

Problèmes vérifiables par agents mobiles

Evangelos Bampas and David Ilcinkas

LaBRI, CNRS & Université de Bordeaux, France

Decision problems

In centralized computing

- Decision: the Turing Machine must answer **yes/no**
 - Example: Is the graph 3-colorable?

Decision problems

In centralized computing

- Decision: the Turing Machine must answer **yes/no**
 - Example: Is the graph 3-colorable?

In classical distributed computing

- Decision: local computations and decisions by the nodes
 - For **yes** instances, all nodes must answer **yes**
 - For **no** instances, at least one node must answer **no**
- Example: Is the graph cycle-free?

Decision problems

In centralized computing

- Decision: the Turing Machine must answer **yes/no**
 - Example: Is the graph 3-colorable?

In classical distributed computing

- Decision: local computations and decisions by the nodes
 - For **yes** instances, all nodes must answer **yes**
 - For **no** instances, at least one node must answer **no**
- Example: Is the graph cycle-free?

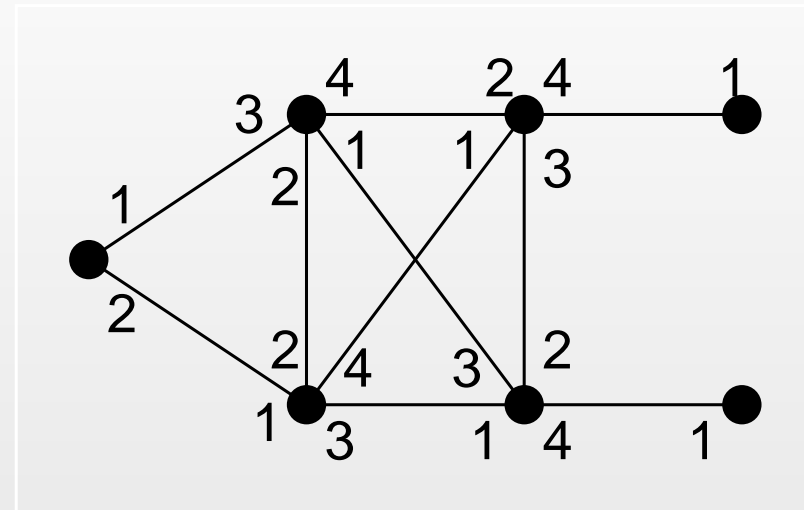
In distributed computing by mobile agents

- The topic of this talk.

Mobile agent computational model

Network

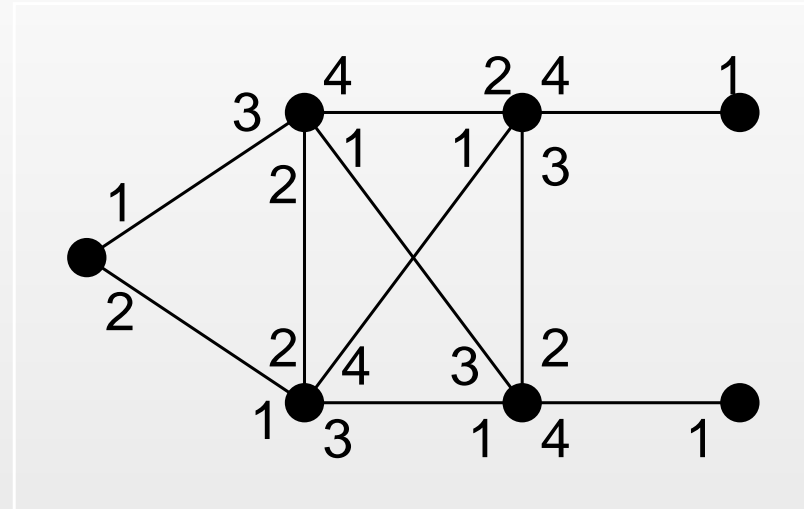
- Connected.
- Anonymous.
- Local port numbering.



Mobile agent computational model

Network

- Connected.
- Anonymous.
- Local port numbering.



Agents

- Copies of a Turing Machine moving in the network.
- Agent i initially receives **unique identifier id_i** and **input string x_i** .
- Execute synchronous steps as follows:
 - Perform finite local computation based on **degree**, **incoming port number**, **configurations of collocated agents**.
 - Halt (**yes/no**), stay idle, or exit through one of the ports.

Mobile agent decision problems

Definition

A decision problem is a set of instances $(G, \mathbf{s}, \mathbf{x})$.

- G : graph.
- \mathbf{s} : list of nodes (starting positions).
- \mathbf{x} : list of strings (inputs).

Examples

path = $\{(G, \mathbf{s}, \mathbf{x}) : G \text{ is a path}\}$

teamsize = $\{(G, \mathbf{s}, \mathbf{x}) : \forall i x_i = |\mathbf{s}| = |\mathbf{x}|\}$

graphsize = $\{(G, \mathbf{s}, \mathbf{x}) : \forall i x_i = |V(G)|\}$

Decidability...

[Fraigniaud and Pelc, LATIN 2012]

Mobile Agent Decidable problems (class MAD)

A problem Π is **decidable** if $\exists M \forall (G, \mathbf{s}, \mathbf{x})$:

if $(G, \mathbf{s}, \mathbf{x}) \in \Pi$, then $\forall id M(id, G, \mathbf{s}, \mathbf{x}) = \text{all-yes}$

if $(G, \mathbf{s}, \mathbf{x}) \notin \Pi$, then $\forall id M(id, G, \mathbf{s}, \mathbf{x}) = \text{some-no}$

Verification

In centralized computing

Verification: thanks to a **certificate**, the Turing Machine must answer **yes/no**

- In **yes** instances, **there exists** a certificate such that the machine answers **yes**
- In **no** instances, **for all** certificates, the machine answers **no**

Verification

In centralized computing

Verification: thanks to a **certificate**, the Turing Machine must answer **yes/no**

- In **yes** instances, **there exists** a certificate such that the machine answers **yes**
- In **no** instances, **for all** certificates, the machine answers **no**

In distributed computing

Verification: local computations with **local certificates**

- In **yes** instances, **there exists** a certificate such that **all** computing entities must answer **yes**
- In **no** instances, **for all** certificates, **at least one** node must answer **no**

... and verifiability

[Fraigniaud and Pelc, LATIN 2012]

Mobile Agent Decidable problems (class MAD)

A problem Π is **decidable** if $\exists M \forall (G, \mathbf{s}, \mathbf{x})$:

if $(G, \mathbf{s}, \mathbf{x}) \in \Pi$, then $\forall id M(id, G, \mathbf{s}, \mathbf{x}) = \text{all-yes}$

if $(G, \mathbf{s}, \mathbf{x}) \notin \Pi$, then $\forall id M(id, G, \mathbf{s}, \mathbf{x}) = \text{some-no}$

... and verifiability

[Fraigniaud and Pelc, LATIN 2012]

Mobile Agent Decidable problems (class MAD)

A problem Π is **decidable** if $\exists M \forall (G, s, \mathbf{x})$:

if $(G, s, \mathbf{x}) \in \Pi$, then $\forall id \ M(id, G, s, \mathbf{x}) = \text{all-yes}$

if $(G, s, \mathbf{x}) \notin \Pi$, then $\forall id \ M(id, G, s, \mathbf{x}) = \text{some-no}$

Mobile Agent Verifiable problems (class MAV)

A problem Π is **verifiable** if $\exists M \forall (G, s, \mathbf{x})$:

if $(G, s, \mathbf{x}) \in \Pi$, then $\exists y \ \forall id \ M(id, G, s, \langle \mathbf{x}, y \rangle) = \text{all-yes}$

if $(G, s, \mathbf{x}) \notin \Pi$, then $\forall y \ \forall id \ M(id, G, s, \langle \mathbf{x}, y \rangle) = \text{some-no}$

Examples

$\text{allempty} = \{(G, s, \mathbf{x}) : \forall i x_i = \epsilon\} \in \text{MAD}$

Each agent tests its input and accepts iff $x_i = \epsilon$.

Examples

allempty = $\{(G, s, \mathbf{x}) : \forall i x_i = \epsilon\} \in \text{MAD}$

Each agent tests its input and accepts iff $x_i = \epsilon$.

treesize = $\{(G, s, \mathbf{x}) : \forall i x_i = |V(G)| \text{ and } G \text{ is a tree}\} \in \text{MAD}$

Each agents checks independently if it is in a tree of size x_i (DFS traversal).

Examples

allempty = $\{(G, s, \mathbf{x}) : \forall i x_i = \epsilon\} \in \text{MAD}$

Each agent tests its input and accepts iff $x_i = \epsilon$.

treesize = $\{(G, s, \mathbf{x}) : \forall i x_i = |V(G)| \text{ and } G \text{ is a tree}\} \in \text{MAD}$

Each agent checks independently if it is in a tree of size x_i (DFS traversal).

degree = $\{(G, s, \mathbf{x}) : \forall i \exists v d_v = x_i\} \in \text{MAV}$

Certificate: path leading from s_i to the node with degree x_i .

Examples

allempty = $\{(G, s, \mathbf{x}) : \forall i x_i = \epsilon\} \in \text{MAD}$

Each agent tests its input and accepts iff $x_i = \epsilon$.

treesize = $\{(G, s, \mathbf{x}) : \forall i x_i = |V(G)| \text{ and } G \text{ is a tree}\} \in \text{MAD}$

Each agent checks independently if it is in a tree of size x_i (DFS traversal).

degree = $\{(G, s, \mathbf{x}) : \forall i \exists v d_v = x_i\} \in \text{MAV}$

Certificate: path leading from s_i to the node with degree x_i .

In fact all agents can **always** decide correctly for treesize.

Strict decidability

- We introduce the “strict” version of MAD, MAD_s :

Definition

$\Pi \in MAD_s$ if $\exists M \forall (G, s, \mathbf{x})$:

if $(G, s, \mathbf{x}) \in \Pi$, then $\forall id M(id, G, s, \mathbf{x}) = \text{all-yes}$

if $(G, s, \mathbf{x}) \notin \Pi$, then $\forall id M(id, G, s, \mathbf{x}) = \text{all-no}$

Strict decidability

- We introduce the “strict” version of MAD, MAD_s :

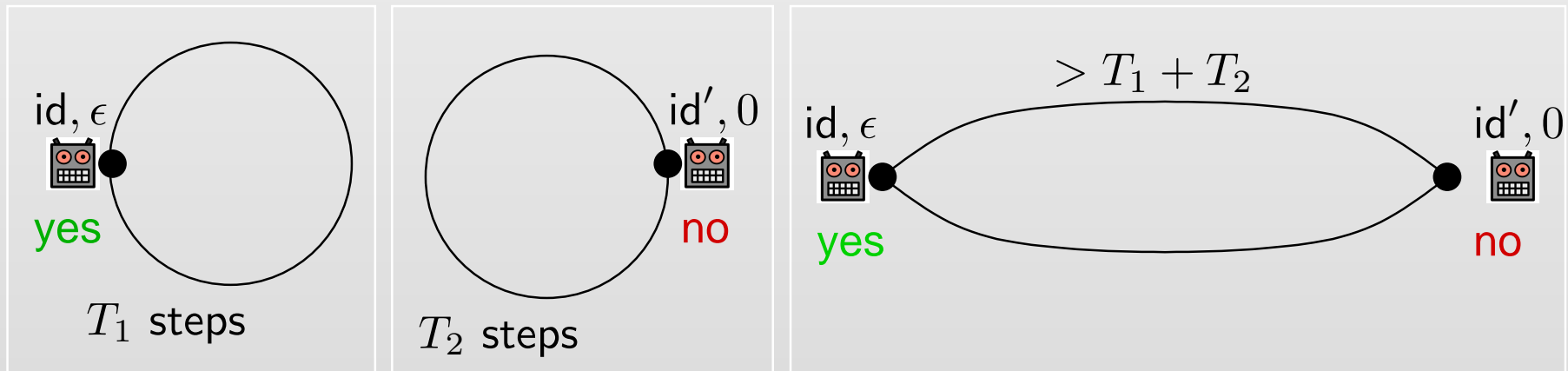
Definition

$\Pi \in MAD_s$ if $\exists M \forall (G, s, \mathbf{x})$:

if $(G, s, \mathbf{x}) \in \Pi$, then $\forall id \ M(id, G, s, \mathbf{x}) = \text{all-yes}$

if $(G, s, \mathbf{x}) \notin \Pi$, then $\forall id \ M(id, G, s, \mathbf{x}) = \text{all-no}$

- $treesize \in MAD_s$, but $allempty \notin MAD_s$.



Our contributions

- New computability classes below MAV and co-MAV. *
- Closure properties with respect to set operations.
- Metaprotocol for parallel execution of mobile agent protocols.

* **Reminder**

If X is a class of mobile agent decision problems, then

$$\text{co-}X = \{\Pi : \bar{\Pi} \in X\}$$

Verifiability classes

- MAV definition: \exists certificate that leads to acceptance for yes instances.
- In co-MAV, the acceptance mechanism is reversed:

Definition (co-MAV)

$\Pi \in \text{co-MAV}$ if $\exists M \forall (G, s, \mathbf{x})$:

if $(G, s, \mathbf{x}) \in \Pi$, then $\forall \mathbf{y} \forall \text{id} M(\text{id}, G, s, \langle \mathbf{x}, \mathbf{y} \rangle) = \text{some-yes}$

if $(G, s, \mathbf{x}) \notin \Pi$, then $\exists \mathbf{y} \forall \text{id} M(\text{id}, G, s, \langle \mathbf{x}, \mathbf{y} \rangle) = \text{all-no}$

Verifiability classes

- MAV definition: \exists certificate that leads to acceptance for yes instances.
- In co-MAV, the acceptance mechanism is reversed:

Definition (co-MAV)

$\Pi \in \text{co-MAV}$ if $\exists M \forall (G, s, \mathbf{x})$:

if $(G, s, \mathbf{x}) \in \Pi$, then $\forall \mathbf{y} \forall \text{id} M(\text{id}, G, s, \langle \mathbf{x}, \mathbf{y} \rangle) = \text{some-yes}$

if $(G, s, \mathbf{x}) \notin \Pi$, then $\exists \mathbf{y} \forall \text{id} M(\text{id}, G, s, \langle \mathbf{x}, \mathbf{y} \rangle) = \text{all-no}$

- “No” certificate with the same acceptance mechanism as in MAV:

Definition (co-MAV')

$\Pi \in \text{co-MAV}'$ if $\exists M \forall (G, s, \mathbf{x})$:

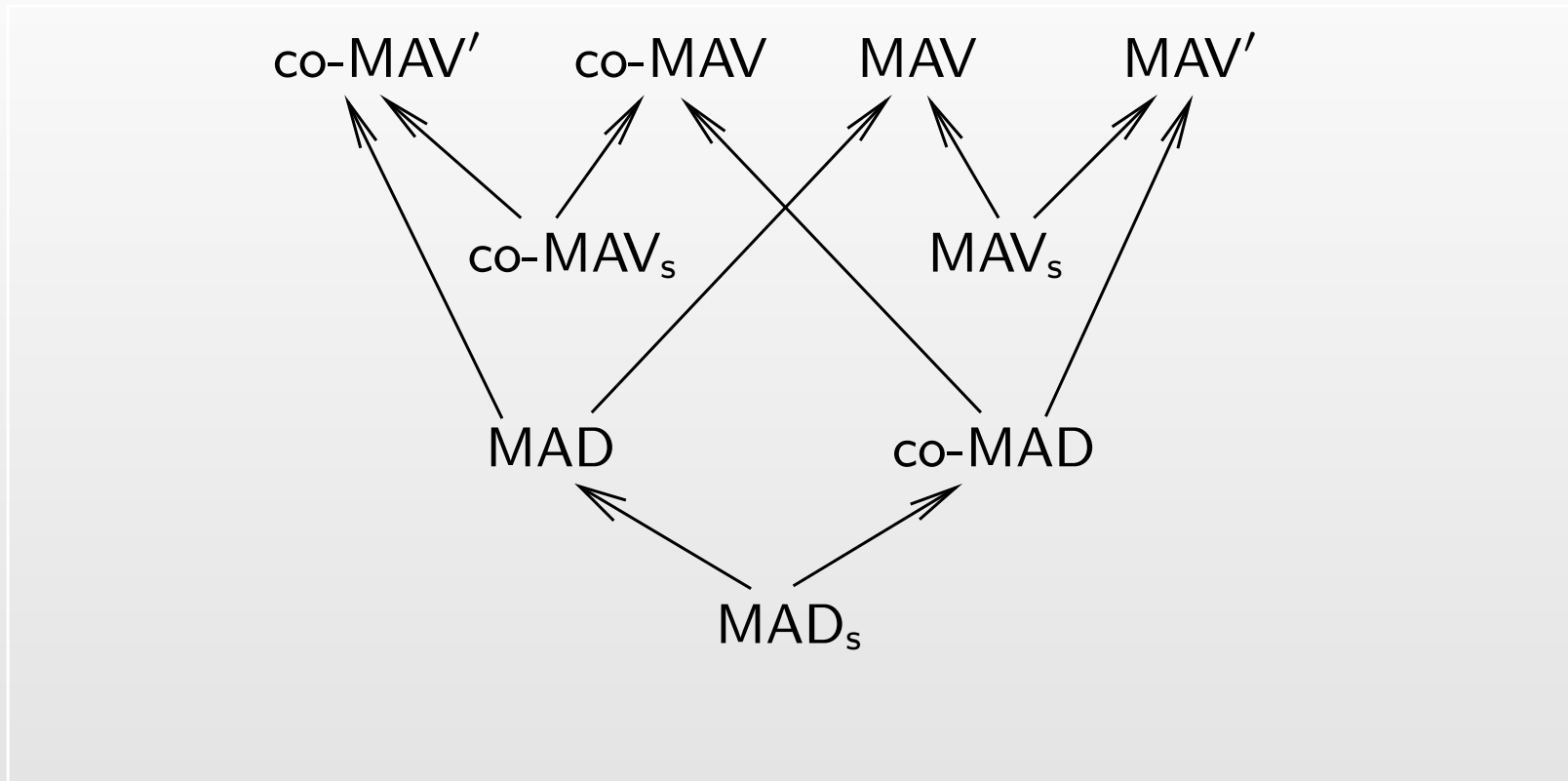
if $(G, s, \mathbf{x}) \in \Pi$, then $\forall \mathbf{y} \forall \text{id} M(\text{id}, G, s, \langle \mathbf{x}, \mathbf{y} \rangle) = \text{all-yes}$

if $(G, s, \mathbf{x}) \notin \Pi$, then $\exists \mathbf{y} \forall \text{id} M(\text{id}, G, s, \langle \mathbf{x}, \mathbf{y} \rangle) = \text{some-no}$

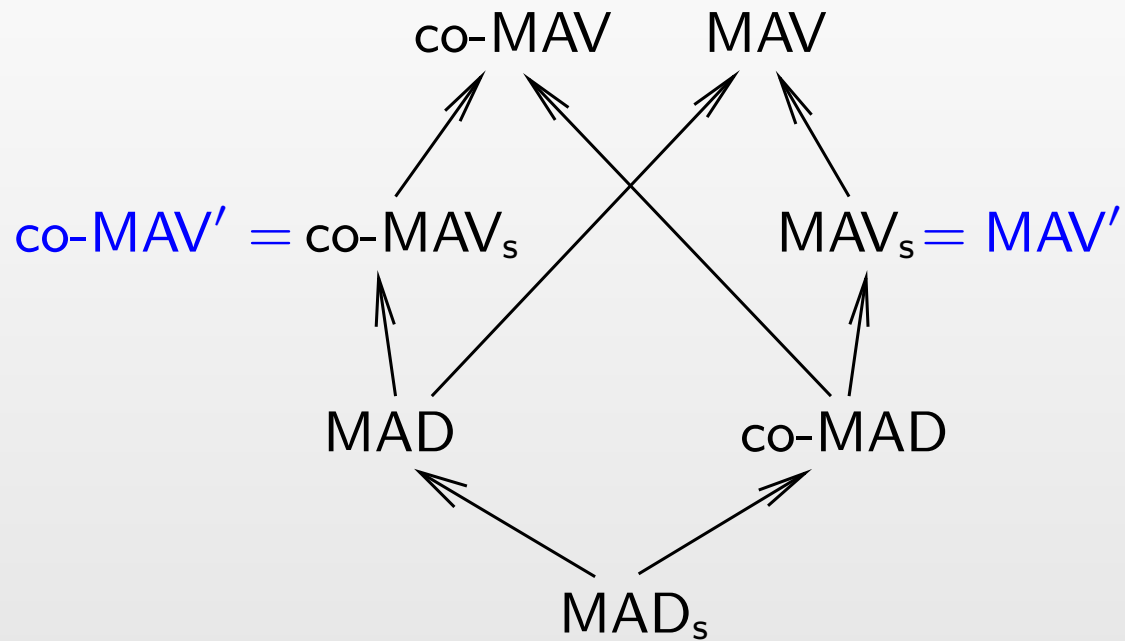
Class overview

	“yes” instances	“no” instances
MAD _s	($\forall y$) all-yes	($\forall y$) all-no
MAD	($\forall y$) all-yes	($\forall y$) some-no
co-MAD	($\forall y$) some-yes	($\forall y$) all-no
MAV _s	$\exists y$ all-yes	$\forall y$ all-no
co-MAV _s	$\forall y$ all-yes	$\exists y$ all-no
MAV	$\exists y$ all-yes	$\forall y$ some-no
co-MAV	$\forall y$ some-yes	$\exists y$ all-no
MAV'	$\exists y$ some-yes	$\forall y$ all-no
co-MAV'	$\forall y$ all-yes	$\exists y$ some-no

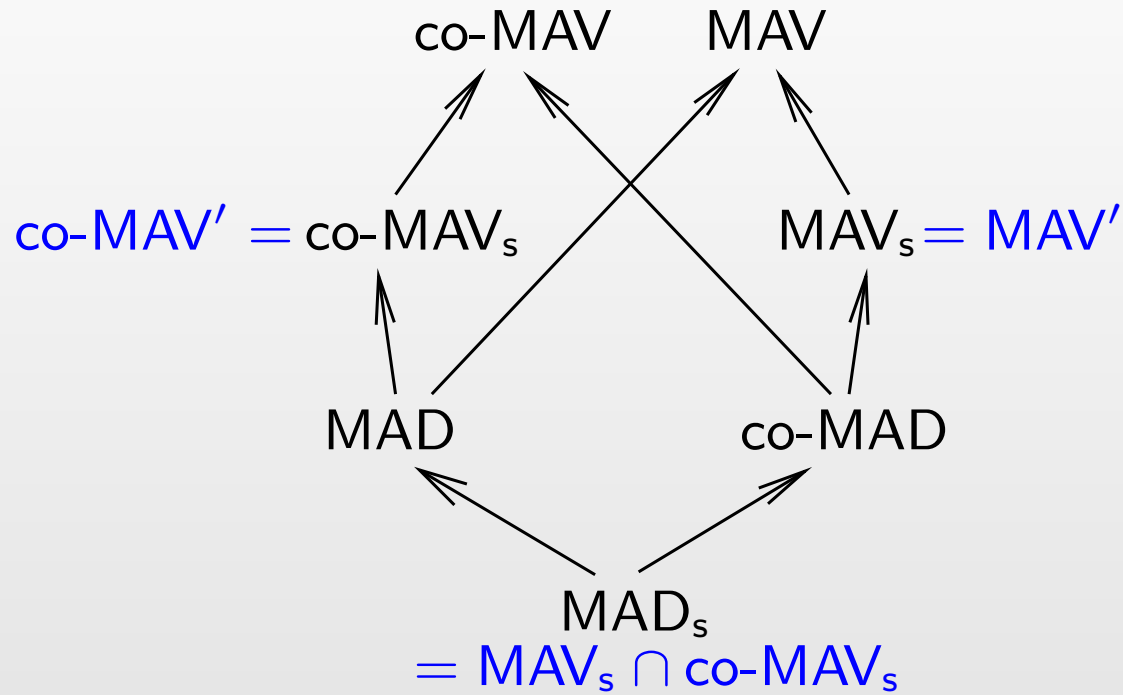
Inclusions by definition



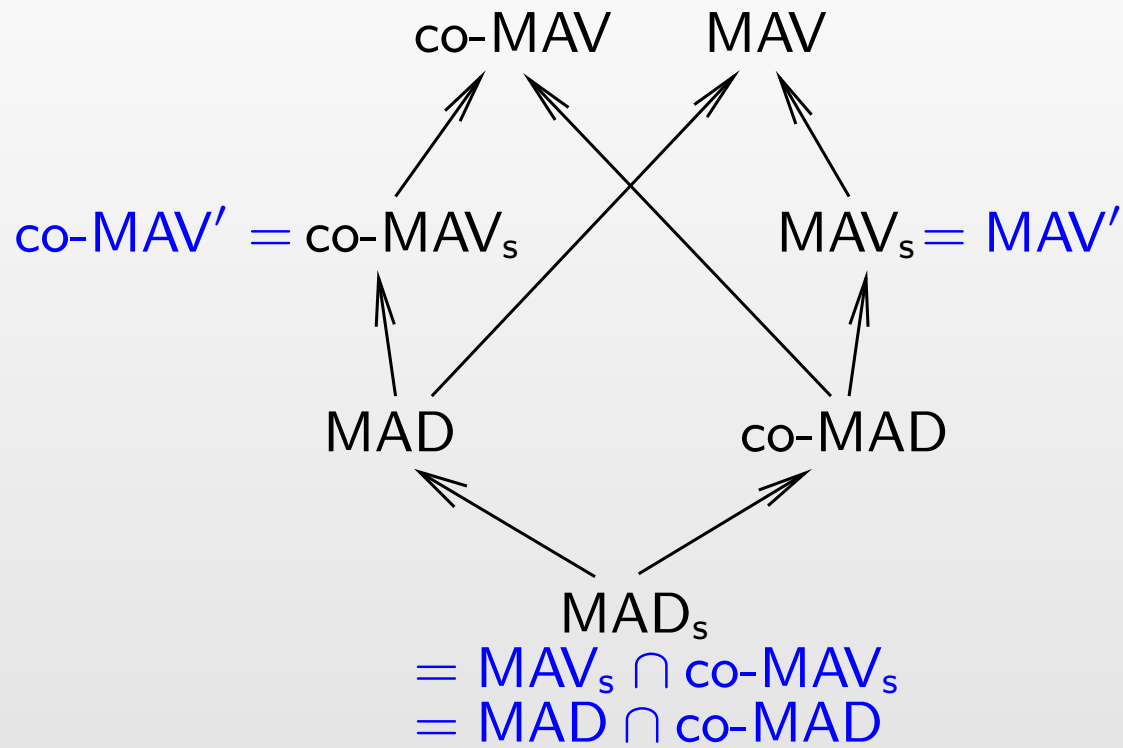
Our results (I)



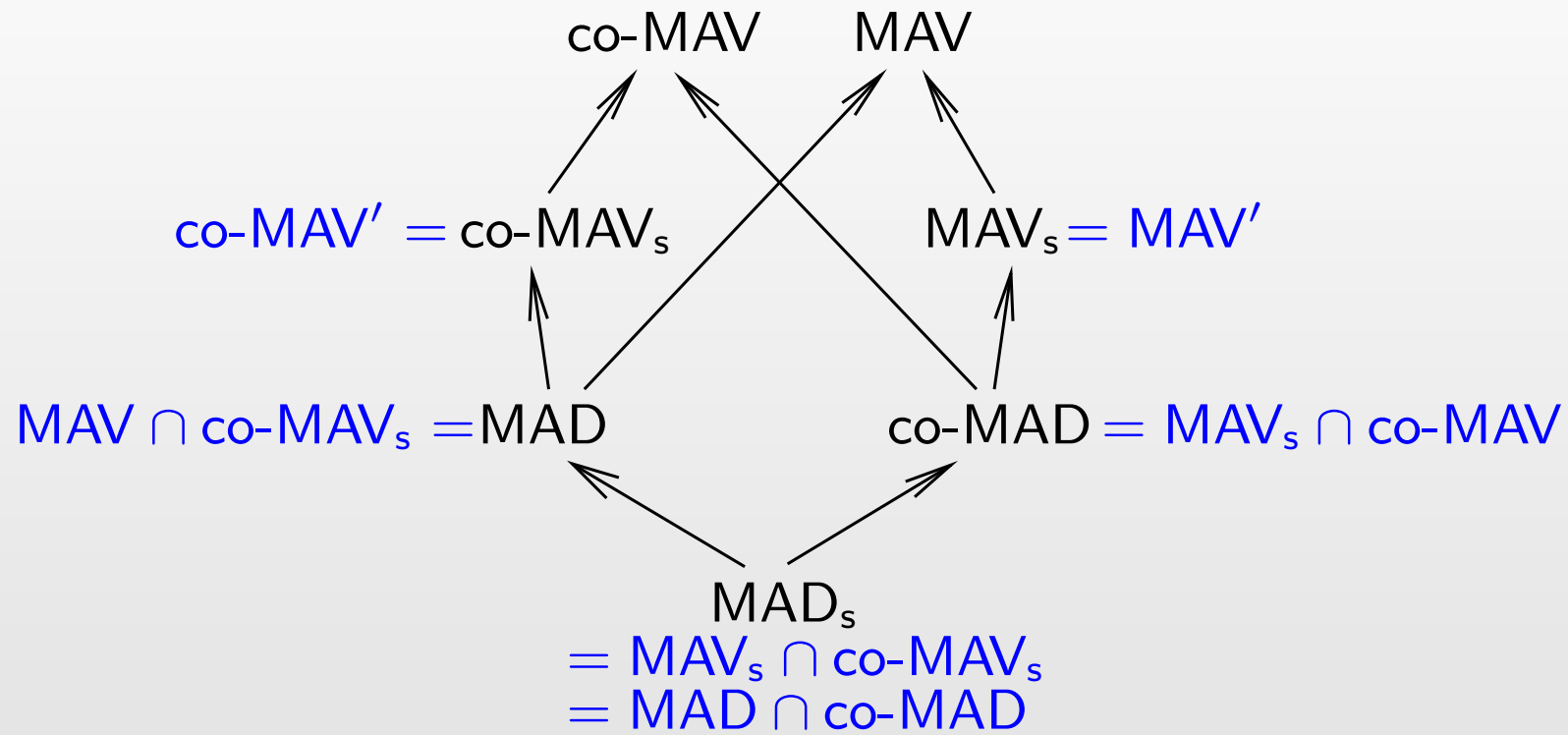
Our results (I)



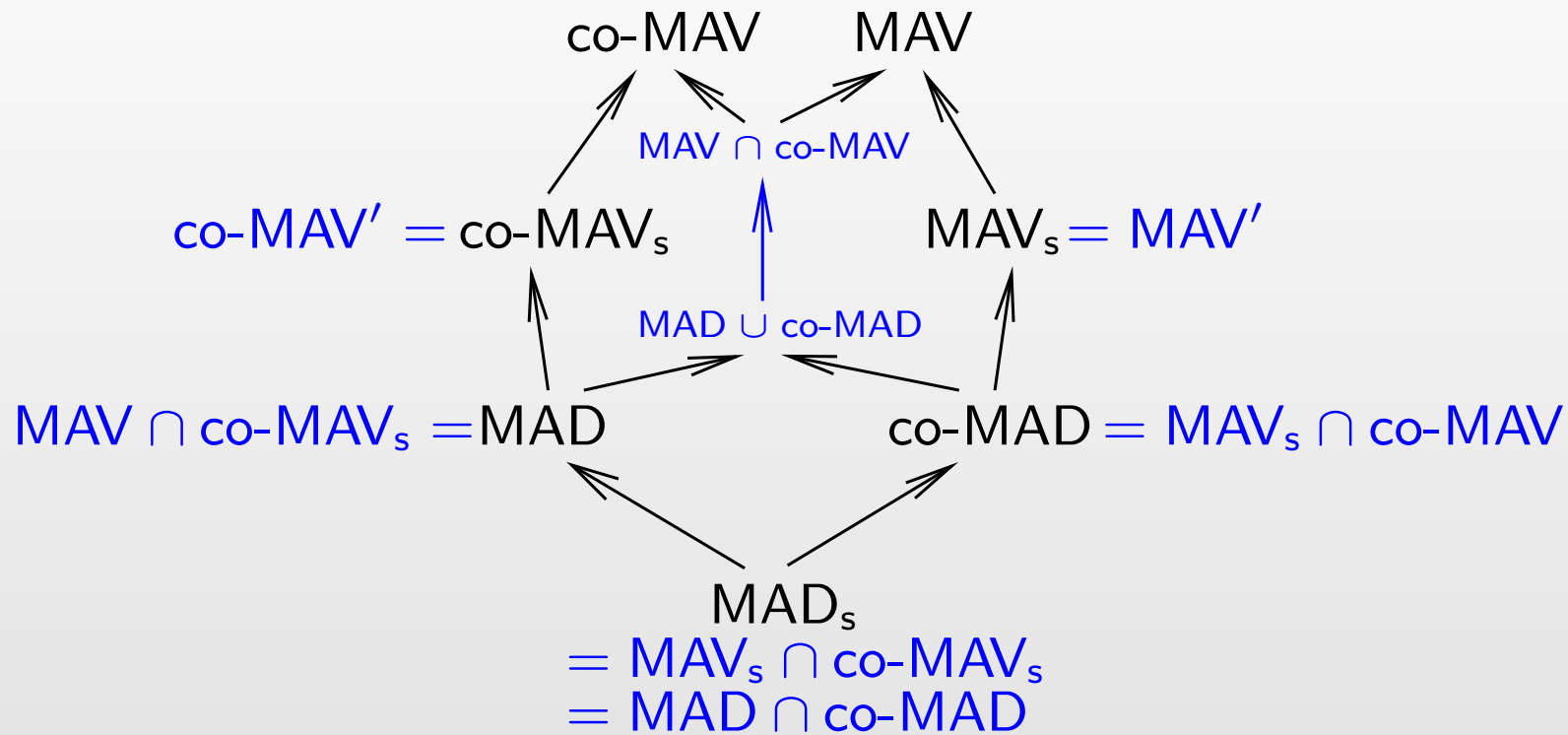
Our results (I)



Our results (I)

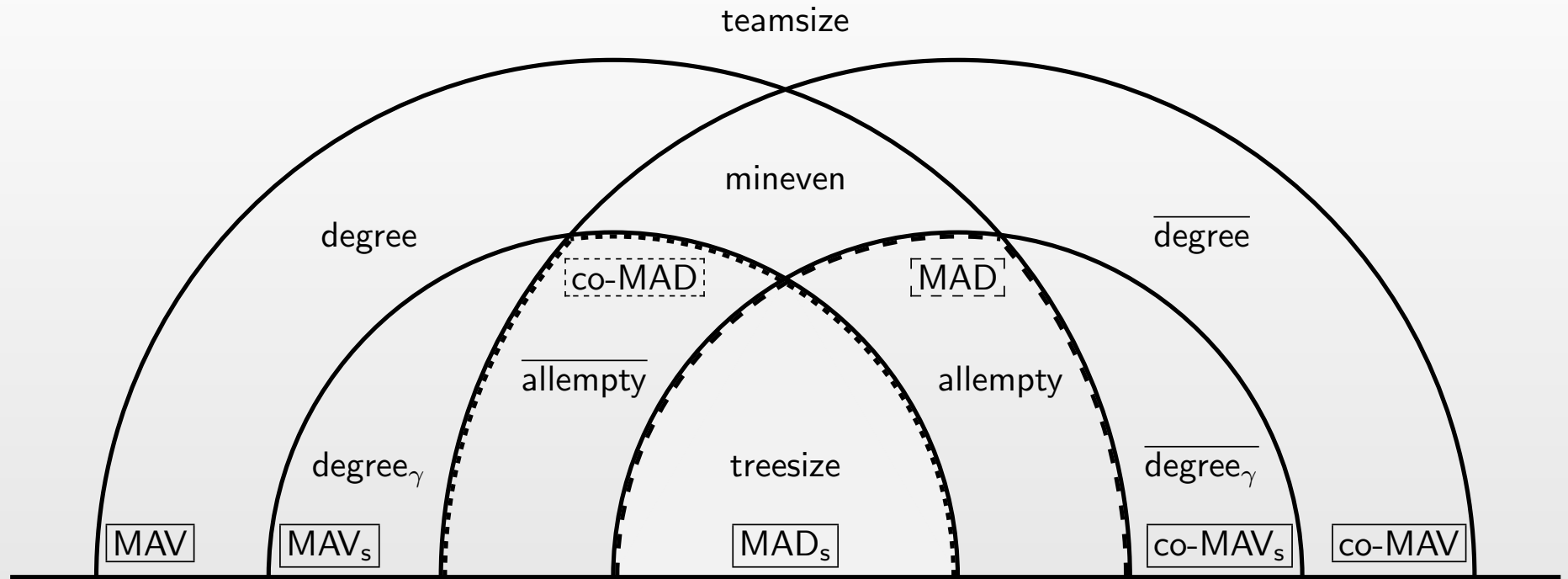


Our results (I)



- All inclusions are strict.
- We can separate all pairs of classes in the diagram.

Another point of view



Characterization of decidability classes

Theorem [Fraigniaud and Pelc 2012]

$$\text{MAD}_1 = \text{MAV}_1 \cap \text{co-MAV}_1.$$

Our decidability class characterizations

$$\text{MAD}_s = \text{MAV}_s \cap \text{co-MAV}_s$$

$$\text{MAD} = \text{MAV} \cap \text{co-MAV}_s$$

$$\text{co-MAD} = \text{MAV}_s \cap \text{co-MAV}$$

can be seen as generalizations of the above Theorem for multi-agent protocols.

Our results (II)

Closure under standard set-theoretic operations:

	Union	Intersection	Complement
MAD_s	✓	✓	✓
MAD	✗	✓	✗
co-MAD	✓	✗	✗
MAV_s	✓	✓	✗
co- MAV_s	✓	✓	✗
MAV	✗	✓	✗
co-MAV	✓	✗	✗

Classical dovetailing

$M_1(x_1)$

$M_2(x_2)$

$M_3(x_3)$

...



⋮

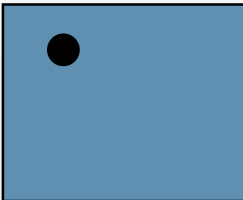
Classical dovetailing

$M_1(x_1)$

$M_2(x_2)$

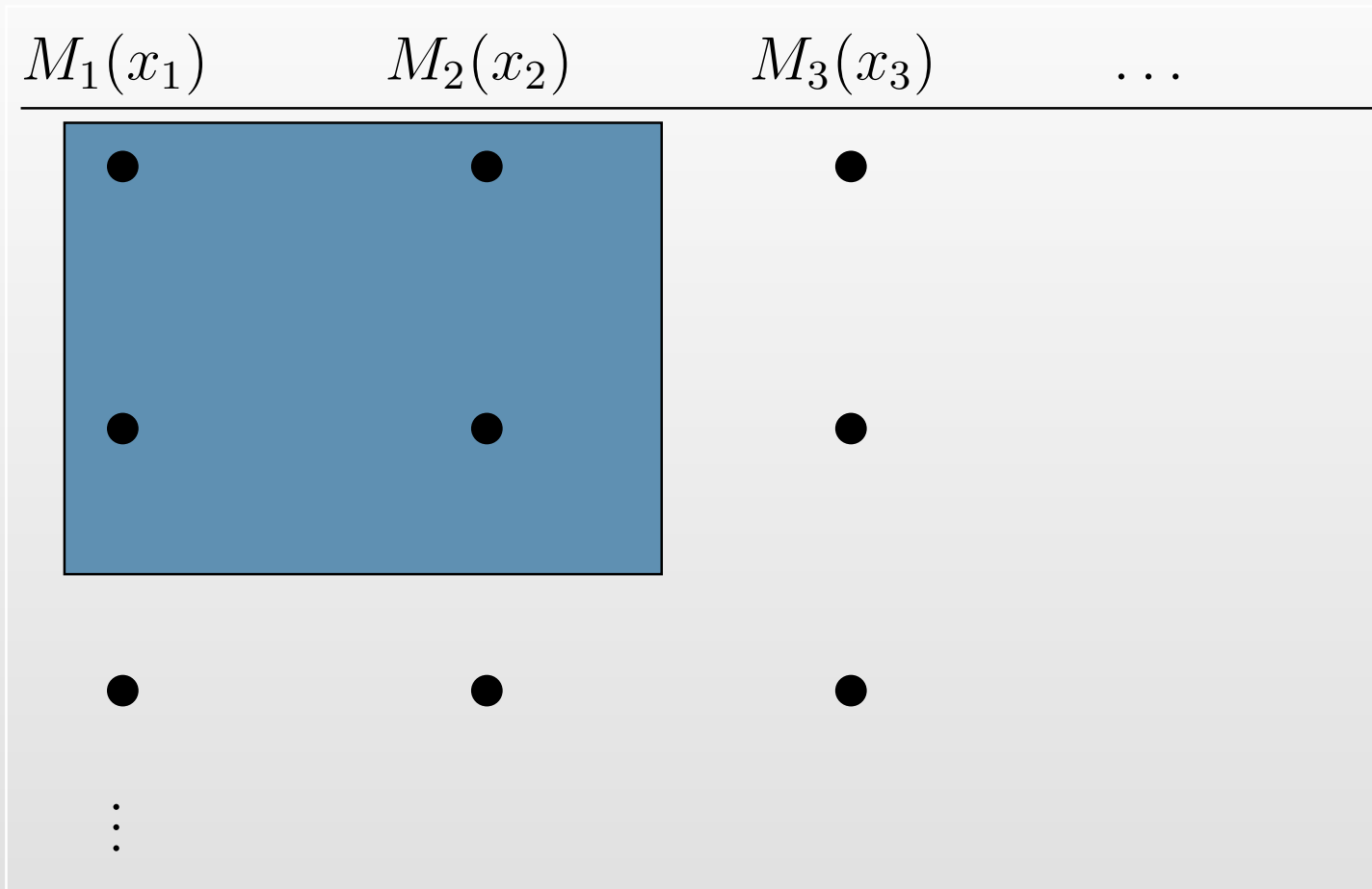
$M_3(x_3)$

...



⋮

Classical dovetailing



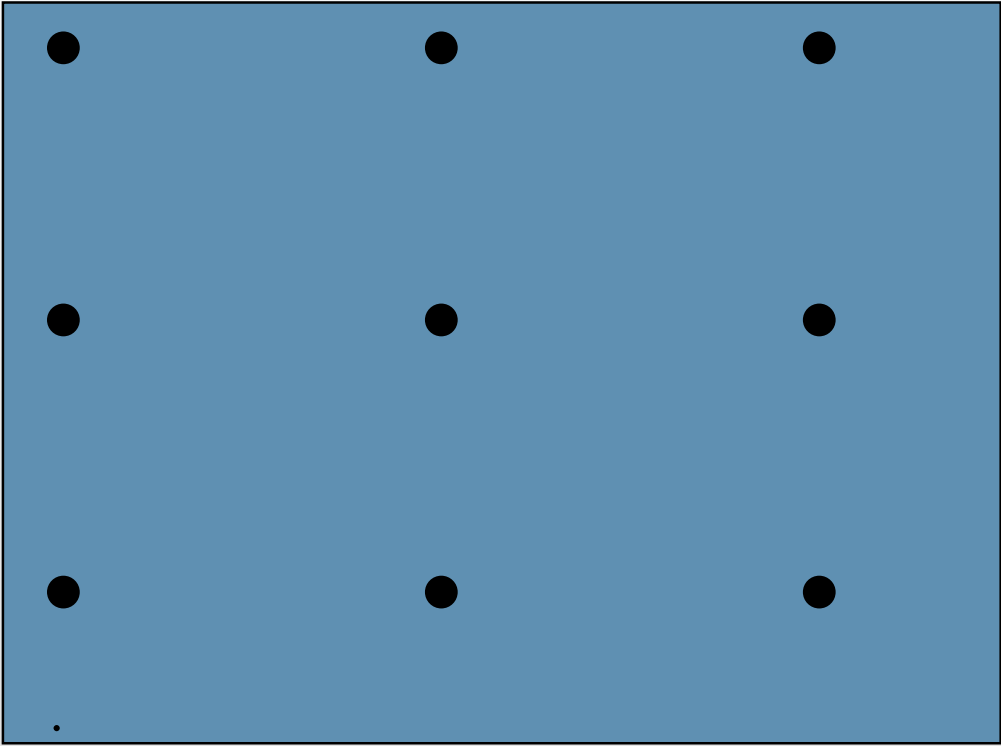
Classical dovetailing

$M_1(x_1)$

$M_2(x_2)$

$M_3(x_3)$

...



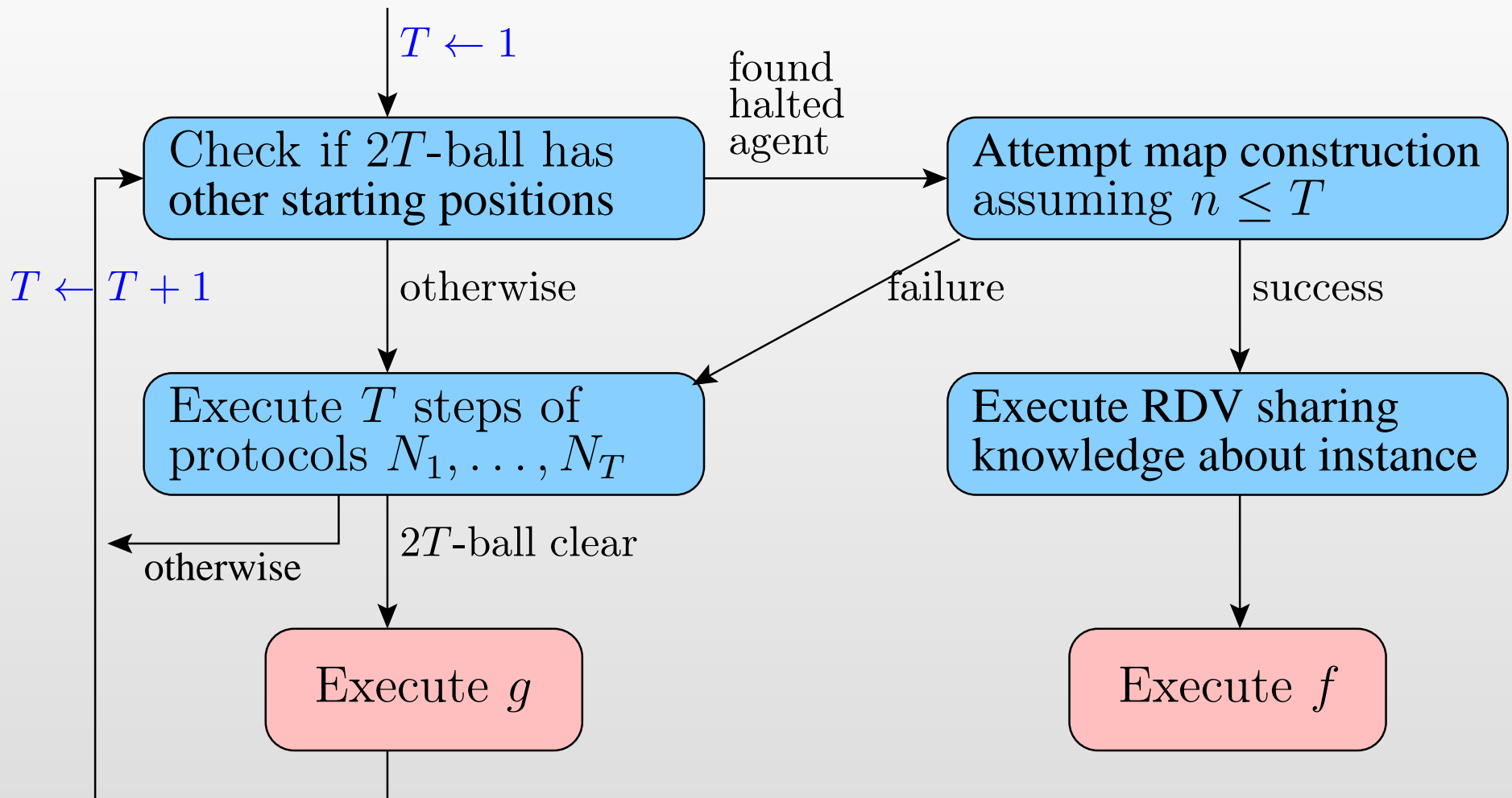
Meta-protocol

- Permits the execution of a number of mobile agent protocols essentially in parallel.
- The agents accept or reject depending on the outcome of the executions of the individual protocols.

Parameters

- $\mathcal{N} = (N_1, N_2, \dots)$: list of protocols to execute (possibly infinite).
- Global decider $f(\text{id}, G, \mathbf{s}, \mathbf{x})$: **yes** or **no** with full knowledge.
- Local decider $g(H_1, \dots, H_\sigma)$: **yes** or **no** or **indecisive** based on the executions of the protocols.

Meta-protocol



Future research

- Extend the hierarchy MAD \rightarrow MAV \rightarrow
- Resource-bounded computations.
- Asynchronous computations.

Future research

- Extend the hierarchy MAD \rightarrow MAV \rightarrow
- Resource-bounded computations.
- Asynchronous computations.

Thank you!