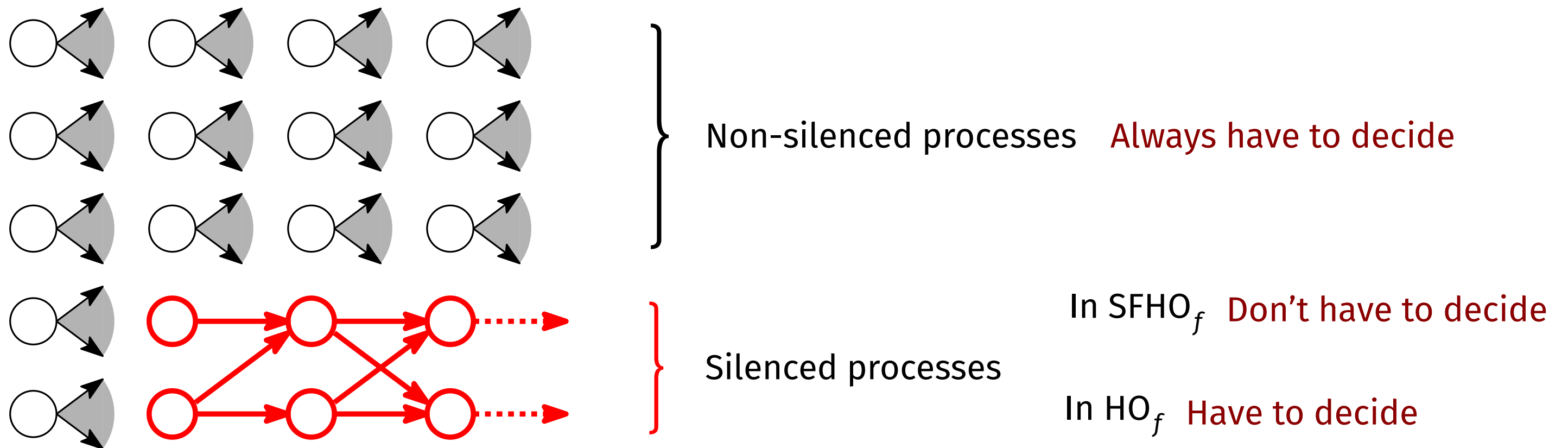
} Not covered in the talk



Goal. To show when $SFHO_f$ is contained in HO_f .

Silenced-Faulty HO_f model ($SFHO_f$)

- ☞ An intermediary model, nearly identical to HO_f
- ☞ Only difference: so-called *silenced processes* **do not have to decide** in $SFHO_f$.



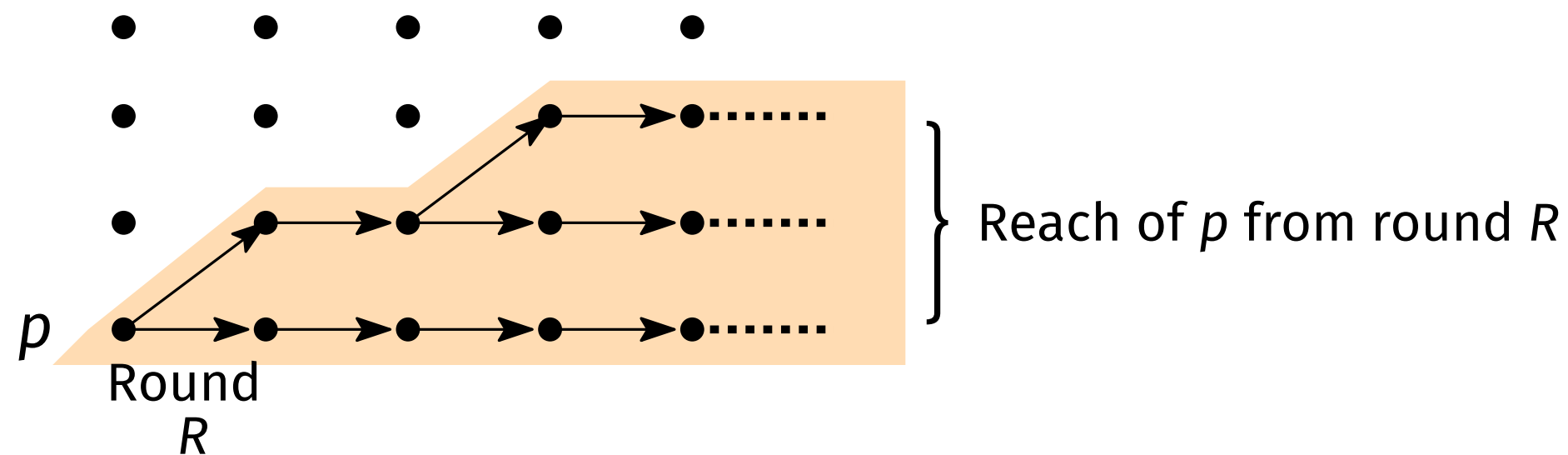
Silenced-Faulty HO_f model ($SFHO_f$)

Silenced processes

Definition (Reach)

The **reach** of a process p from round r is the union of:

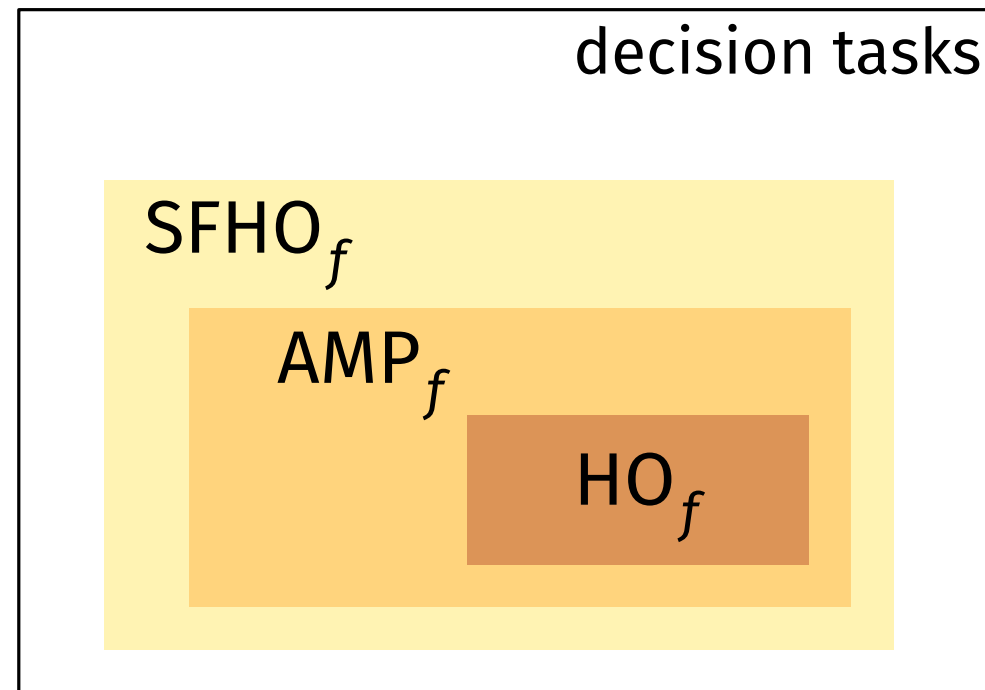
- ☞ the set of processes which hear from p in round r ,
- ☞ those who hear from one of the above in round $r + 1$,
- ☞ those who hear from one of the above in round $r + 2$,
- ☞ and so on till ∞



Definition (Silenced process)

A process p is **silenced** if from some round, its reach is at most f .

Colorless tasks: $SFHO_f$ in HO_f




Goal. To show when $SFHO_f$ is contained in HO_f .

	Colorless tasks	Colored tasks
$f \leq 1$ ($n \geq 3$)	Yes (Gafni, Losa 2023)	
$1 < f < n/2$		

Question. Do AMP_f and HO_f coincide?

Colorless tasks: $SFHO_f$ in HO_f

Suppose that a task T is solvable in $SFHO_f$ by protocol \mathcal{P} . WLOG assume that \mathcal{P} is full-information.



Every process always broadcasts its entire history till that point.

Colorless tasks: $SFHO_f$ in HO_f

Suppose that a task T is solvable in $SFHO_f$ by protocol \mathcal{P} . WLOG assume that \mathcal{P} is full-information.

Goal. Modify \mathcal{P} into a protocol that solves T in HO_f .



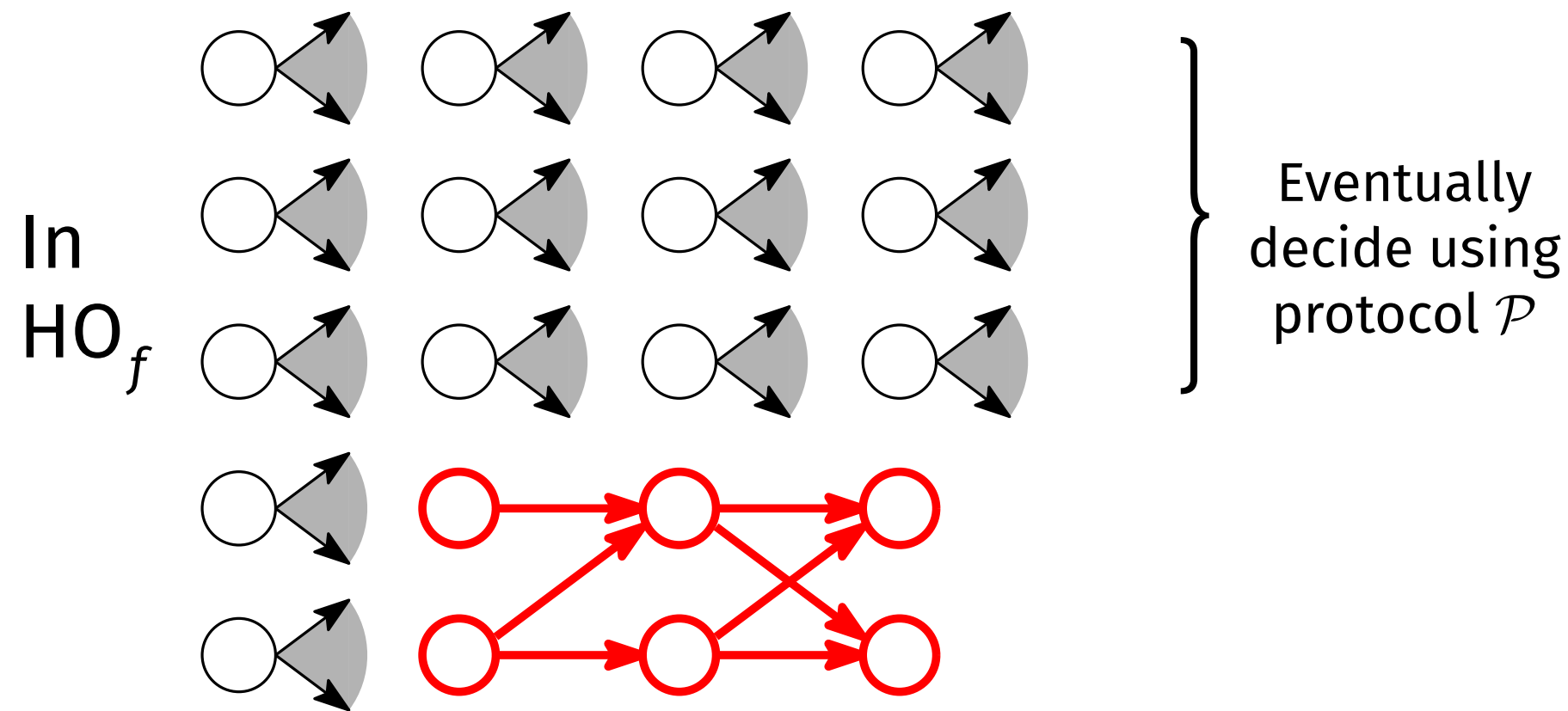
Every process always broadcasts its entire history till that point.

Colorless tasks: $SFHO_f$ in HO_f

Suppose that a task T is solvable in $SFHO_f$ by protocol \mathcal{P} . WLOG assume that \mathcal{P} is full-information.

Goal. Modify \mathcal{P} into a protocol that solves T in HO_f .

Every process always broadcasts its entire history till that point.



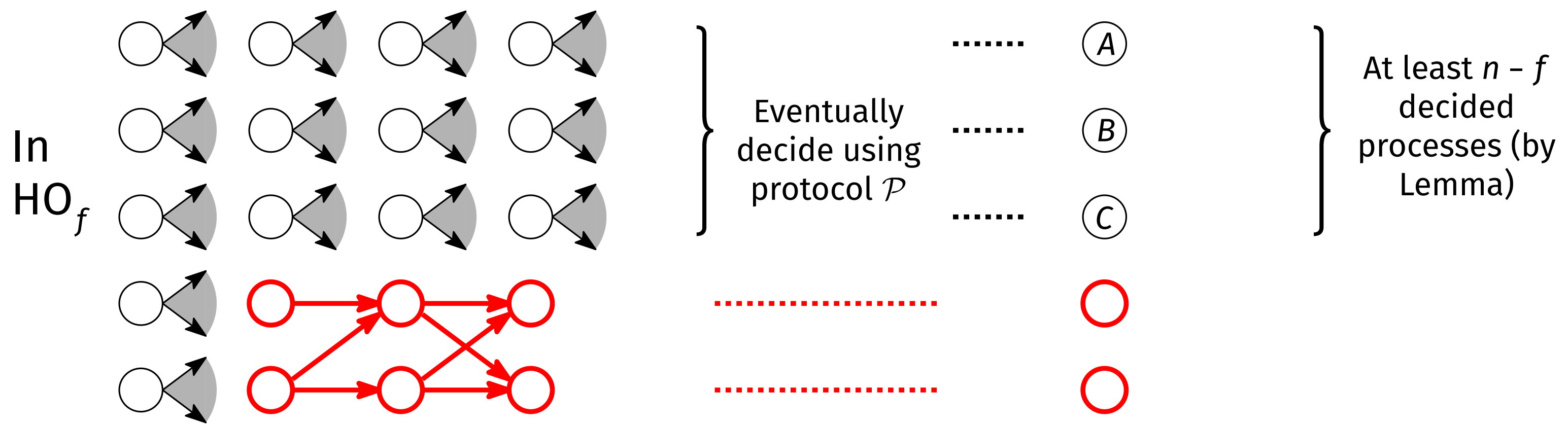
Colorless tasks: $SFHO_f$ in HO_f

Suppose that a task T is solvable in $SFHO_f$ by protocol \mathcal{P} . WLOG assume that \mathcal{P} is full-information.

Goal. Modify \mathcal{P} into a protocol that solves T in HO_f .

Every process always broadcasts its entire history till that point.

Lemma
 For $f < n/2$, there are at least $n - f$ non-silenced processes.



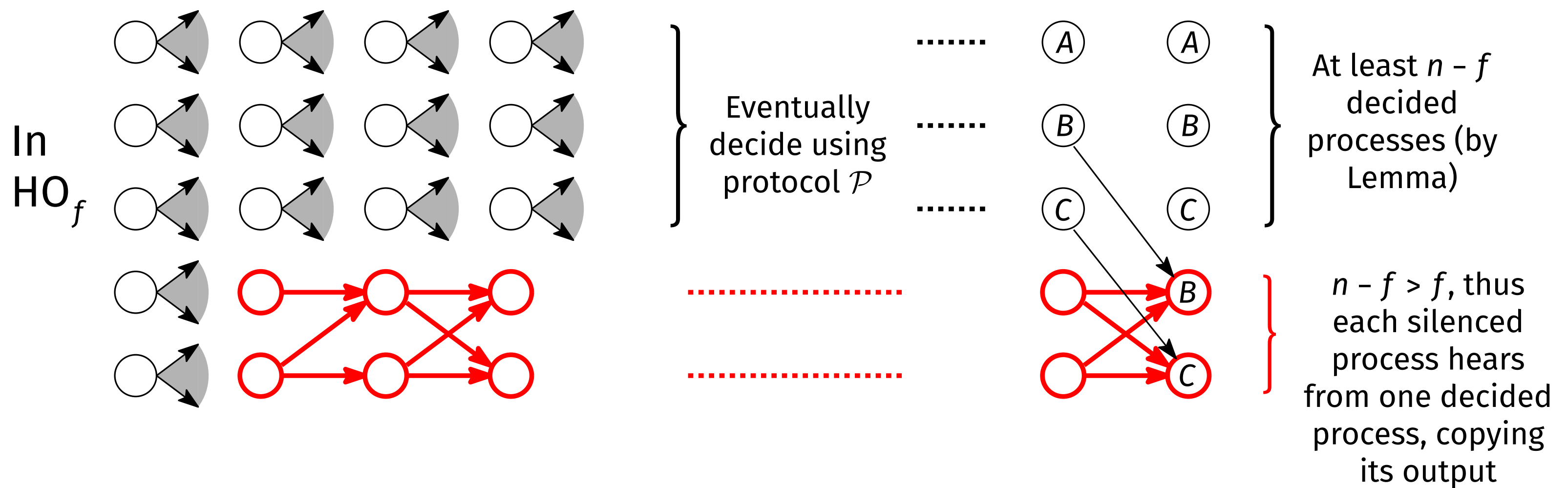
Colorless tasks: $SFHO_f$ in HO_f

Suppose that a task T is solvable in $SFHO_f$ by protocol \mathcal{P} . WLOG assume that \mathcal{P} is full-information.

Goal. Modify \mathcal{P} into a protocol that solves T in HO_f .

Every process always broadcasts its entire history till that point.

Lemma
 For $f < n/2$, there are at least $n - f$ non-silenced processes.



Colored tasks: $SFHO_1$ in HO_1

	Colorless tasks	Colored tasks
$f \leq 1 (n \geq 3)$	Yes (Gafni, Losa 2023)	
$1 < f < n/2$	Yes	

Question. Do AMP_f and HO_f coincide?

Colored tasks: $SFHO_1$ in HO_1

Suppose that a task T is solvable in $SFHO_1$ by protocol \mathcal{P} . WLOG assume \mathcal{P} is full-information.

Goal. Modify \mathcal{P} into a protocol that solves T in HO_1 .

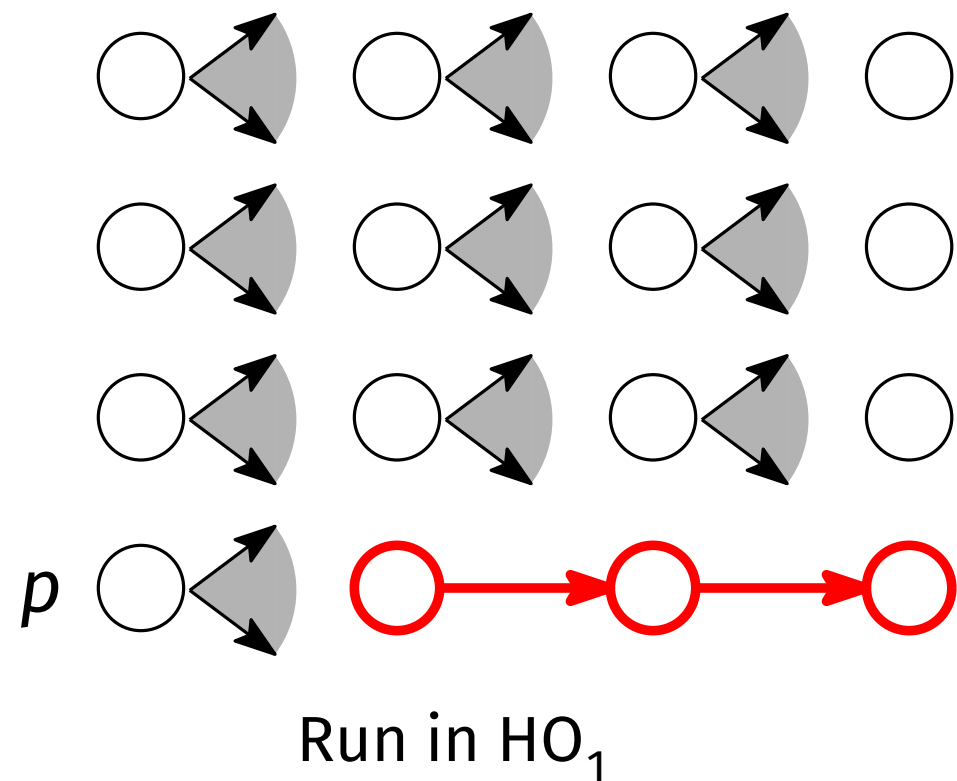
Colored tasks: SFHO₁ in HO₁

Suppose that a task T is solvable in SFHO₁ by protocol \mathcal{P} . WLOG assume \mathcal{P} is full-information.

Goal. Modify \mathcal{P} into a protocol that solves T in HO₁.

Lemma

For $n \geq 3$, there can be at most *one* silenced process in HO₁.



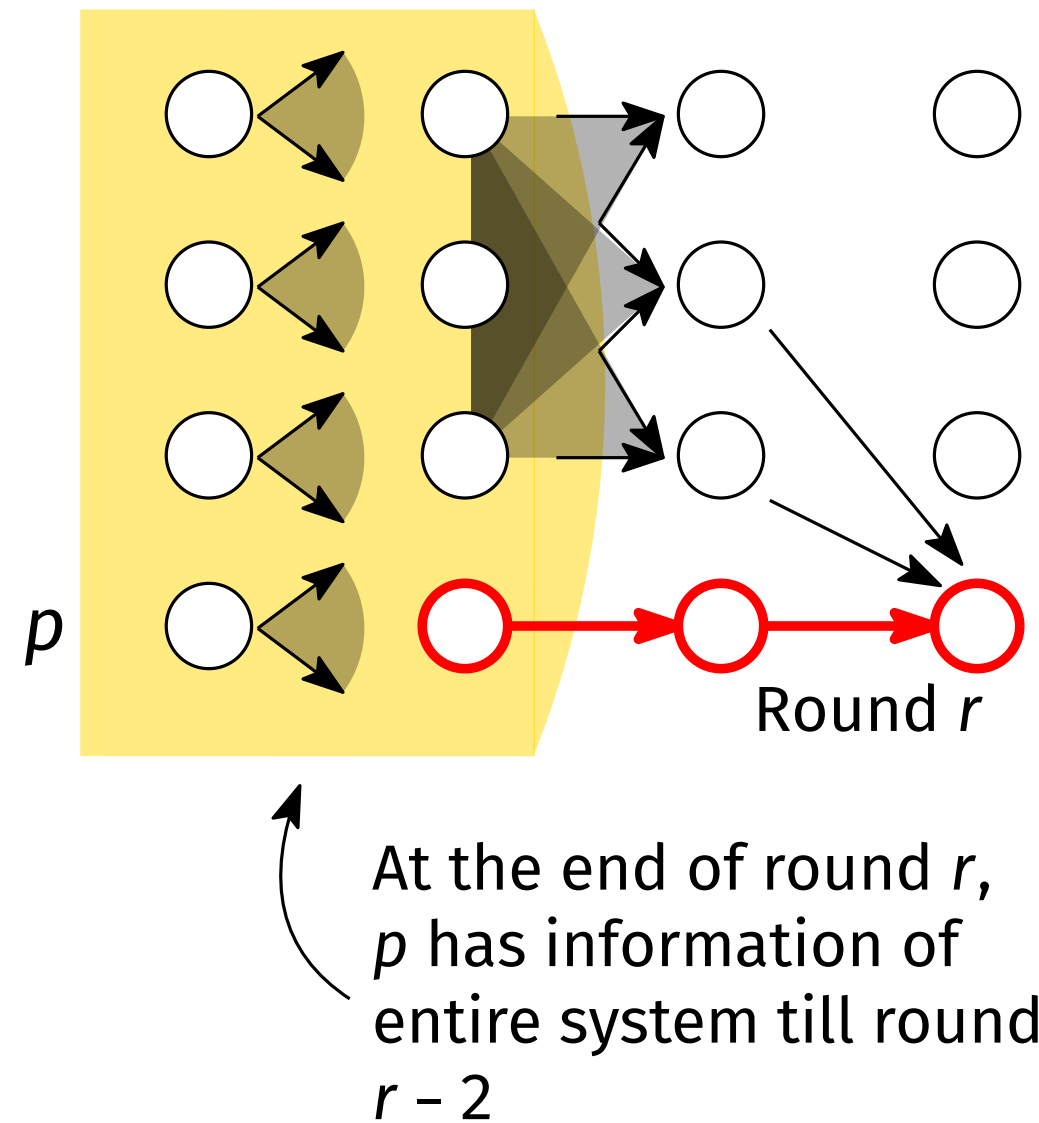
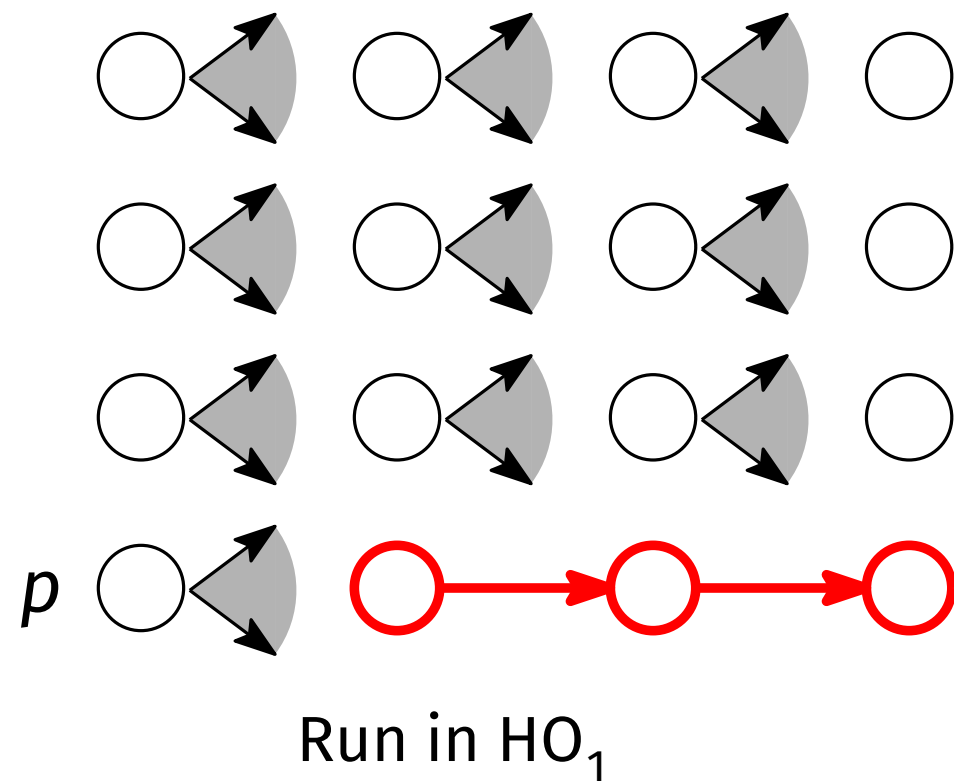
Colored tasks: SFHO₁ in HO₁

Suppose that a task T is solvable in SFHO₁ by protocol \mathcal{P} . WLOG assume \mathcal{P} is full-information.

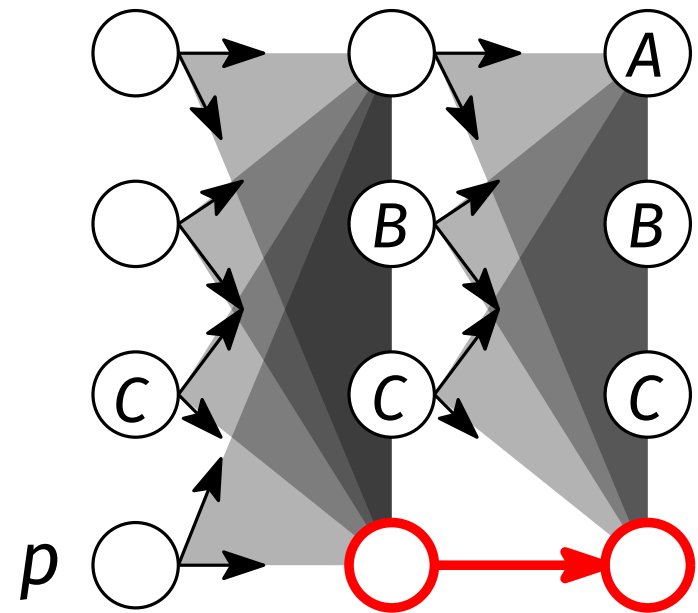
Goal. Modify \mathcal{P} into a protocol that solves T in HO₁.

Lemma

For $n \geq 3$, there can be at most *one* silenced process in HO₁.

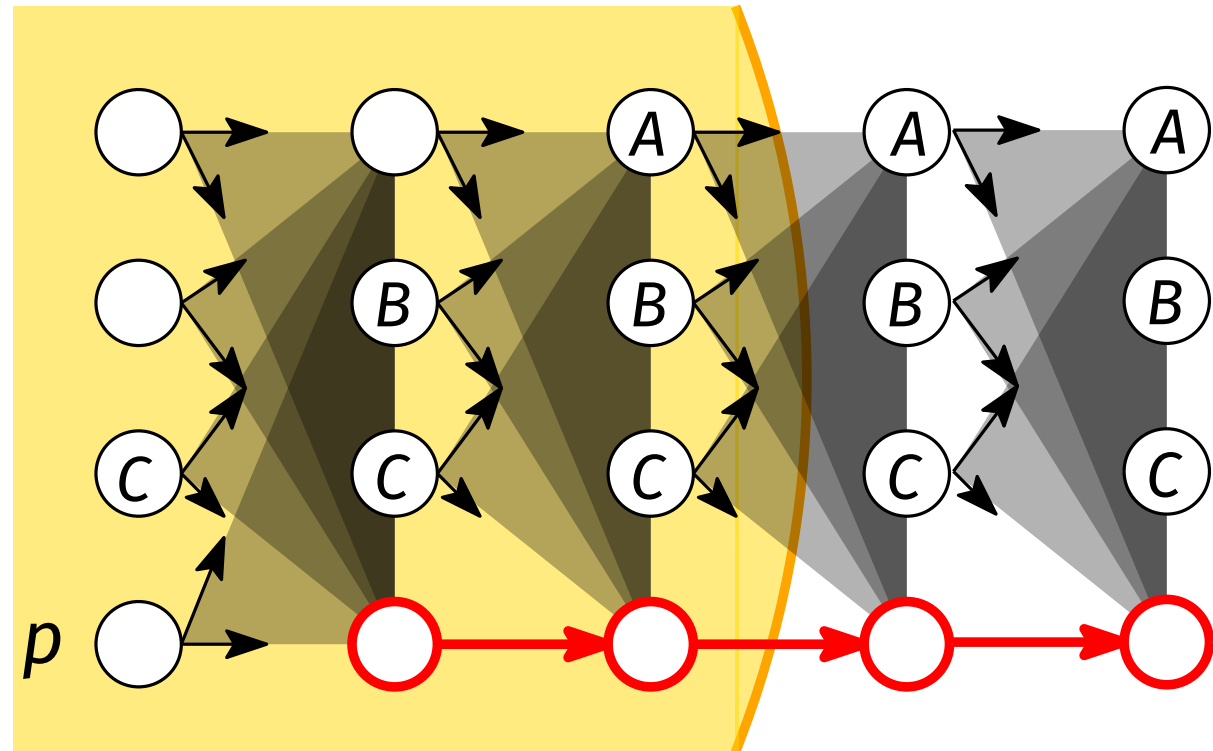


Colored tasks: SFO_1 in HO_1



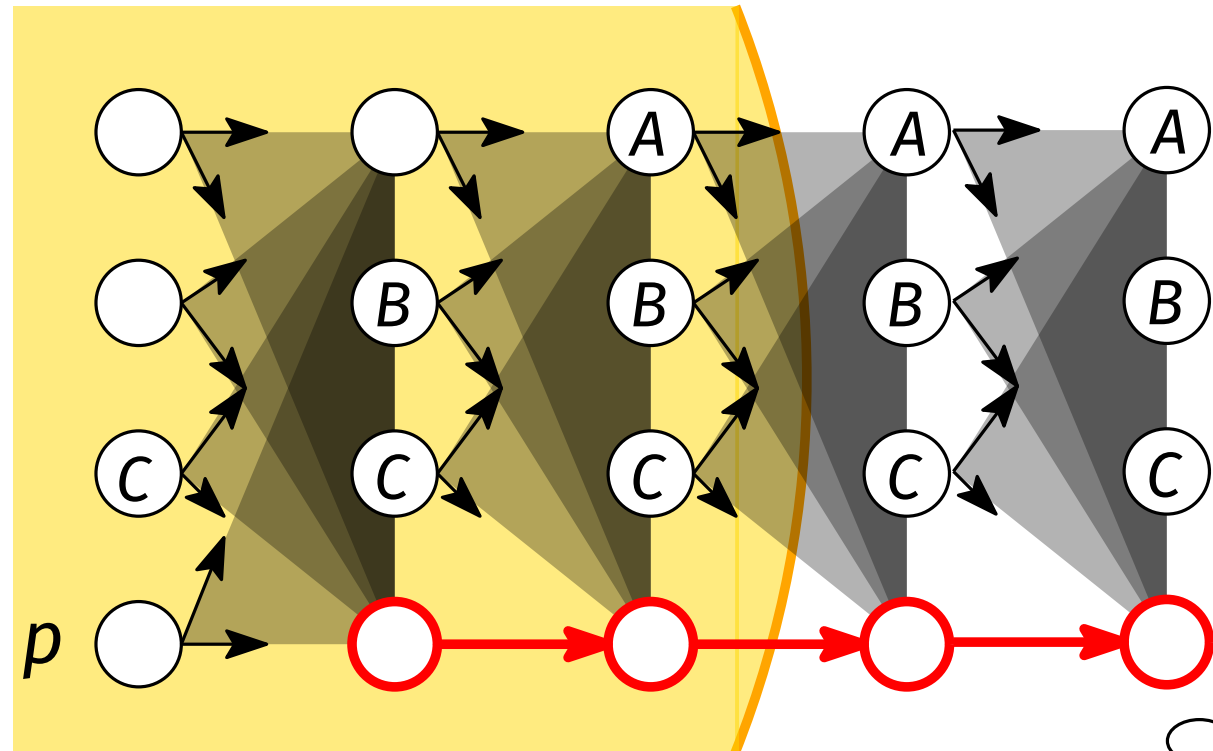
Suppose p is silenced
and all other processes
have decided.

Colored tasks: SFO_1 in HO_1



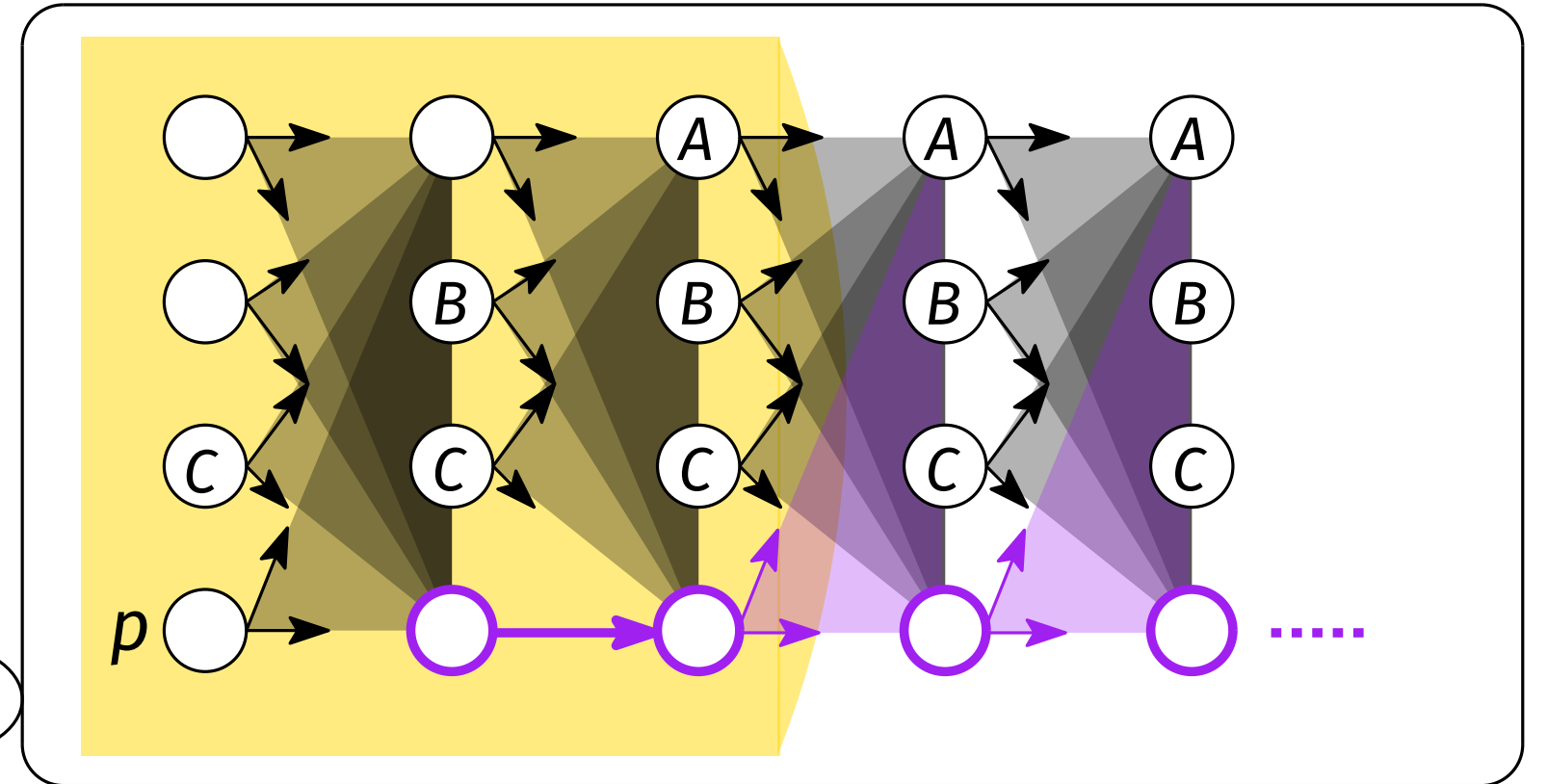
Suppose p is silenced
and all other processes
have decided.

Colored tasks: $SFHO_1$ in HO_1

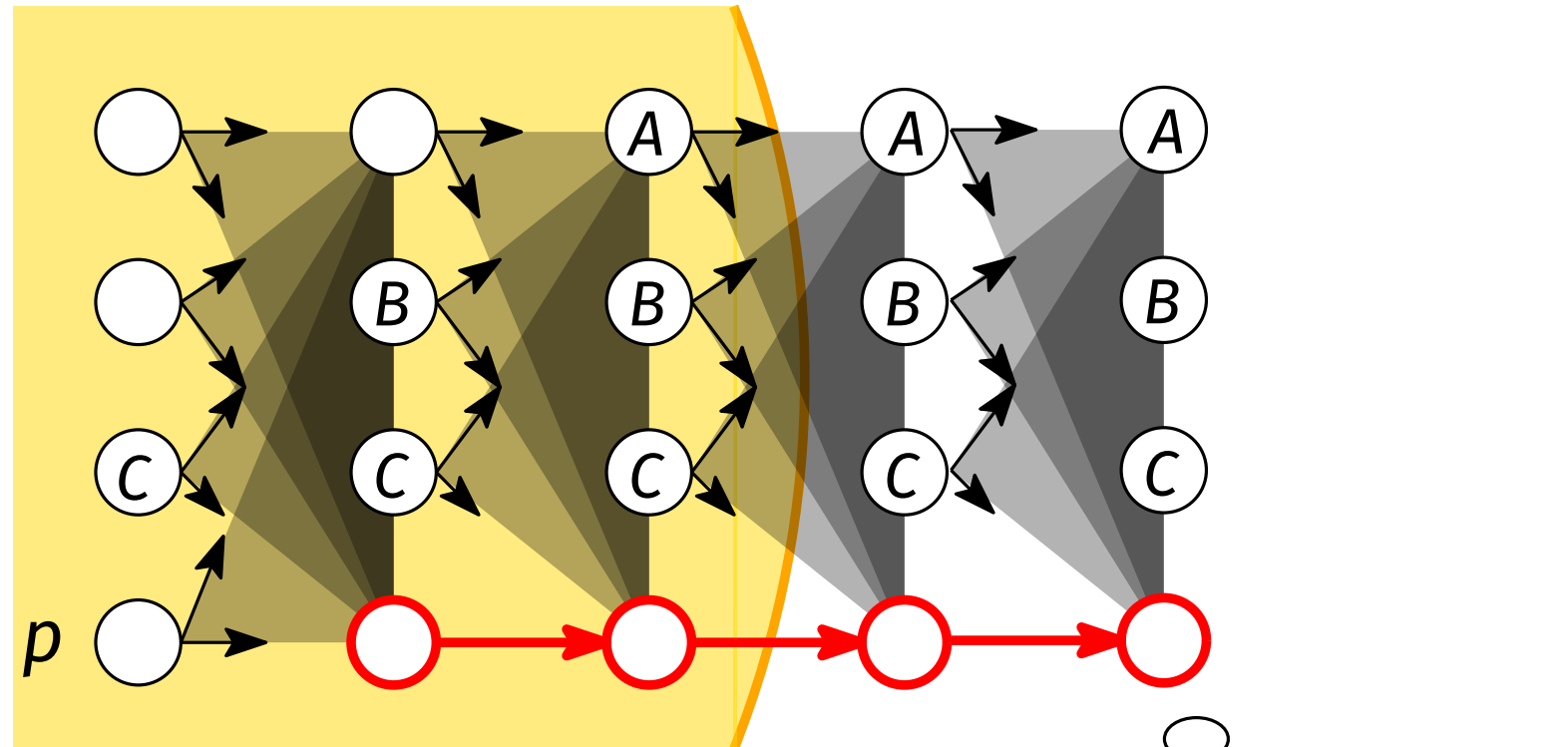


Suppose p is silenced and all other processes have decided.

p considers a run where it is not silenced...

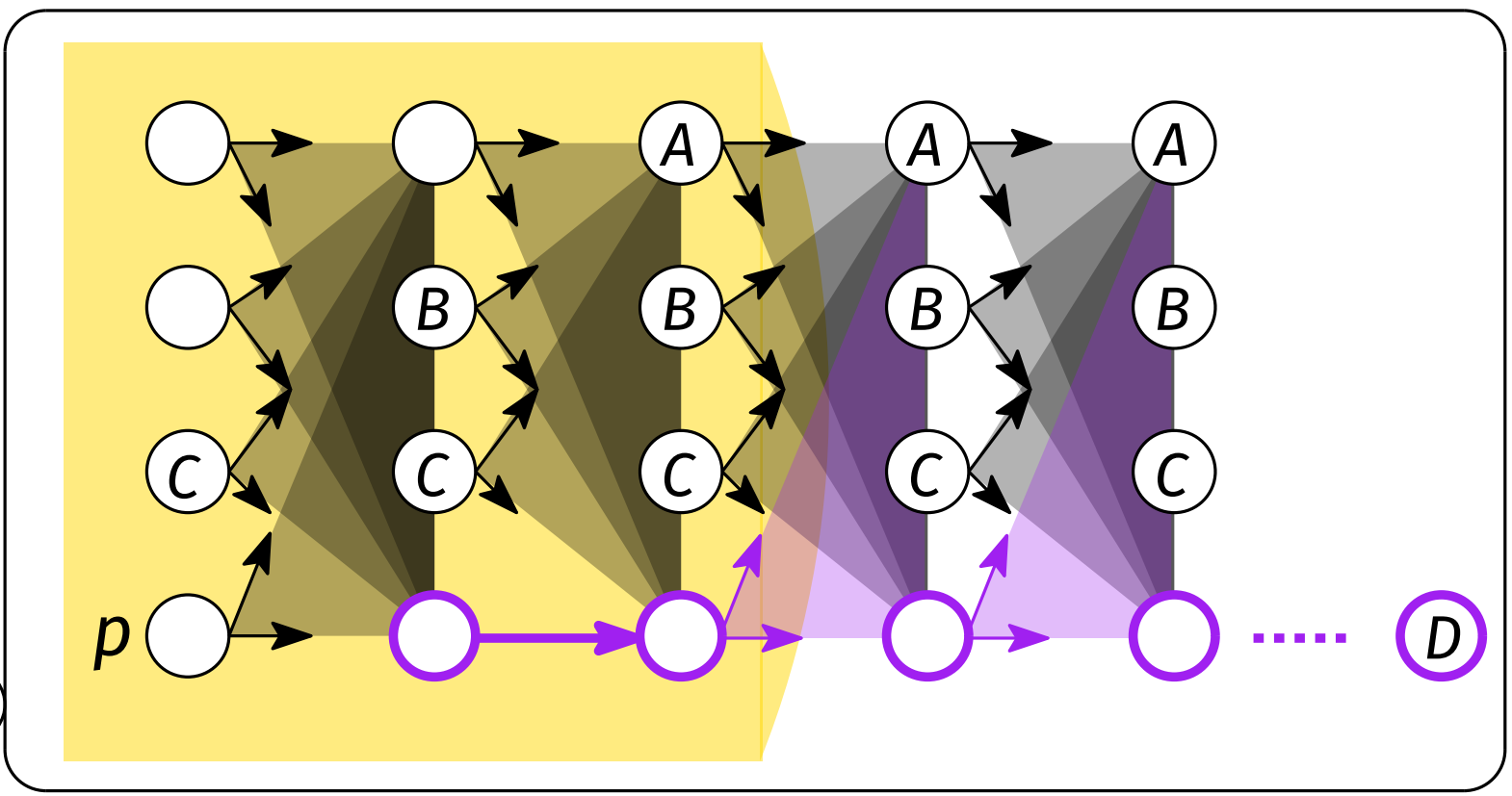


Colored tasks: $SFHO_1$ in HO_1



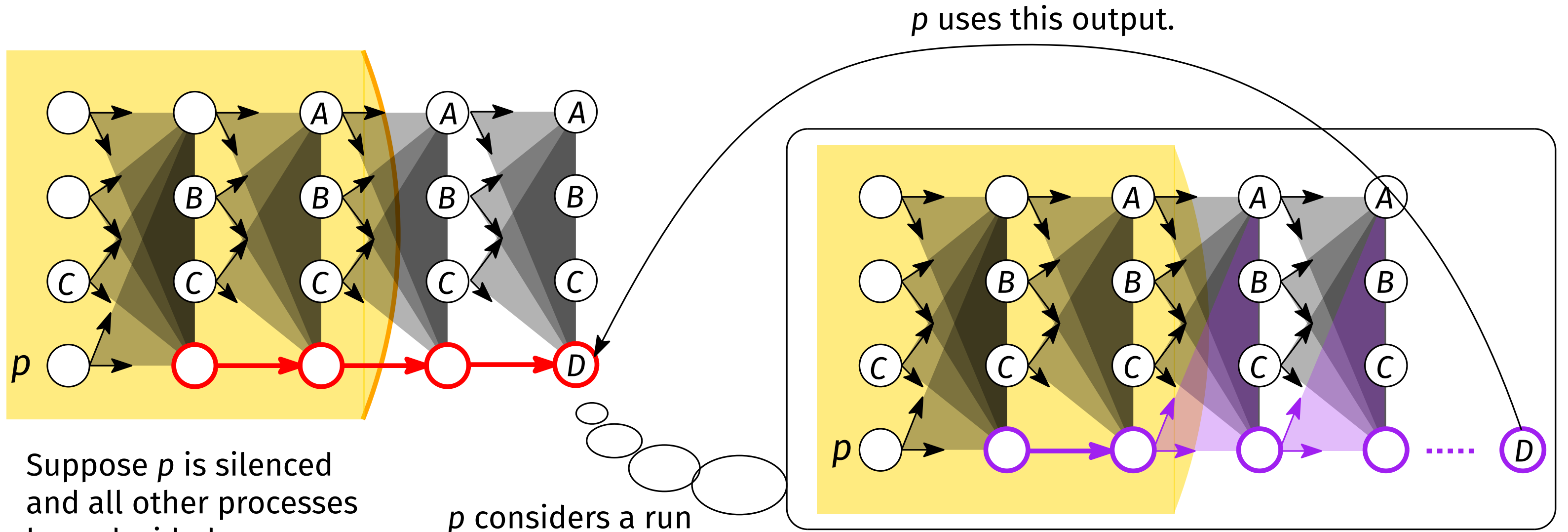
Suppose p is silenced and all other processes have decided.

p considers a run where it is not silenced...



This is a run in $SFHO_f$. Also p is not silenced, protocol \mathcal{P} gives p output D .

Colored tasks: SFHO₁ in HO₁



Suppose p is silenced and all other processes have decided.

p considers a run where it is not silenced...

This is a run in SFHO_f. Also p is not silenced, protocol \mathcal{P} gives p output D .

Colored tasks and $f > 1$?

	Colorless tasks	Colored tasks
$f \leq 1$ ($n \geq 3$)	Yes (Gafni, Losa 2023)	Yes
$1 < f < n/2$	Yes	

Question. Do AMP_f and HO_f coincide?

Colored tasks and $f > 1$: $\mathbf{HO}_f \neq \mathbf{AMP}_f$

Definition (Renaming task)

The renaming task¹ on an *initial name space* of size M and *new name space* of size N where $M > N \geq n$ is given by the triple $T = (\mathcal{I}, \mathcal{O}, \Delta)$ where

- any vector $I \in \mathcal{I}$ consists of distinct values from initial name space
- any vector $O \in \mathcal{O}$ has distinct non- \perp values from new name space
- $\Delta = \mathcal{I} \times \mathcal{O}$

Example. Initial name space = \mathbb{N} New name space = $[1 \dots N]$
 $I = (2^1, 2^2, \dots, 2^{n-1}, 2^n)$ $O = (1, 2, \dots, n-2, \perp, \perp)$

Theorem (Attiya, Bar-Noy, Dolev, Peleg, Reischuk, 1990¹)

For $f < n/2$, the renaming task with initial name space of unbounded size and new name space of size $n + f$ is solvable in \mathbf{AMP}_f .

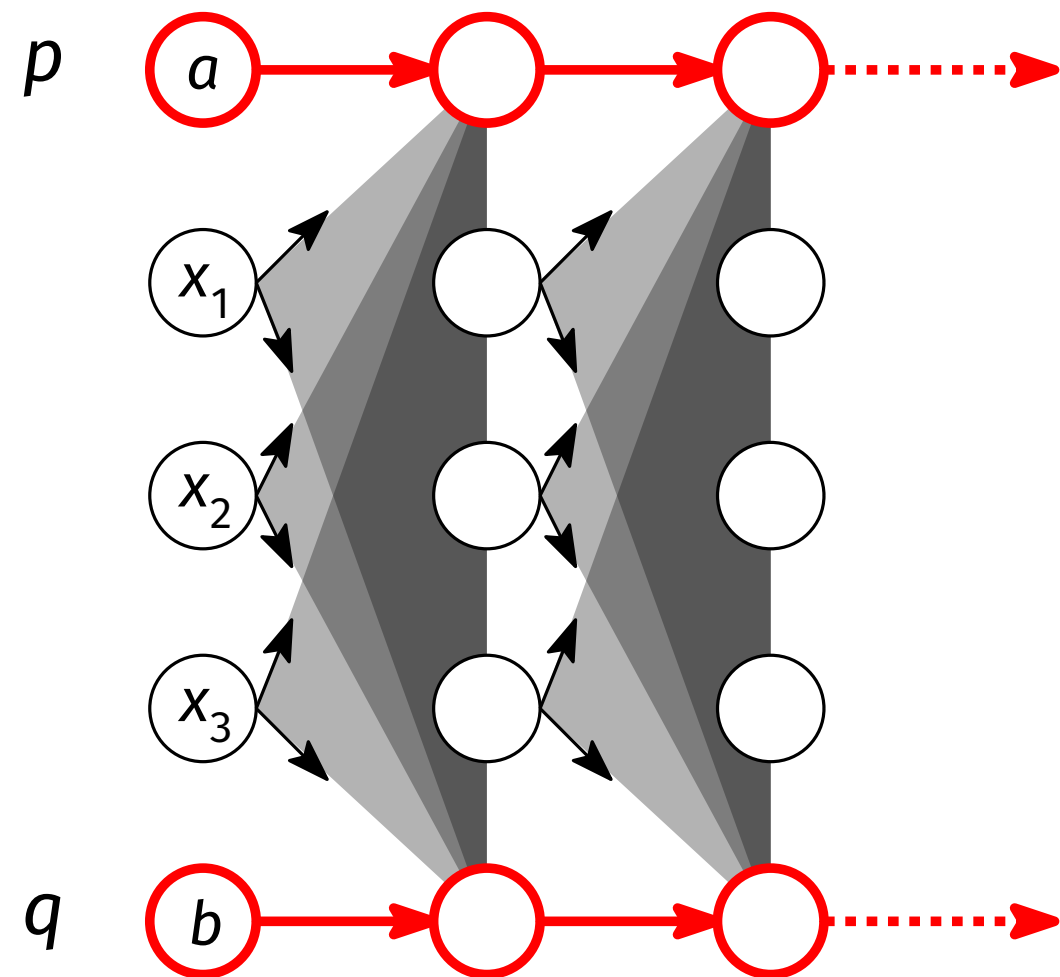
¹ Attiya, Bar-Noy, Dolev, Peleg, Reischuk. Renaming in an asynchronous environment. J. ACM, 37(3):524–548, July 1990.

Colored tasks and $f > 1$: $\text{HO}_f \neq \text{AMP}_f$

Impossibility of renaming in HO_f

Assume that there is a protocol \mathcal{P} (WLOG full-information) solving renaming in HO_f .

Consider the following run of \mathcal{P} where p, q are silenced from round 1:

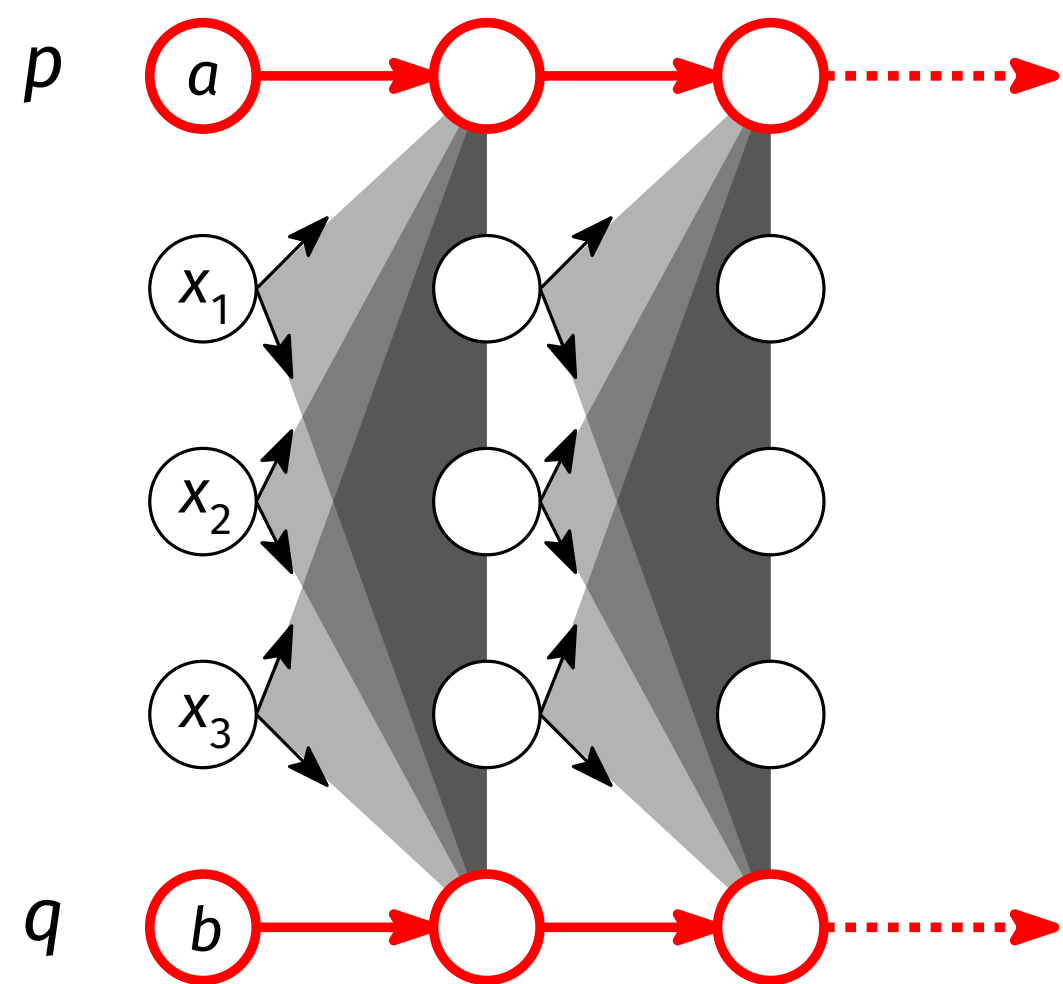


Colored tasks and $f > 1$: $\text{HO}_f \neq \text{AMP}_f$

Impossibility of renaming in HO_f

Assume that there is a protocol \mathcal{P} (WLOG full-information) solving renaming in HO_f .

Consider the following run of \mathcal{P} where p, q are silenced from round 1:



History of p =
 $send(a), recv(a, x_1, x_2, x_3), send(send(a), recv(a, x_1, x_2, x_3)), \dots$

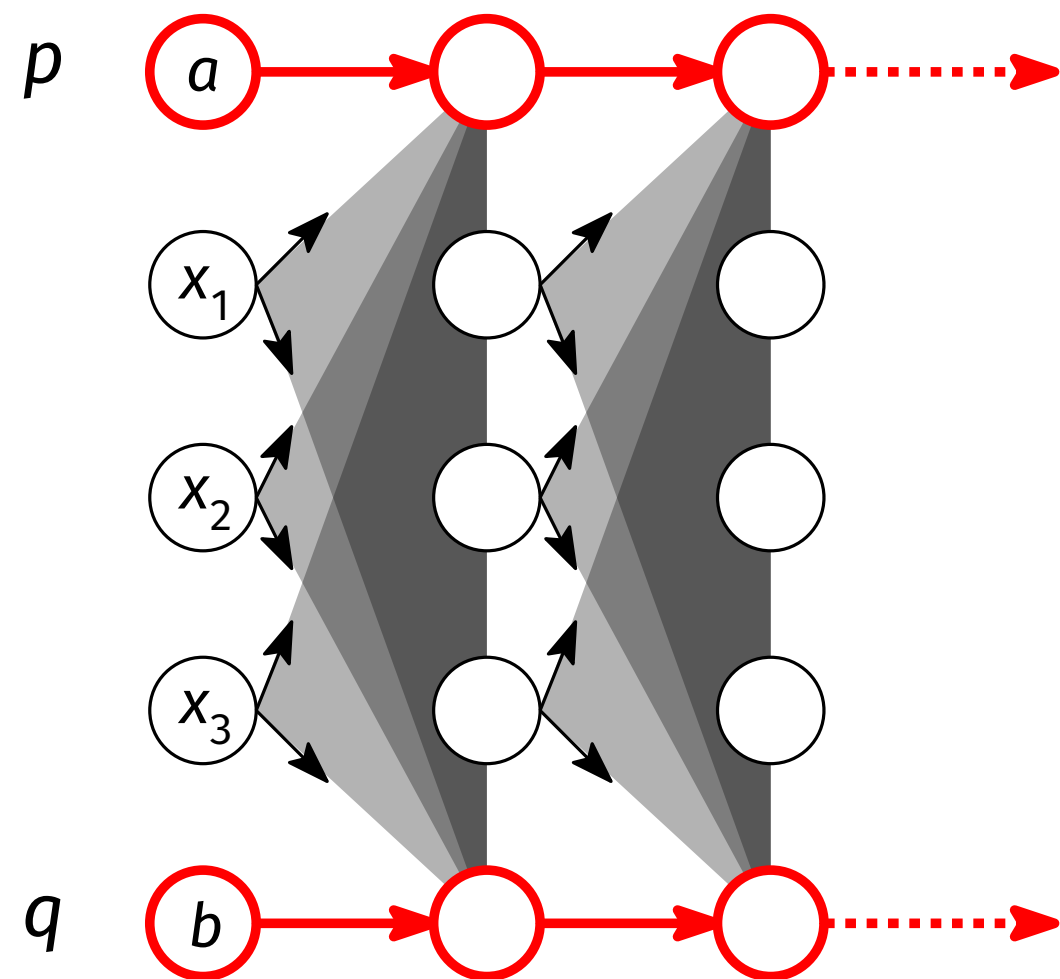
History of q =
 $send(b), recv(b, x_1, x_2, x_3), send(send(b), recv(b, x_1, x_2, x_3)), \dots$

Colored tasks and $f > 1$: $\text{HO}_f \neq \text{AMP}_f$

Impossibility of renaming in HO_f

Assume that there is a protocol \mathcal{P} (WLOG full-information) solving renaming in HO_f .

Consider the following run of \mathcal{P} where p, q are silenced from round 1:



History of p =
 $send(a), recv(a, x_1, x_2, x_3), send(send(a), recv(a, x_1, x_2, x_3)), \dots$

Output of p = $\delta(\text{History of } p)$ where δ is a decision function
 $= f(a)$ assuming x_1, x_2, x_3 fixed.

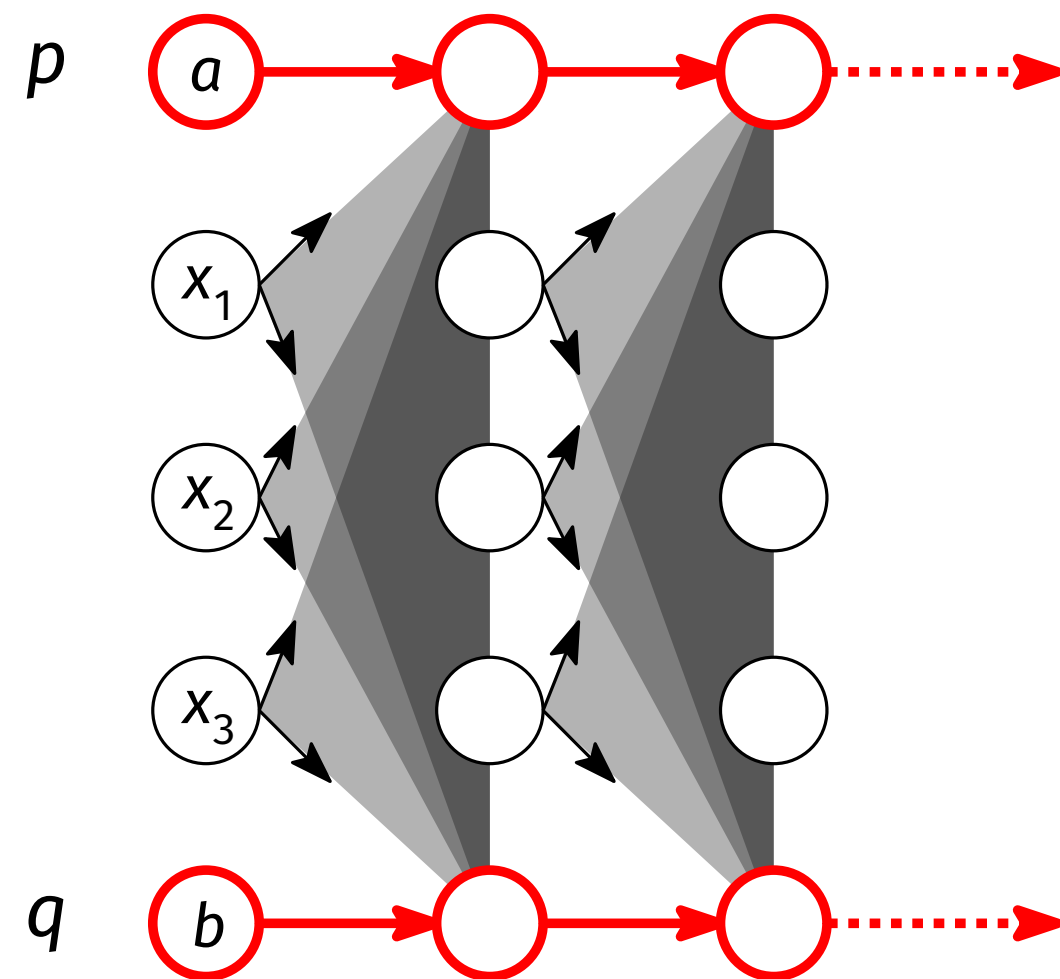
History of q =
 $send(b), recv(b, x_1, x_2, x_3), send(send(b), recv(b, x_1, x_2, x_3)), \dots$

Colored tasks and $f > 1$: $\text{HO}_f \neq \text{AMP}_f$

Impossibility of renaming in HO_f

Assume that there is a protocol \mathcal{P} (WLOG full-information) solving renaming in HO_f .

Consider the following run of \mathcal{P} where p, q are silenced from round 1:



History of p =
 $send(a), recv(a, x_1, x_2, x_3), send(send(a), recv(a, x_1, x_2, x_3)), \dots$

Output of p = $\delta(\text{History of } p)$ where δ is a decision function
 $= f(a)$ assuming x_1, x_2, x_3 fixed.

History of q =
 $send(b), recv(b, x_1, x_2, x_3), send(send(b), recv(b, x_1, x_2, x_3)), \dots$

Output of q = $\delta(\text{History of } q)$
 $= f(b)$.

As there are only finitely many new names, we can choose a, b such that $f(a) = f(b)$, thus forcing the two silenced processes to choose the same new names. Contradiction!

Conclusion and Open Questions

	Colorless tasks	Colored tasks
$f \leq 1 (n \geq 3)$	Yes (Gafni, Losa 2023)	Yes
$1 < f < n/2$	Yes	No

Question. Do AMP_f and HO_f coincide?

Further work

- ☞ A comparison of AMP_f and HO_f for $f \geq n/2$.
- ☞ Is there class of tasks larger than that of colorless tasks on which $\text{AMP}_f = \text{HO}_f$?

References

- [1] Bernadette Charron-Bost and André Schiper. The Heard-Of model: computing in distributed systems with benign faults. *Distributed Computing*, 22(1):49–71, April 2009.
- [2] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, apr 1985.
- [3] Eli Gafni and Giuliano Losa. Time Is Not a Healer, but It Sure Makes Hindsight 20:20, *Stabilization, Safety, and Security of Distributed Systems*, 2023
- [4] Attiya, Bar-Noy, Dolev, Peleg, Reischuk. Renaming in an asynchronous environment. *J. ACM*, 37(3):524–548, July 1990.
- [5] Hagit Attiya, Armando Castañeda, Dhrubajyoti Ghosh, Thomas Nowak. Equivalence and Separation between Heard-Of and Asynchronous Message-Passing Models. *arXiv*, 2026

Problem Studied

The overall goal is to compare the power of AMP_f and HO_f . But how are they related in the first place?

Approach to solving a task in AMP: *construct a virtual round-based system.*

Problem Studied

The overall goal is to compare the power of AMP_f and HO_f . But how are they related in the first place?

Approach to solving a task in AMP: *construct a virtual round-based system.*

Consider $n = 3$ and $f = 1$.

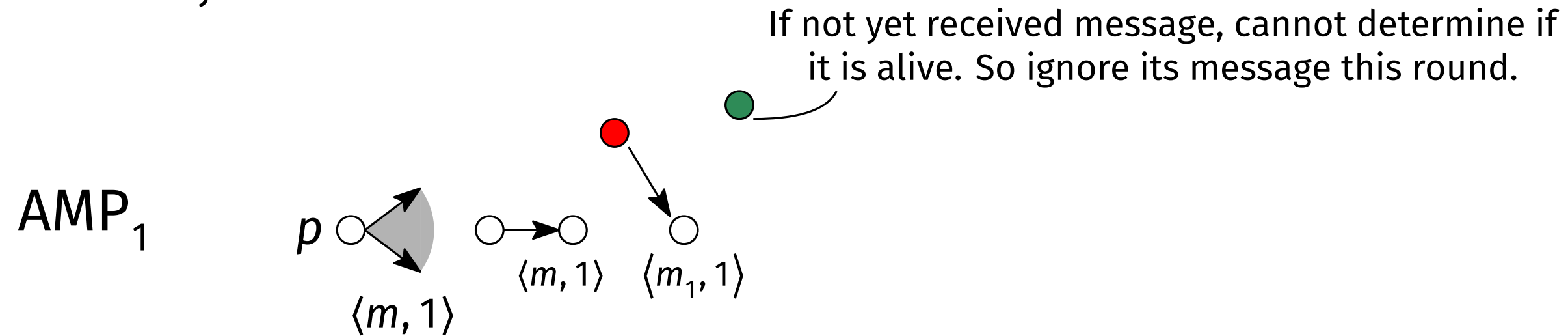


Problem Studied

The overall goal is to compare the power of AMP_f and HO_f . But how are they related in the first place?

Approach to solving a task in AMP: *construct a virtual round-based system.*

Consider $n = 3$ and $f = 1$.

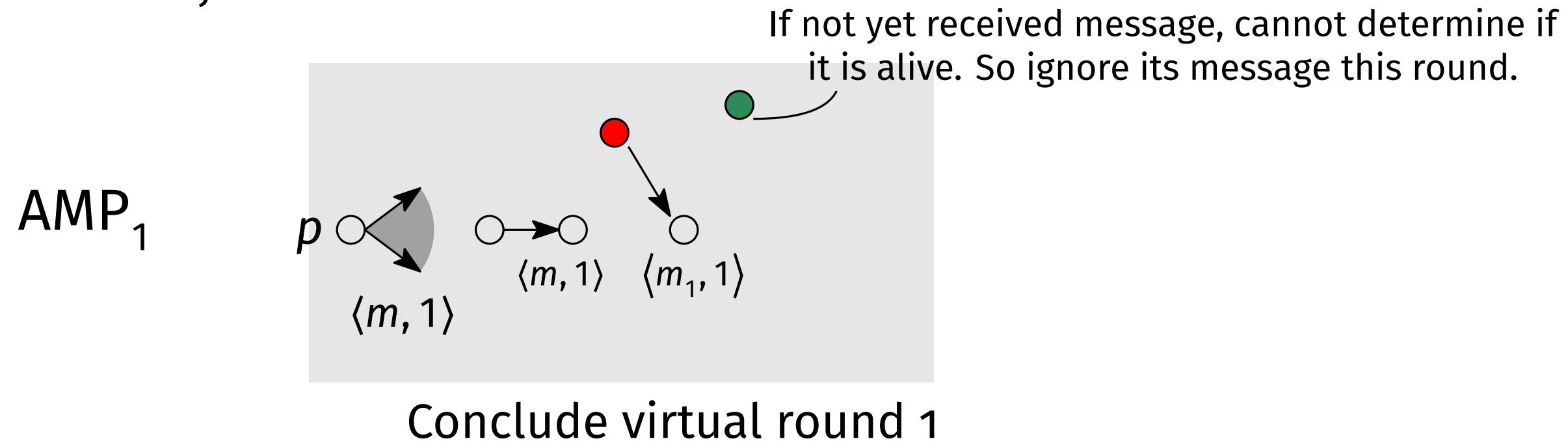


Problem Studied

The overall goal is to compare the power of AMP_f and HO_f . But how are they related in the first place?

Approach to solving a task in AMP: *construct a virtual round-based system.*

Consider $n = 3$ and $f = 1$.

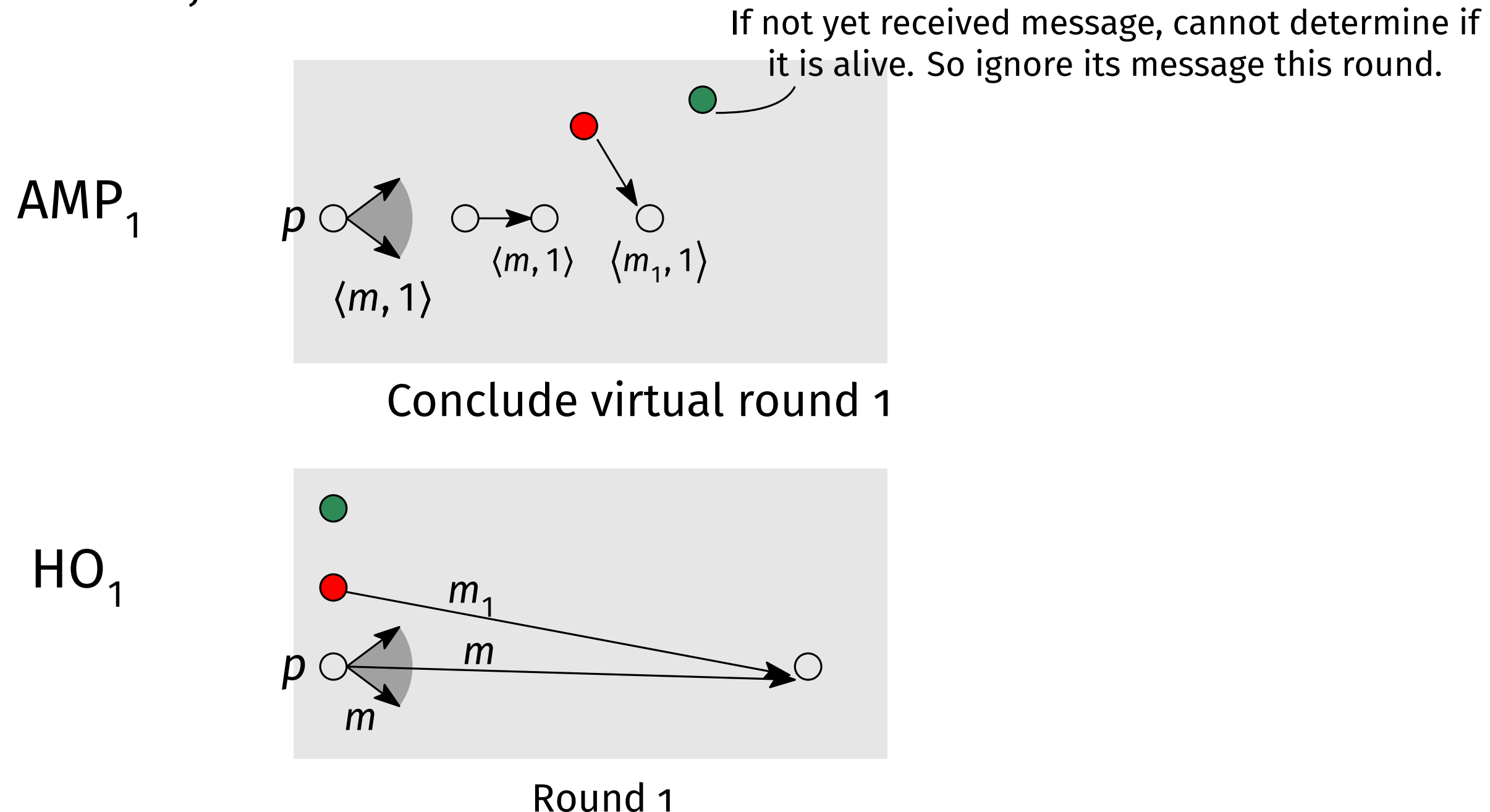


Problem Studied

The overall goal is to compare the power of AMP_f and HO_f . But how are they related in the first place?

Approach to solving a task in AMP: *construct a virtual round-based system.*

Consider $n = 3$ and $f = 1$.



Problem Studied

Motivation

Do AMP_f and HO_f solve the same set of decision tasks?

Why HO_f ?

Problem Studied

Motivation

Do AMP_f and HO_f solve the same set of decision tasks?

Why HO_f ?

- 👉 Round-based models are easier for designing and analysing algorithms as well as for lower bounds and impossibility proofs.

Problem Studied

Motivation

Do AMP_f and HO_f solve the same set of decision tasks?

Why HO_f ?

- ▶ Round-based models are easier for designing and analysing algorithms as well as for lower bounds and impossibility proofs.
- ▶ Models like HO_f allow the set of executions to be *compact* in the *longest-common-prefix* topology¹.

¹ Bowen Alpern and Fred B. Schneider. Defining liveness. Information Processing Letters, 21(4):181–185, 1985

Problem Studied

Motivation

Do AMP_f and HO_f solve the same set of decision tasks?

Why HO_f ?

- ▶ Round-based models are easier for designing and analysing algorithms as well as for lower bounds and impossibility proofs.
- ▶ Models like HO_f allow the set of executions to be *compact* in the *longest-common-prefix* topology¹.
- ▶ If a protocol solves a task T in such a model, it must do so in a bounded number of rounds^{2,3}.

¹ Bowen Alpern and Fred B. Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, 1985

² Thomas Nowak. *Topology in distributed computing*. Master's thesis, TU Wien, March 2010.

³ Ronit Rubinfeld and Shlomo Moran. Closed schedulers: a novel technique for analyzing asynchronous protocols. *Distributed Computing*, 8:203–210, 1995.