On mobile agents verifiable problems

(published in LATIN'16)

Evangelos Bampas[1] David Ilcinkas[2]

[1]Aix-Marseille Univ. (LIF), France
[2]CNRS & Univ. Bordeaux (LaBRI), France

Micro MAC
September 26, 2016

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 properly 3-colored?

In distributed computing by mobile agents

- The topic of this talk.

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 properly 3-colored?

In distributed computing by mobile agents

- The topic of this talk.

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 properly 3-colored?

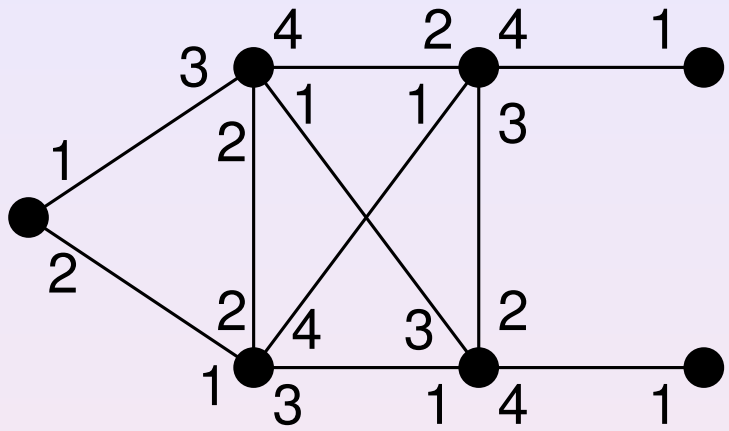
In distributed computing by mobile agents

- The topic of this talk.

Mobile agent computational model

Network

- Connected.
- Anonymous.
- Local port numbering.



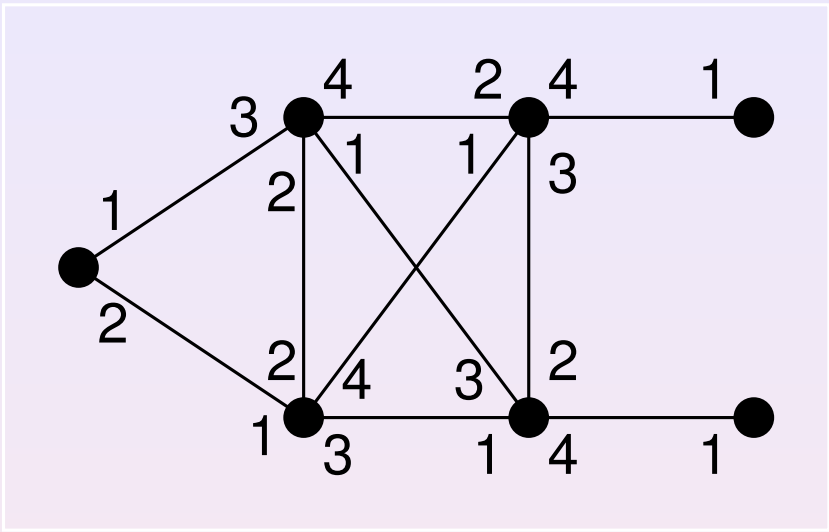
Agents

- 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 current memory state, degree, incoming port number, configurations of collocated agents.
  - Halt (yes/no), stay idle, or exit through one of the ports.

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 current memory state, 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 [Fraigniaud and Pelc, LATIN 2012]

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

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

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

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

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 $\forall \mathrm{id}\ M(\mathrm{id}, G, \mathbf{s}, \mathbf{x}) =$ all-yes

if $(G, \mathbf{s}, \mathbf{x}) \notin \Pi$, then $\forall \mathrm{id}\ M(\mathrm{id}, G, \mathbf{s}, \mathbf{x}) =$ 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

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

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 $\forall \mathrm{id}\ M(\mathrm{id}, G, \mathbf{s}, \mathbf{x}) =$ all-yes

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

Mobile Agent Verifiable problems (class MAV)

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

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

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

... and verifiability

[Fraigniaud and Pelc, LATIN 2012]

Mobile Agent Decidable problems (class MAD)

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

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

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

Mobile Agent Verifiable problems (class MAV)

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

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

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

Examples

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

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

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

Certificate: path leading from s_i to a degree-x_i node.

pathsize $= \{(G, \mathbf{s}, \mathbf{x}) : \forall i \ x_i = |V(G)|$ and G path$\} \in$ MAD

Each agents checks independently if it is in a path of size x_i.

In fact all agents can always decide correctly for pathsize.

Examples

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

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

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

Certificate: path leading from s_i to a degree-x_i node.

pathsize $= \{(G, \mathbf{s}, \mathbf{x}) : \forall i \ x_i = |V(G)|$ and G path$\} \in$ MAD

Each agents checks independently if it is in a path of size x_i.

In fact all agents can always decide correctly for pathsize.

Examples

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

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

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

Certificate: path leading from s_i to a degree-x_i node.

pathsize $= \{(G, \mathbf{s}, \mathbf{x}) : \forall i \ x_i = |V(G)| \ \mathbf{and} \ G \ \mathbf{path}\} \in$ MAD

Each agents checks independently if it is in a path of size x_i.

In fact all agents can always decide correctly for pathsize.

Examples

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

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

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

Certificate: path leading from s_i to a degree-x_i node.

pathsize $= \{(G, \mathbf{s}, \mathbf{x}) : \forall i \ x_i = |V(G)| \ \mathbf{and} \ G \ \mathbf{path}\} \in$ MAD

Each agents checks independently if it is in a path of size x_i.

In fact all agents can always decide correctly for pathsize.

Strict decidability

- We introduce the "strict" version of MAD, MAD_s:

Definition

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

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

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

- pathsize $\in \text{MAD}_s$, but allempty $\notin \text{MAD}_s$.

Strict decidability

- We introduce the "strict" version of MAD, MAD_s:

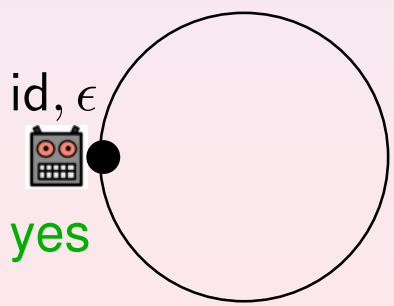
Definition

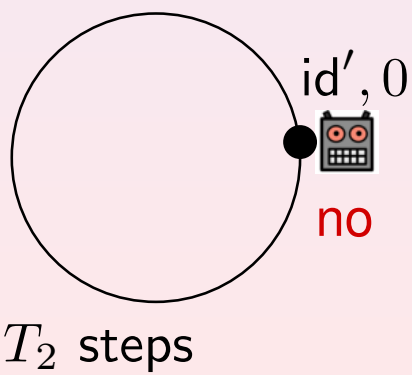
$\Pi \in \text{MAD}_s$ if $\exists M\ \forall(G, \mathbf{s}, \mathbf{x})$:

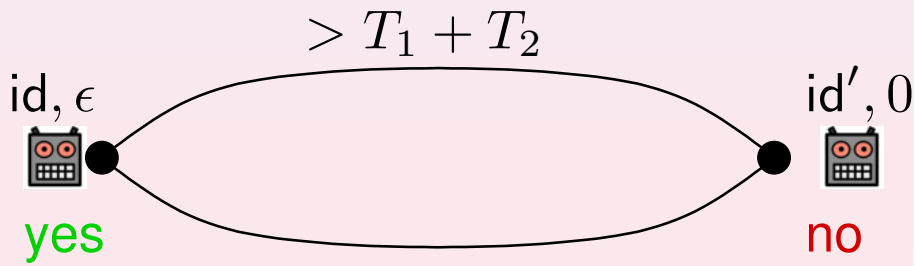
if $(G, \mathbf{s}, \mathbf{x}) \in \Pi$, then $\forall \text{id}\ M(\text{id}, G, \mathbf{s}, \mathbf{x}) = \text{all-yes}$

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

- pathsize $\in \text{MAD}_s$, but allempty $\notin \text{MAD}_s$.







Our contributions

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

*** Reminder**

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

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

Verifiability classes

MAV def.: $\exists$ cert. that leads to acceptance for yes instances.

In co-MAV, the acceptance mechanism is reversed:

Definition (co-MAV)

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

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

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

"No" cert. with the same acceptance mechanism as in MAV:

Definition (co-MAV')

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

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

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

Verifiability classes

MAV def.: $\exists$ cert. that leads to acceptance for yes instances.

In co-MAV, the acceptance mechanism is reversed:

Definition (co-MAV)

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

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

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

"No" cert. with the same acceptance mechanism as in MAV:

Definition (co-MAV$'$)

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

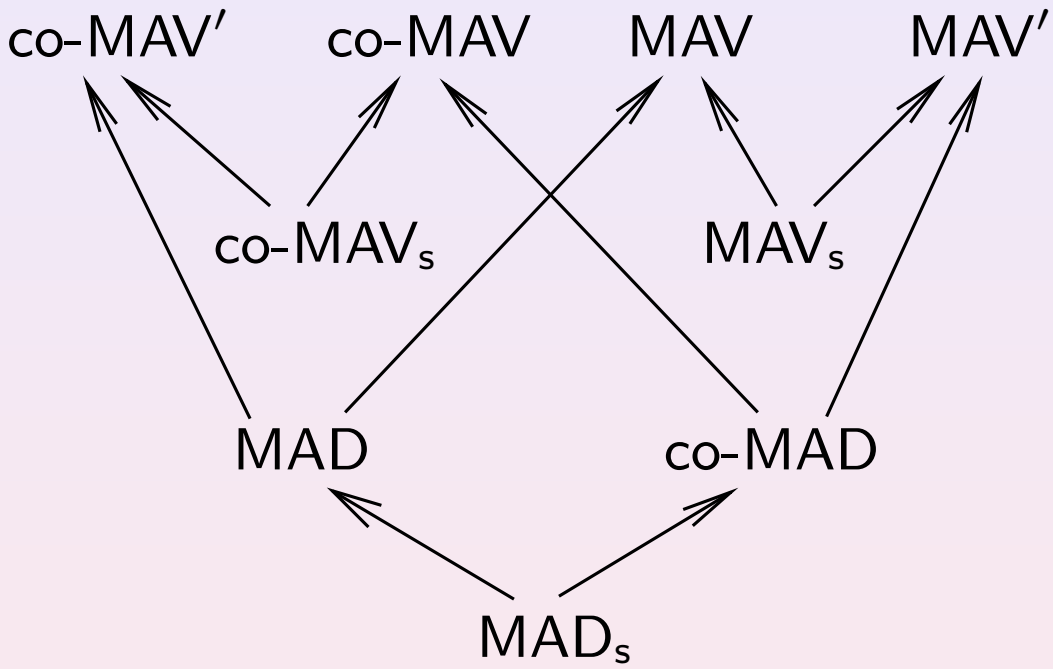
if $(G, \mathbf{s}, \mathbf{x}) \in \Pi$, then $\forall \mathbf{y}\ \forall \mathrm{id}\ M(\mathrm{id}, G, \mathbf{s}, \langle \mathbf{x}, \mathbf{y} \rangle) =$ all-yes

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

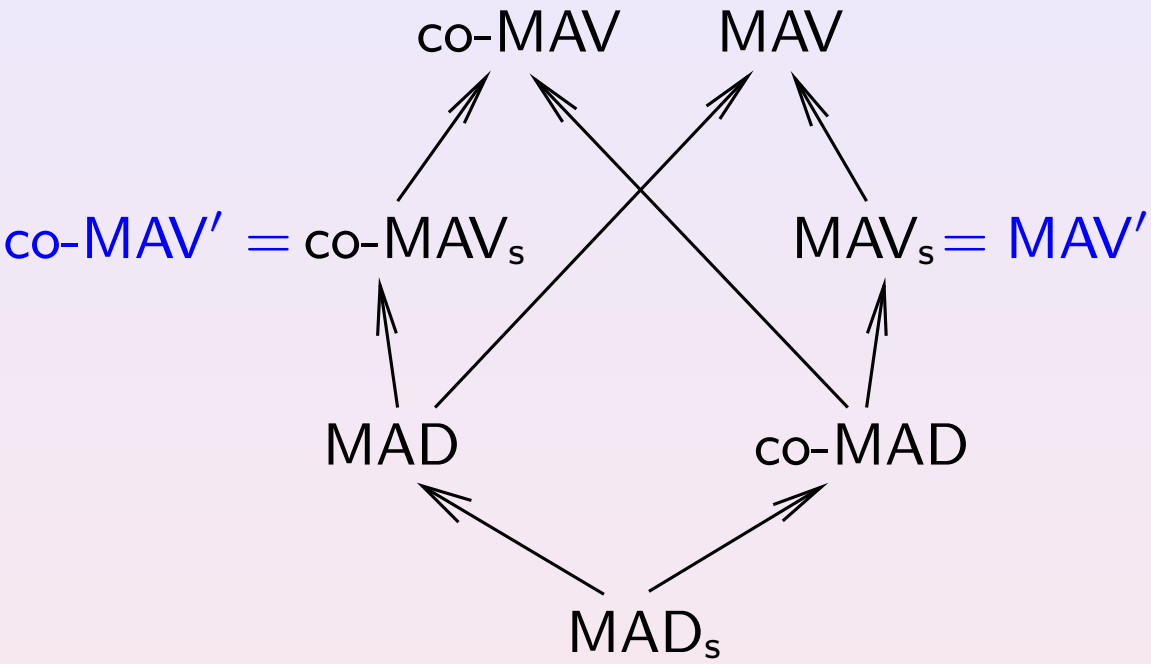
Class overview

	"yes" instances	"no" instances
MAD_s	$(\forall \mathbf{y})$ all-yes	$(\forall \mathbf{y})$ all-no
MAD	$(\forall \mathbf{y})$ all-yes	$(\forall \mathbf{y})$ some-no
co-MAD	$(\forall \mathbf{y})$ some-yes	$(\forall \mathbf{y})$ all-no
MAV_s	$\exists \mathbf{y}$ all-yes	$\forall \mathbf{y}$ all-no
co-MAV_s	$\forall \mathbf{y}$ all-yes	$\exists \mathbf{y}$ all-no
MAV	$\exists \mathbf{y}$ all-yes	$\forall \mathbf{y}$ some-no
co-MAV	$\forall \mathbf{y}$ some-yes	$\exists \mathbf{y}$ all-no
MAV′	$\exists \mathbf{y}$ some-yes	$\forall \mathbf{y}$ all-no
co-MAV′	$\forall \mathbf{y}$ all-yes	$\exists \mathbf{y}$ some-no

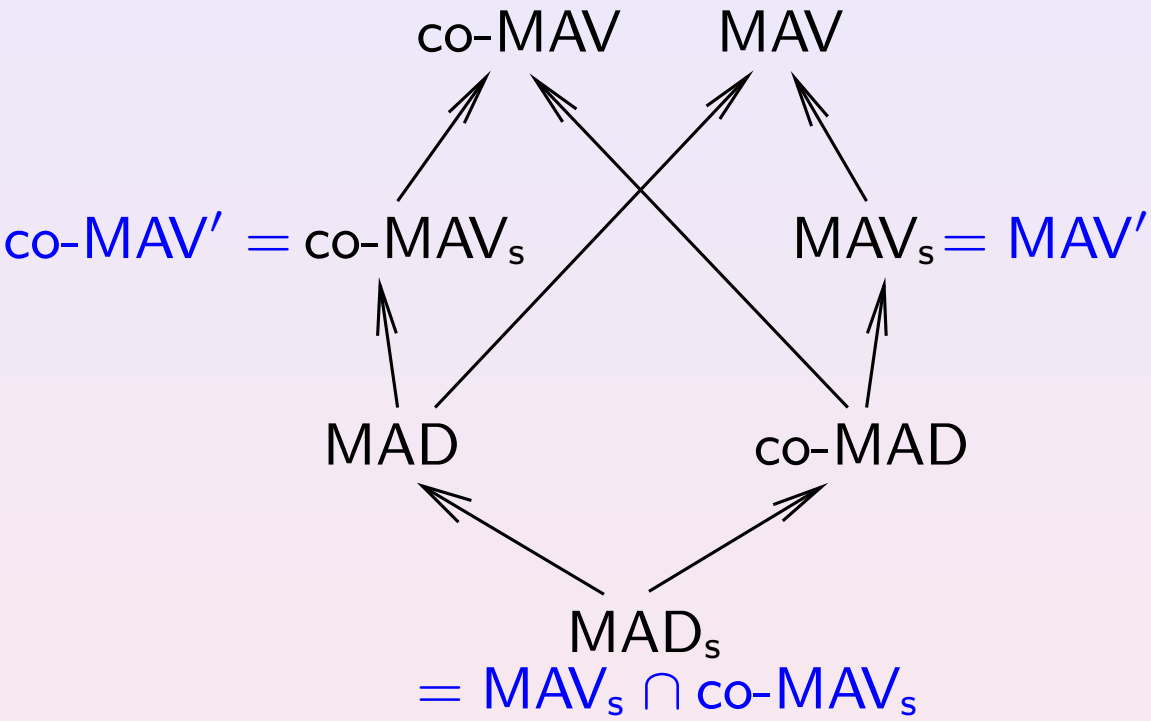
Inclusions by definition



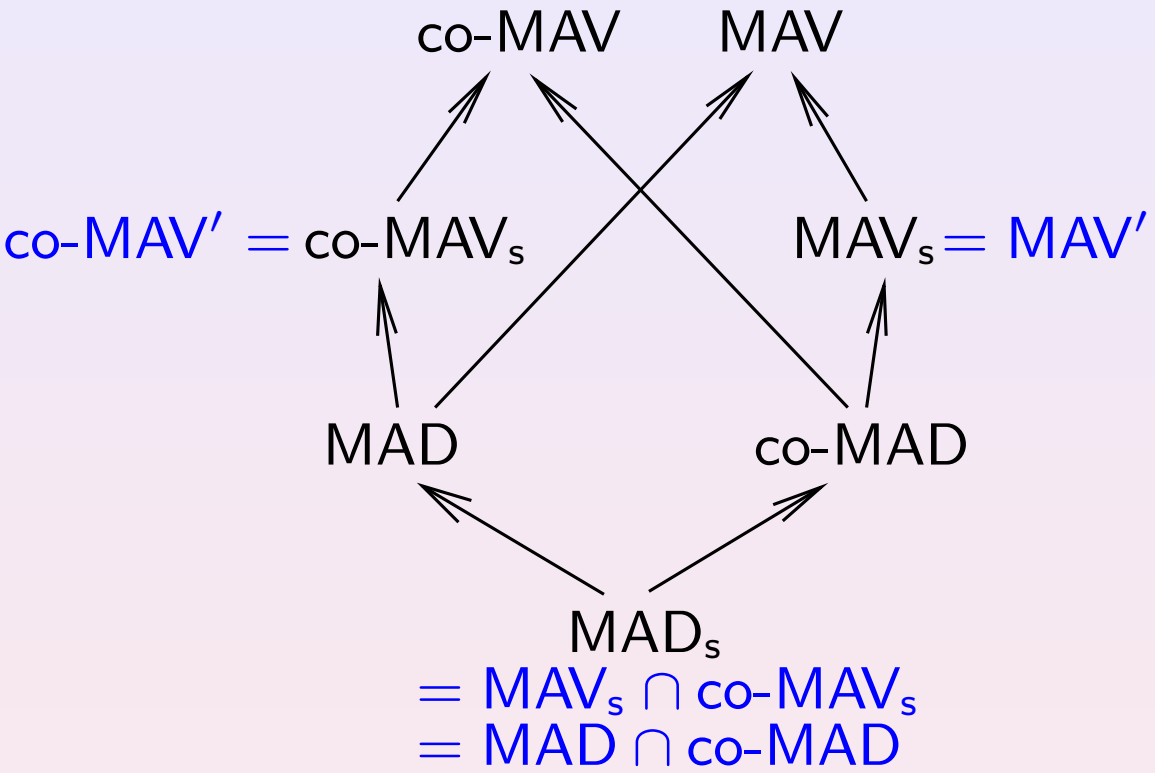
Our results (I)



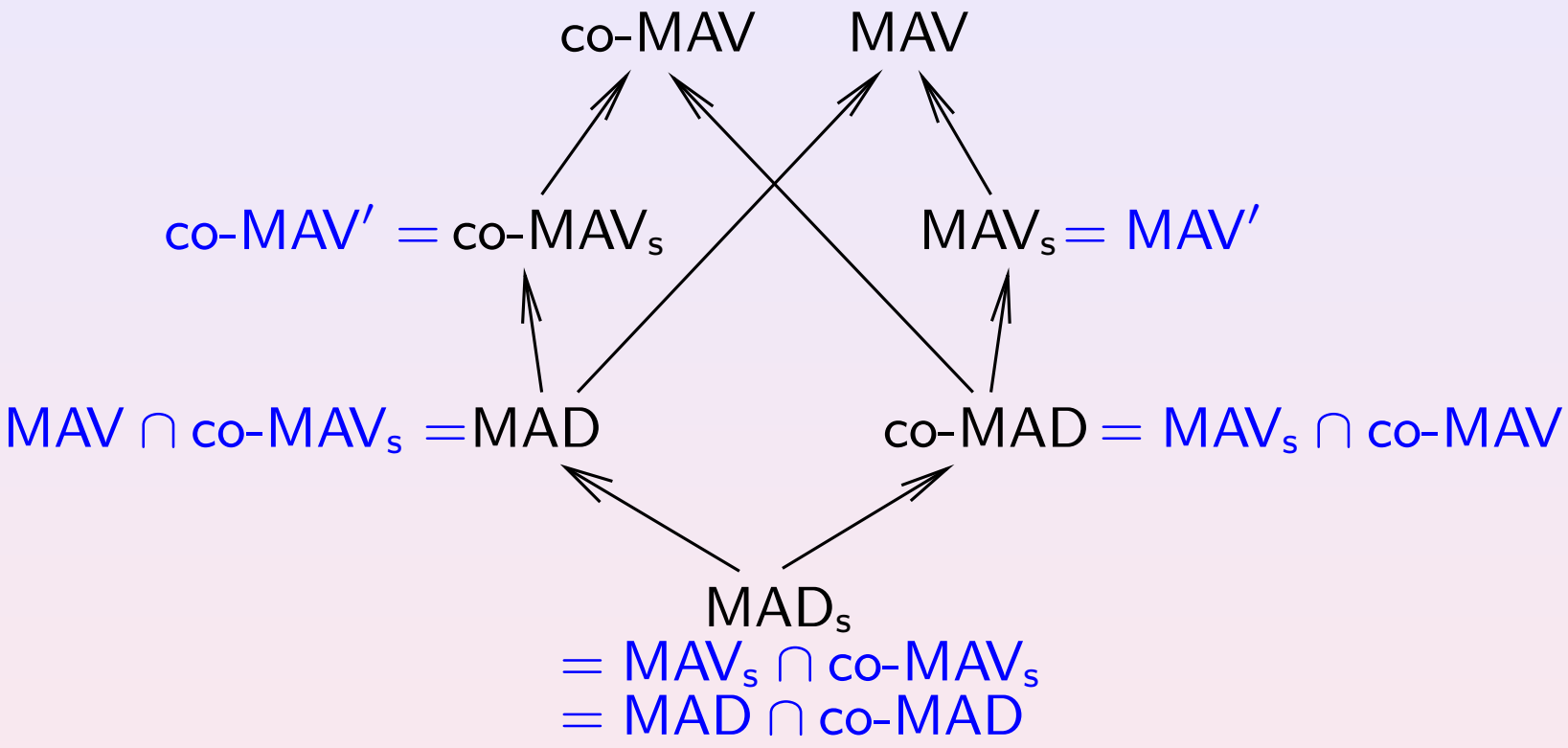
Our results (I)



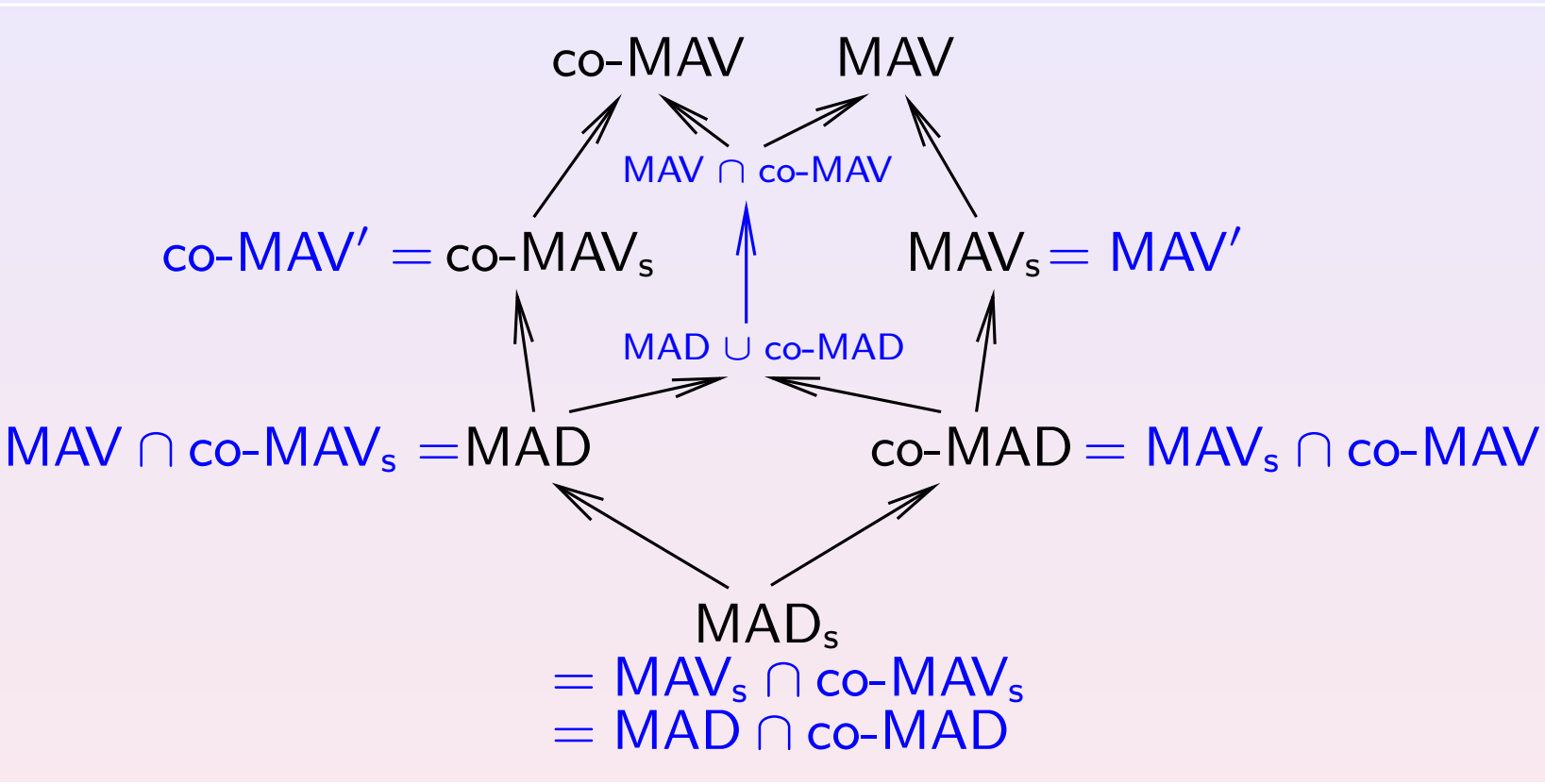
Our results (I)



Our results (I)

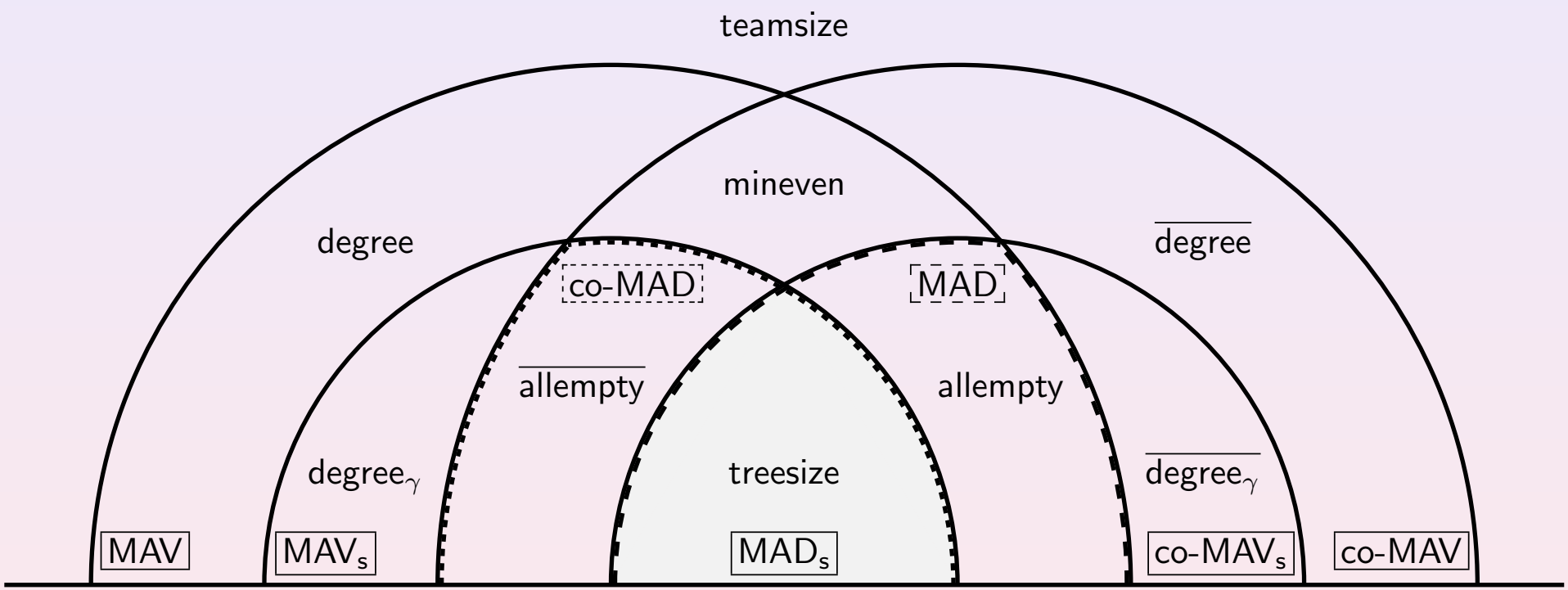


Our results (I)



- 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]

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

Our decidability class characterizations

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

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	✓	✗	✗

Main tool: our meta-protocol

Meta-protocol for parallel execution of mobile agent protocols.

- Possibly infinite number of mobile agent protocols
- Mobile agent computing analogue of the classical dovetailing technique

Example of use

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

	"yes" instances	"no" instances
MAD_s	all-yes	all-no
MAD	all-yes	some-no
co-MAD	some-yes	all-no

Main tool: our meta-protocol

Meta-protocol for parallel execution of mobile agent protocols.

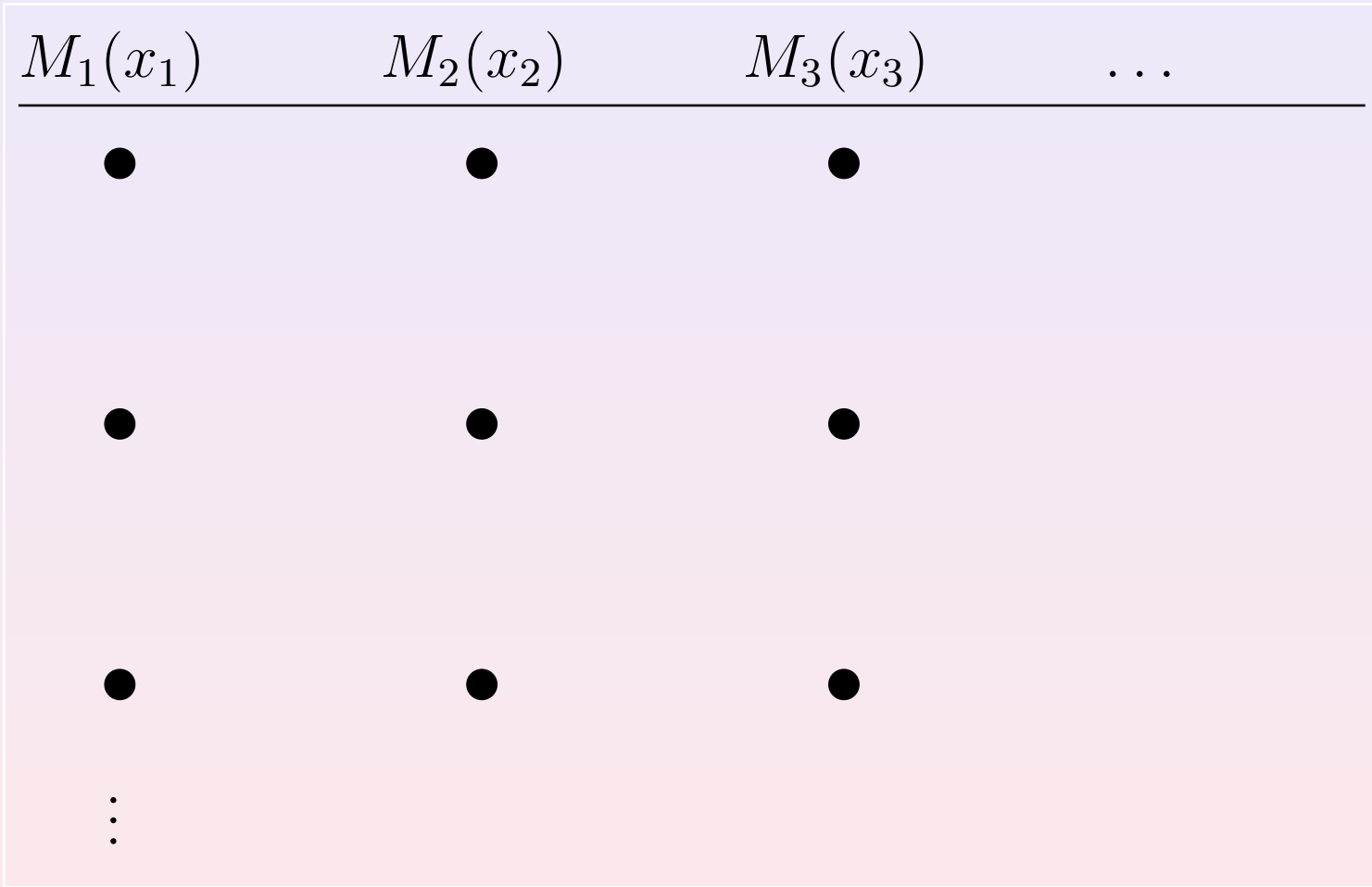
- Possibly infinite number of mobile agent protocols
- Mobile agent computing analogue of the classical dovetailing technique

Example of use

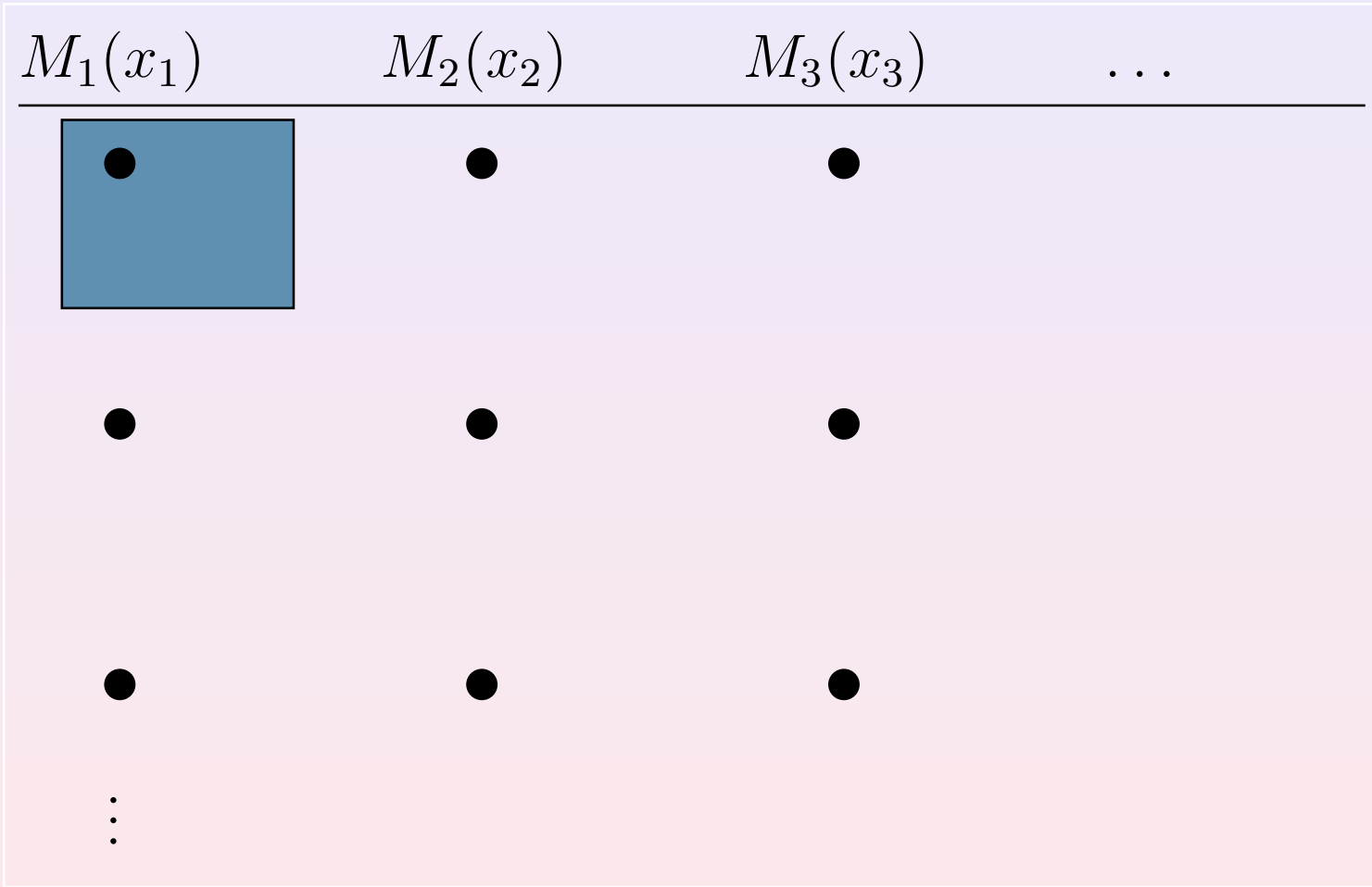
$\text{MAD}_s = \text{MAD} \cap \text{co-MAD}$

	"yes" instances	"no" instances
MAD_s	all-yes	all-no
MAD	all-yes	some-no
co-MAD	some-yes	all-no

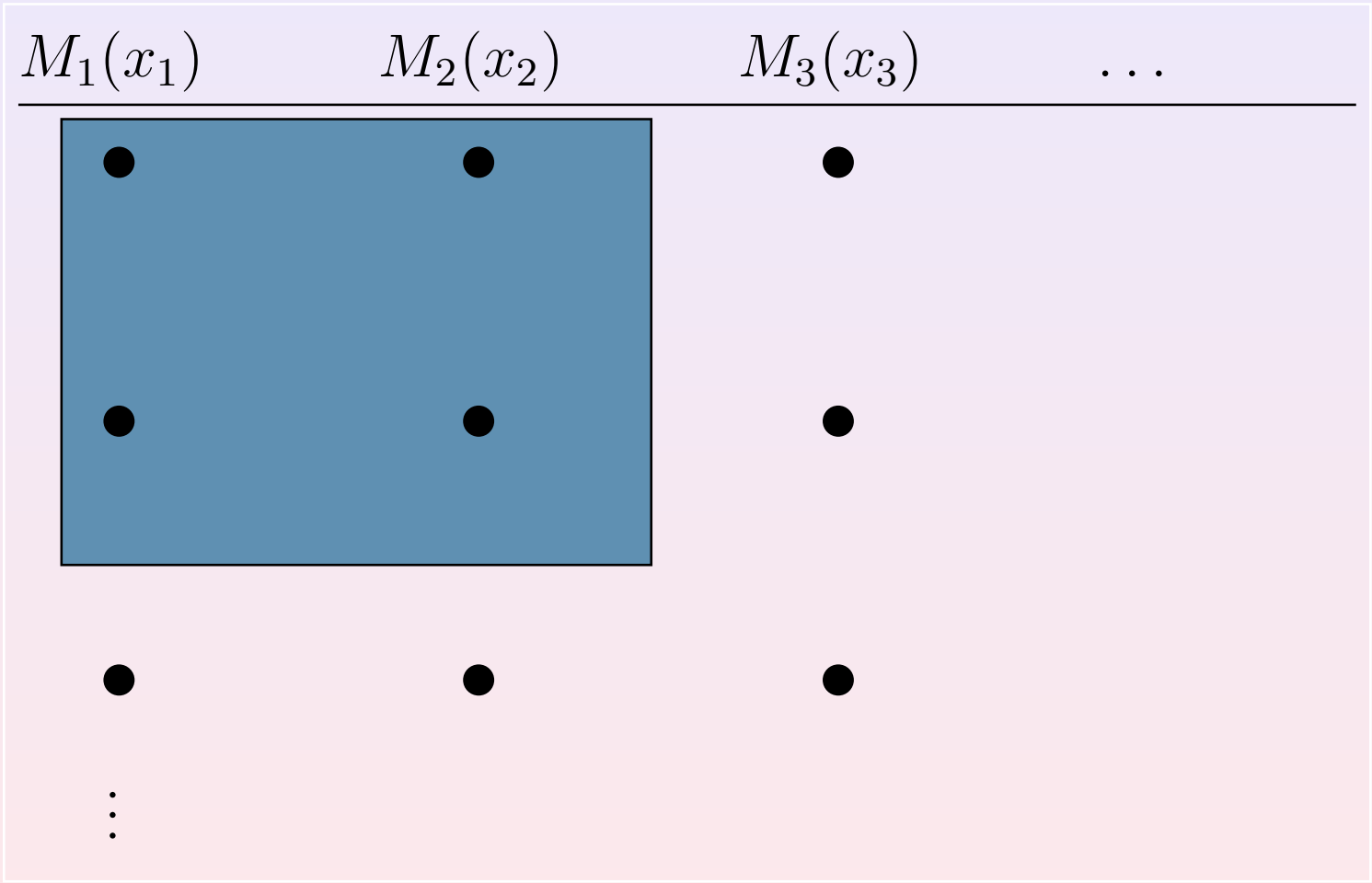
Classical dovetailing



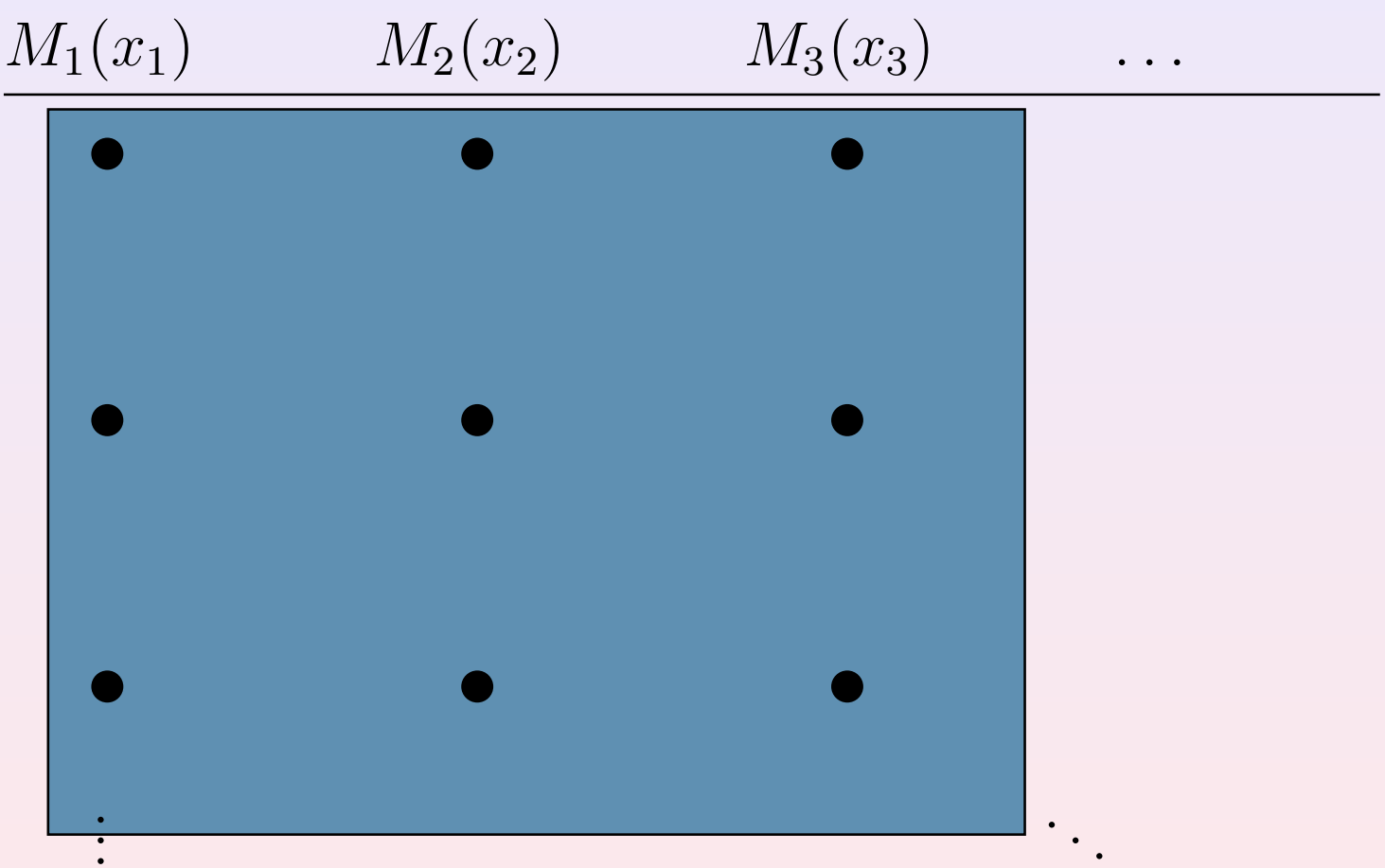
Classical dovetailing



Classical dovetailing



Classical dovetailing



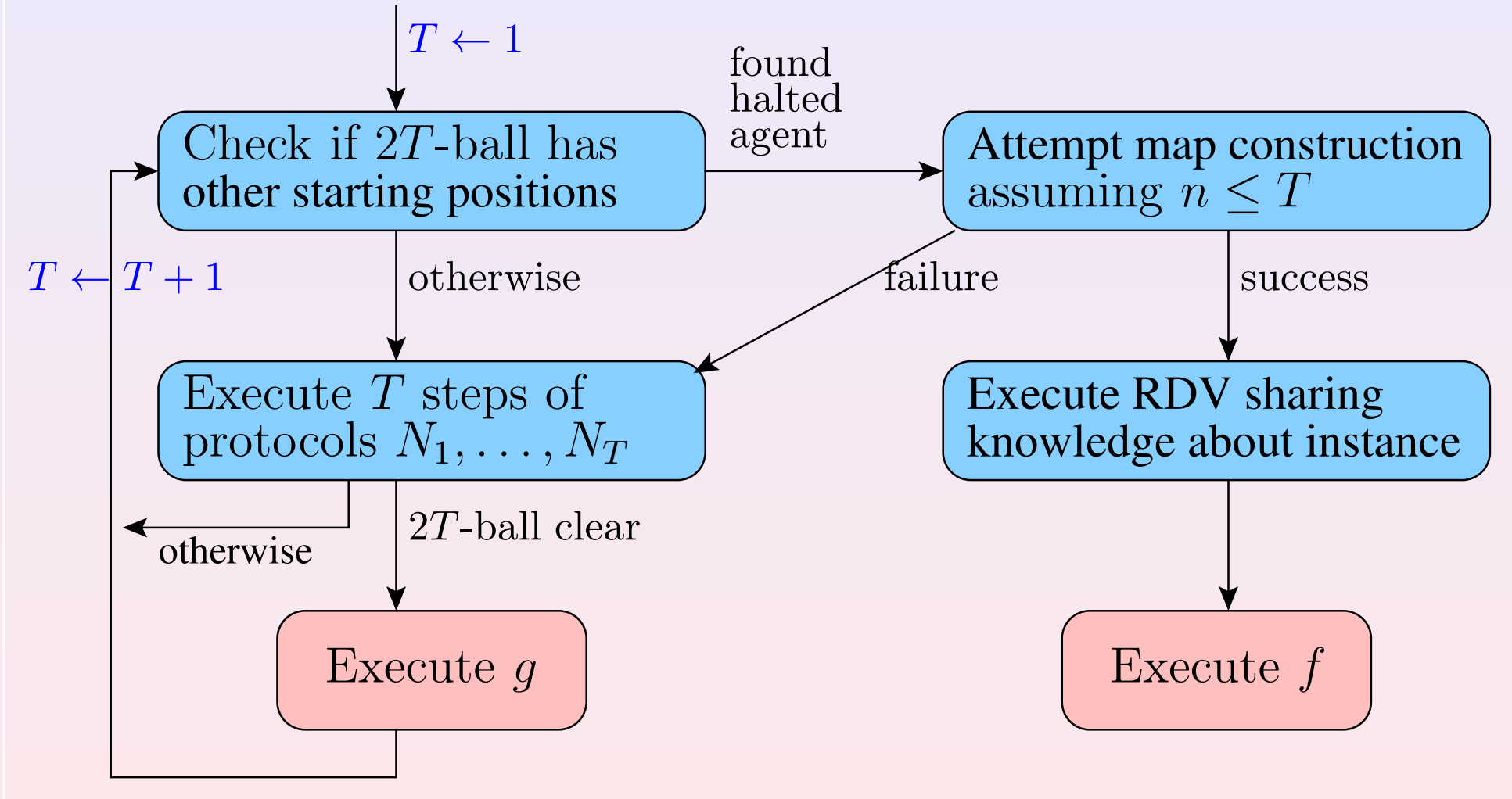
Meta-protocol

- 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.

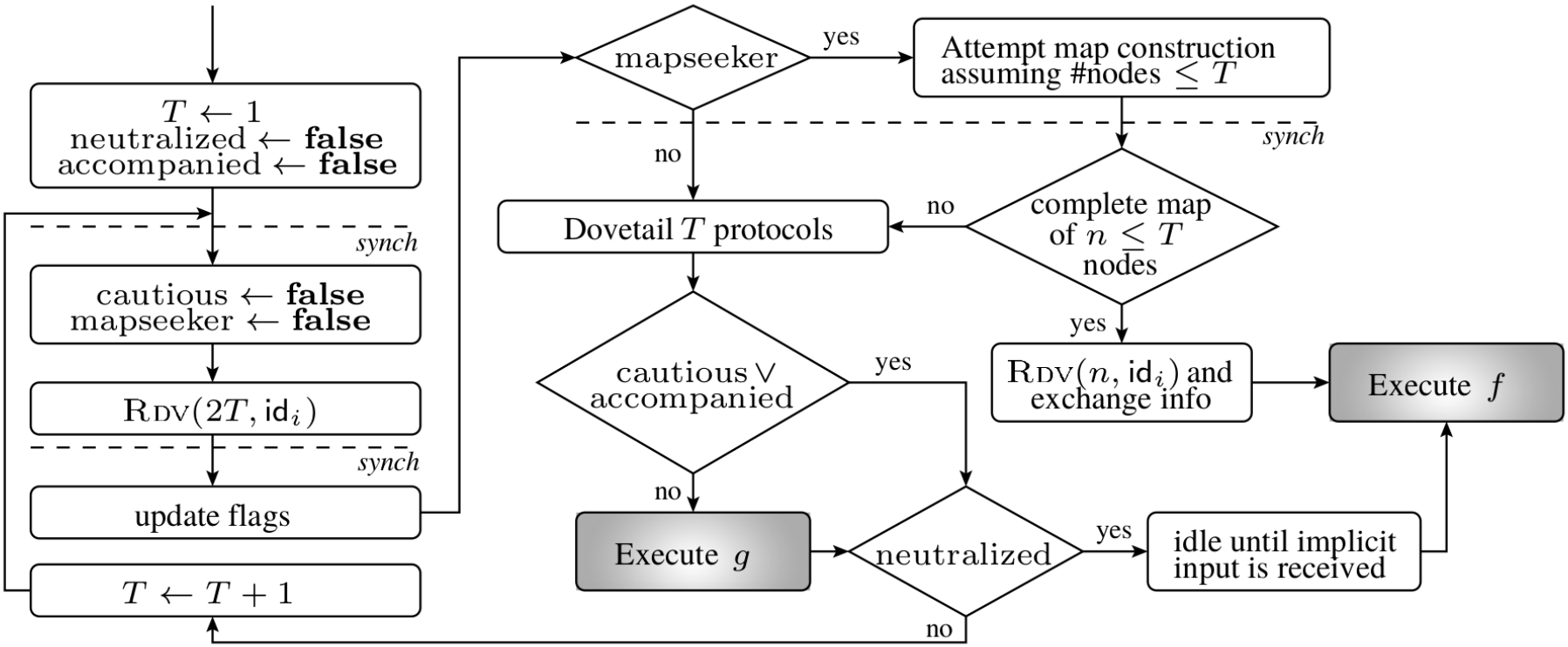
Parameters

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

Meta-protocol



Meta-protocol



Future research

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

Thank you!

Future research

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

Thank you!