Immediate Observation in Mediated Population Protocols

Tobias Prehn Myron Rotter

Technische Universität Berlin

Modelle und Theorie Verteilter Systeme

31 October 2019
@ Munich

Immediate Observation in Mediated Population Protocols

& several other things

Tobias Prehn Myron Rotter

Technische Universität Berlin

Modelle und Theorie Verteilter Systeme

31 October 2019

@ Munich

Emergence

Emergence

$$\text{system} \quad > \quad \sum \text{parts}$$

Emergence

system	$>$	$\sum$ parts
global	vs.	local

Emergence

$$\begin{array}{rcl} \text{system} & > & \sum \text{parts} \\ \text{global} & \text{vs.} & \text{local} \\ \text{top-down} & \text{vs.} & \text{bottom-up} \end{array}$$

Emergence

$$\begin{array}{rcl} \text{system} & > & \sum \text{parts} \\ \text{global} & \text{vs.} & \text{local} \\ \text{top-down} & \text{vs.} & \text{bottom-up} \end{array}$$

Emergence

system	$>$	$\sum$ parts
global	vs.	local
top-down	vs.	bottom-up

Javier Esparza @ Marktoberdorf 2015

Parameterized Verification of Crowds of Anonymous Processes

Emergence

 system $>$ $\sum$ parts 

global vs. local

top-down vs. bottom-up

Javier Esparza @ Marktoberdorf 2015

Parameterized Verification of Crowds of Anonymous Processes

Emergence

 system $>$ $\sum$ parts 

 global vs. local 

top-down vs. bottom-up

Javier Esparza @ Marktoberdorf 2015

Parameterized Verification of Crowds of Anonymous Processes

Emergence

system $>$ $\sum$ parts

global vs. local

top-down vs. bottom-up

Javier Esparza @ Marktoberdorf 2015

Parameterized Verification of Crowds of Anonymous Processes

Preliminaries

PP

Preliminaries

$$SEM \xrightarrow{\text{[D. Angluin et al. 2004 / 2007]}} \mathbf{PP}$$

Preliminaries

$$SEM \xrightarrow{\text{[D. Angluin et al. 2004 / 2007]}} \textbf{PP}$$

IOPP

Preliminaries

$$SEM \xrightarrow{\text{[D. Angluin et al. 2004 / 2007]}} \mathbf{PP}$$

$$\vee$$

$$COUNT_{*} \xrightarrow{\text{[D. Angluin et al. 2007]}} \mathbf{IOPP}$$

Preliminaries

MPP

SEM $\xrightarrow{\text{[D. Angluin et al. 2004 / 2007]}}$ **PP**

$\vee$

$COUNT_*$ $\xrightarrow{\text{[D. Angluin et al. 2007]}}$ **IOPP**

Preliminaries

$$MPS \xrightarrow{\text{[O. Michail et al. 2011]}} \mathbf{MPP}$$

$$\vee$$

$$SEM \xrightarrow{\text{[D. Angluin et al. 2004 / 2007]}} \mathbf{PP}$$

$$\vee$$

$$COUNT_* \xrightarrow{\text{[D. Angluin et al. 2007]}} \mathbf{IOPP}$$

Preliminaries

$MPS \xrightarrow{\text{[O. Michail et al. 2011]}}$ **MPP**

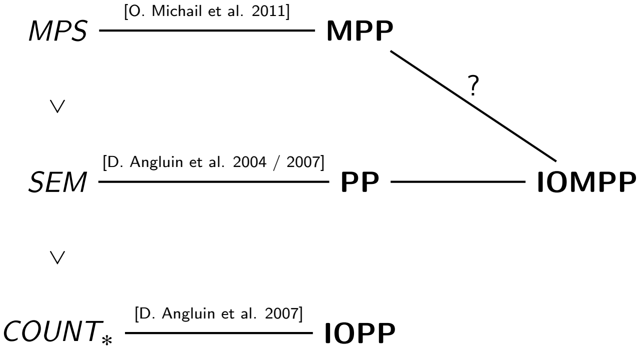
$\vee$

$SEM \xrightarrow{\text{[D. Angluin et al. 2004 / 2007]}}$ **PP** **IOMPP**

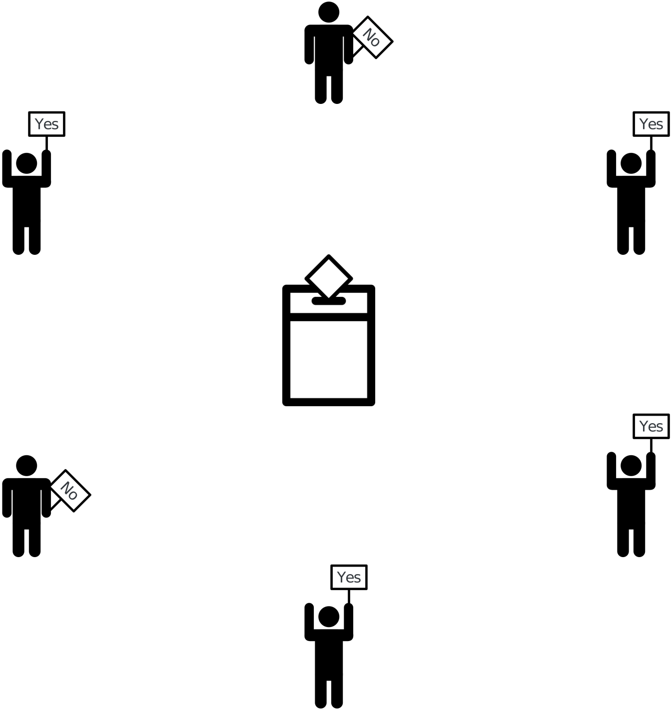
$\vee$

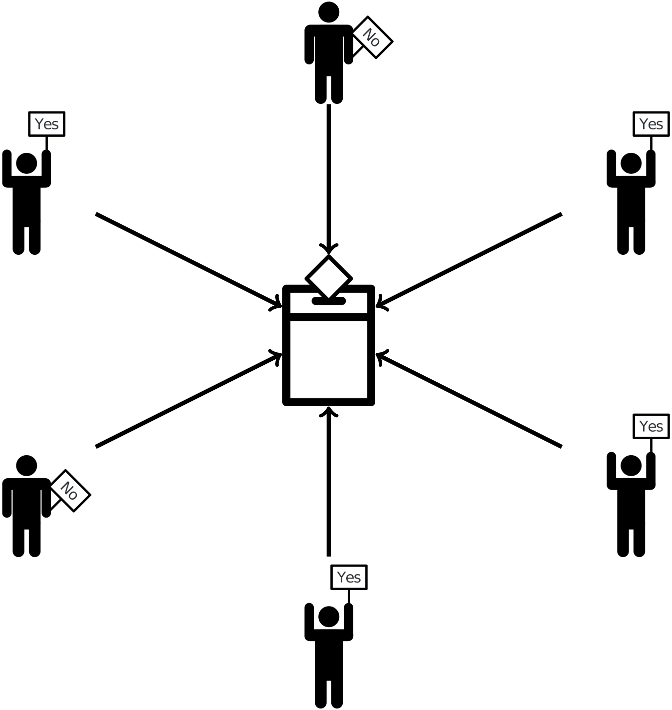
$COUNT_* \xrightarrow{\text{[D. Angluin et al. 2007]}}$ **IOPP**

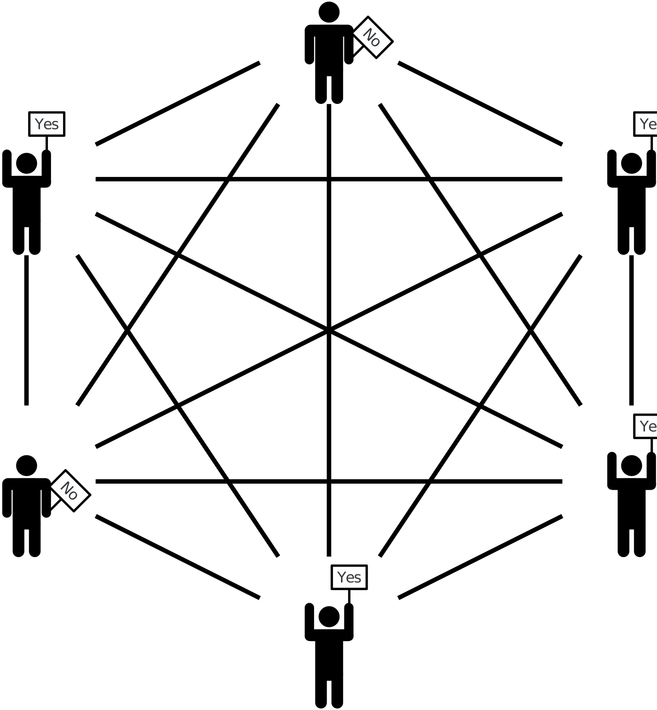
Preliminaries

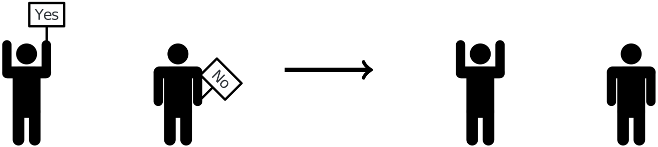


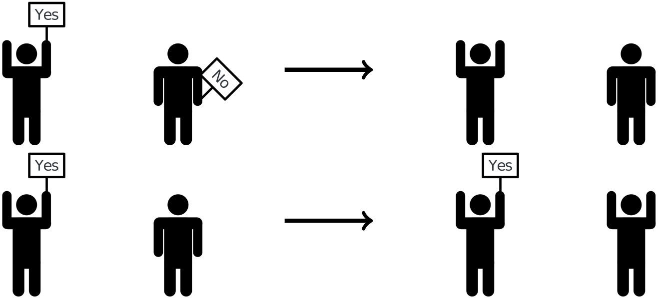
Berliner Morgenpost, pa

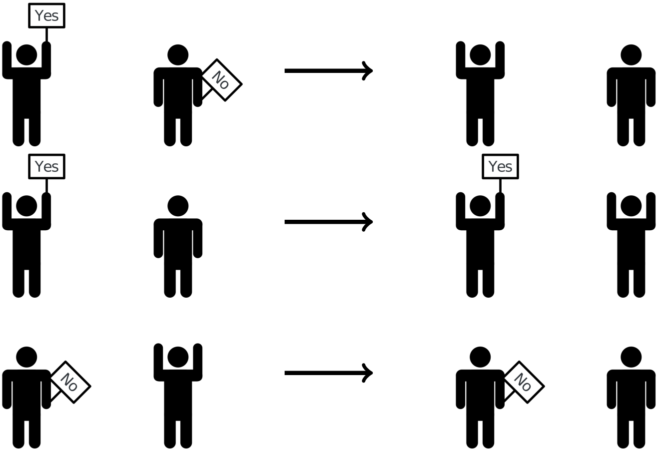

No
Yes
Yes
Yes
No
Yes

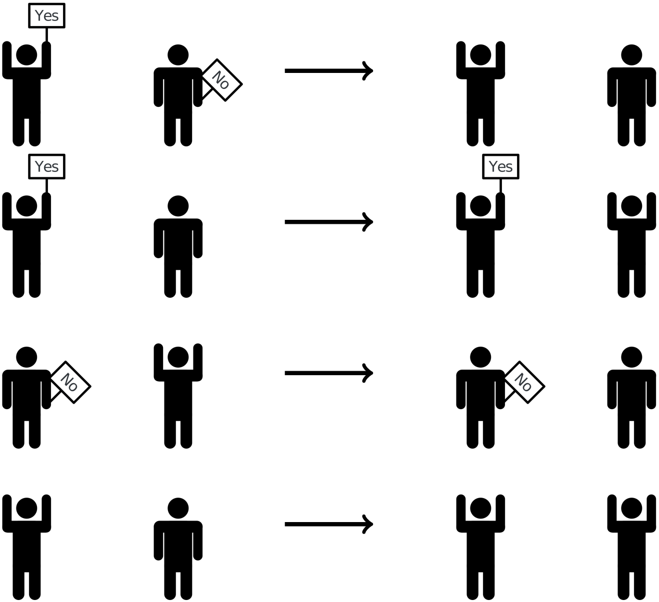

No
Yes
Yes
No
Yes
Yes


No
Yes
Yes
Yes
No
Yes

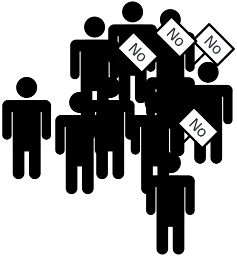

Yes
No


Yes
No
Yes
Yes

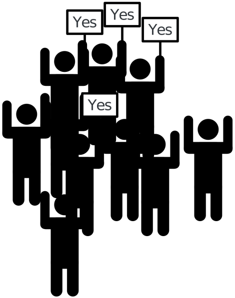

Yes
No
Yes
Yes
No
No



Yes
No
Yes
Yes
No
No



or



Population Protocols

Model designed to represent sensor networks consisting of
very limited mobile agents
with no control over their own movement.

Population Protocols

Model designed to represent sensor networks consisting of very limited mobile agents with no control over their own movement.

Features:

- finite state agents

Population Protocols

Model designed to represent sensor networks consisting of
very limited mobile agents
with no control over their own movement.

Features:

- finite state agents

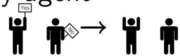
- uniformity - same program for every agent

Population Protocols

Model designed to represent sensor networks consisting of very limited mobile agents with no control over their own movement.

Features:

- finite state agents 

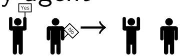
- uniformity - same program for every agent
- computation by direct interaction 

Population Protocols

Model designed to represent sensor networks consisting of
very limited mobile agents
with no control over their own movement.

Features:

- finite state agents 

- uniformity - same program for every agent
- computation by direct interaction 

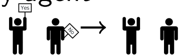
- distributed inputs and outputs

Population Protocols

Model designed to represent sensor networks consisting of
very limited mobile agents
with no control over their own movement.

Features:

- finite state agents 

- uniformity - same program for every agent
- computation by direct interaction 

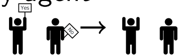
- distributed inputs and outputs
- convergence instead of termination

Population Protocols

Model designed to represent sensor networks consisting of
very limited mobile agents
with no control over their own movement.

Features:

- finite state agents 

- uniformity - same program for every agent
- computation by direct interaction 

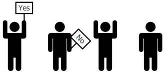
- distributed inputs and outputs
- convergence instead of termination
- strong fairness assumption
  (every reachable configuration will eventually be reached)

Population Protocols

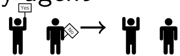
Model designed to represent sensor networks consisting of
very limited mobile agents
with no control over their own movement.

Features:

- finite state agents



- uniformity - same program for every agent
- computation by direct interaction



- distributed inputs and outputs
- convergence instead of termination
- strong fairness assumption
  (every reachable configuration will eventually be reached)
- anonymous

Model — PP [D. Angluin et al. 2004]

A protocol $P = (Q, \Sigma, I, O, \delta)$ is formally specified by

- Q, a finite set of agent states,

Model — PP [D. Angluin et al. 2004]

A protocol $P = (Q, \Sigma, I, O, \delta)$ is formally specified by

- Q, a finite set of agent states,
- Σ, a finite input alphabet,

Model — PP [D. Angluin et al. 2004]

A protocol $P = (Q, \Sigma, I, O, \delta)$ is formally specified by

- Q, a finite set of agent states,
- Σ, a finite input alphabet,
- I, an input mapping,

Model — PP [D. Angluin et al. 2004]

A protocol $P = (Q, \Sigma, I, O, \delta)$ is formally specified by

- Q, a finite set of agent states,
- Σ, a finite input alphabet,
- I, an input mapping,
- O, an output mapping, and

Model — PP [D. Angluin et al. 2004]

A protocol $P = (Q, \Sigma, I, O, \delta)$ is formally specified by

- Q, a finite set of agent states,
- Σ, a finite input alphabet,
- I, an input mapping,
- O, an output mapping, and
- $\delta \subseteq Q^4$, a transition relation.

Model — PP [D. Angluin et al. 2004]

A protocol $P = (Q, \Sigma, I, O, \delta)$ is formally specified by

- Q, a finite set of agent states,
- Σ, a finite input alphabet,
- I, an input mapping,
- O, an output mapping, and
- $\delta \subseteq Q^4$, a transition relation.

↳ each agent has stable output x $\Rightarrow$ global output is x

Model — PP [D. Angluin et al. 2004]

A protocol $P = (Q, \Sigma, I, O, \delta)$ is formally specified by

- Q, a finite set of agent states,
- Σ, a finite input alphabet,
- I, an input mapping,
- O, an output mapping, and
- $\delta \subseteq Q^4$, a transition relation.

↳ each agent has stable output x $\Rightarrow$ global output is x

↳ computational power:
SEM (modulo, threshold, and boolean combinations)

Immediate Observation — IOPP

- only interactions where the first agent does not change

Immediate Observation — IOPP

- only interactions where the first agent does not change
- i.e. $(p, q) \to (p', q')$ with $p = p'$

Immediate Observation — IOPP

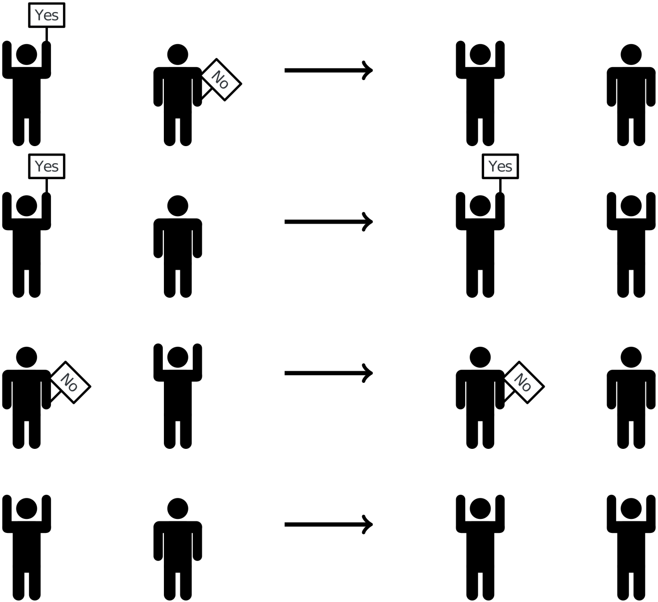
- only interactions where the first agent does not change
- i.e. $(p, q) \rightarrow (p', q')$ with $p = p'$
- fewer synchronization issues in implementation

Immediate Observation — IOPP

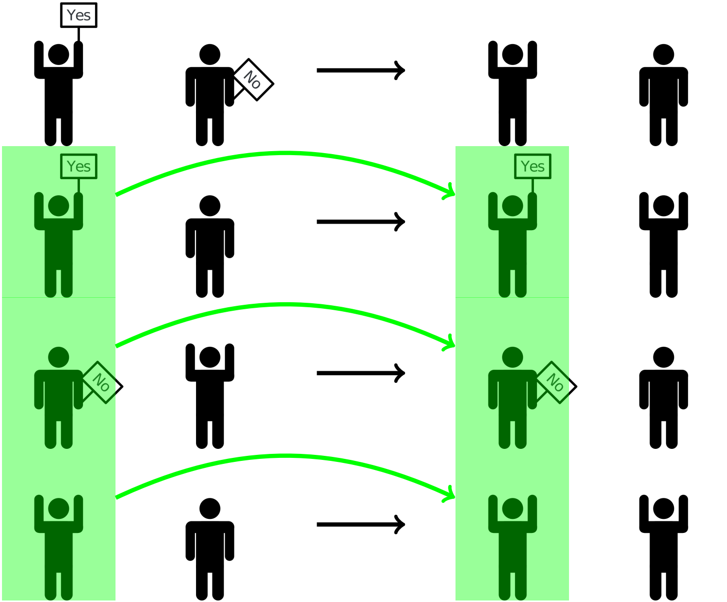
- only interactions where the first agent does not change
- i.e. $(p, q) \to (p', q')$ with $p = p'$
- fewer synchronization issues in implementation

↳ computational power:
$COUNT_*$ (count number of initial states)

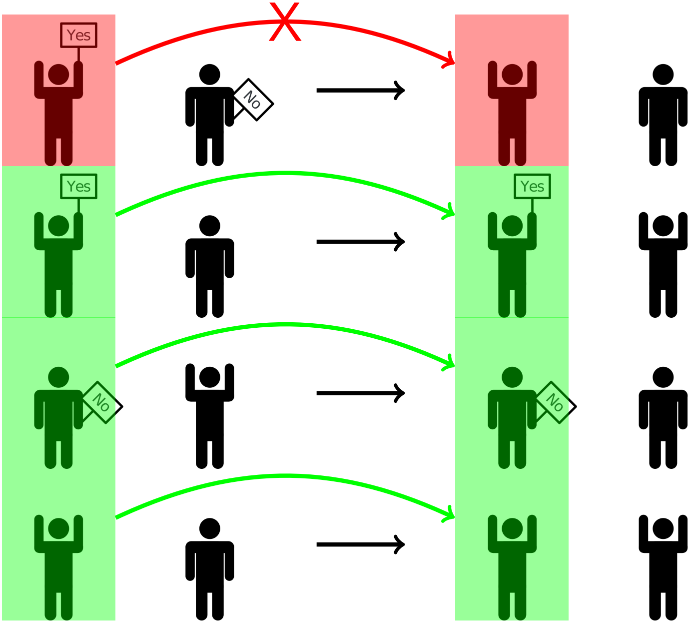
Immediate Observation — IOPP



Immediate Observation — IOPP



Immediate Observation — IOPP



Mediated - MPP

- information may be stored in edges

Mediated - MPP

- information may be stored in edges
- finite set S of edge states

Mediated - MPP

- information may be stored in edges
- finite set S of edge states
- initialized identically to $s_0 \in S$

Mediated - MPP

- information may be stored in edges
- finite set S of edge states
- initialized identically to $s_0 \in S$
- more information on interaction partners

Mediated - MPP

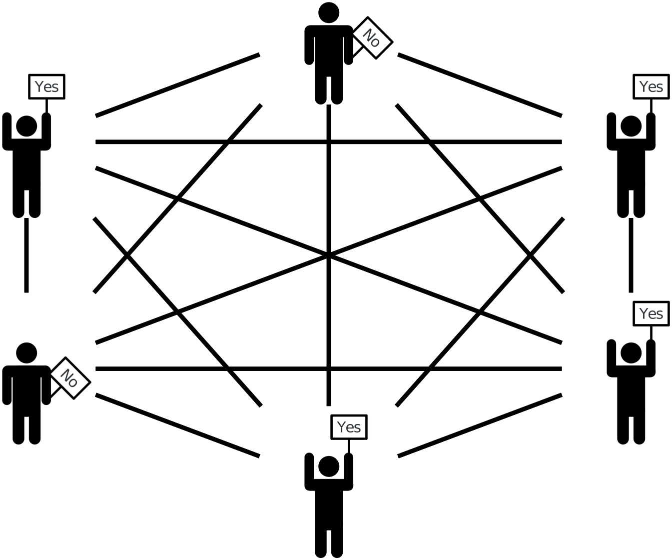
- information may be stored in edges
- finite set S of edge states
- initialized identically to $s_0 \in S$
- more information on interaction partners
- still anonymous

Mediated - MPP

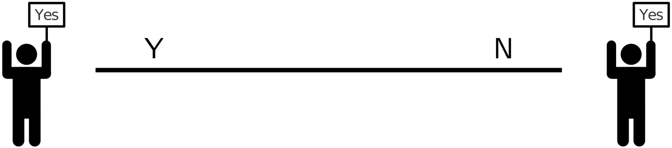
- information may be stored in edges
- finite set S of edge states
- initialized identically to $s_0 \in S$
- more information on interaction partners
- still anonymous

↳ computational power:
MPS (symmetric predicates in NSPACE(n^2))

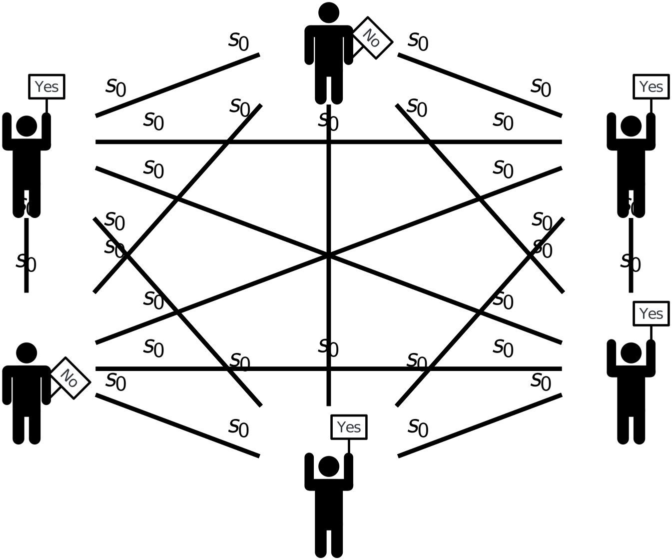
Mediated — MPP



Mediated — MPP



Mediated — MPP



Model — IOMPP

An immediate observation mediated population protocol P is a 7-tuple $P = (Q, \Sigma, \qquad I, O, \delta)$

- Q, a finite set of agent states
- Σ, a finite input alphabet

- $I : \Sigma \to Q$, an input function
- $O : Q \to \{0, 1\}$, an output function
- $\delta : (Q \times S)^2 \to (Q \times S)^2$, a transition function

Model — IOMPP

An immediate observation mediated population protocol P is a 7-tuple $P = (Q, \Sigma, S, s_0, I, O, \delta)$

- Q, a finite set of agent states
- Σ, a finite input alphabet
- S, a finite set of edge states
- s_0, an initial edge state
- $I : \Sigma \to Q$, an input function
- $O : Q \to \{0, 1\}$, an output function
- $\delta : (Q \times S)^2 \to (Q \times S)^2$, a transition function

Model — IOMPP

An immediate observation mediated population protocol P is a 7-tuple $P = (Q, \Sigma, S, s_0, I, O, \delta)$

- Q, a finite set of agent states
- Σ, a finite input alphabet
- S, a finite set of edge states
- s_0, an initial edge state
- $I : \Sigma \to Q$, an input function
- $O : Q \to \{0, 1\}$, an output function
- $\delta : (Q \times S)^2 \to (Q \times S)^2$, a transition function
  $\delta(p, s, q, r) = (p, s, q', r')$

Computational Power Proof

literature: boolean combination of modulo and threshold predicates

Computational Power Proof

literature: boolean combination of modulo and threshold predicates
here: find “good” simulation for each population protocol

Computational Power Proof

literature: boolean combination of modulo and threshold predicates
here: find “good” simulation for each population protocol

D. Gorla 2010 — *Towards a unified approach to encodability and separation results for process calculi*
↳ criteria for encodings to be good

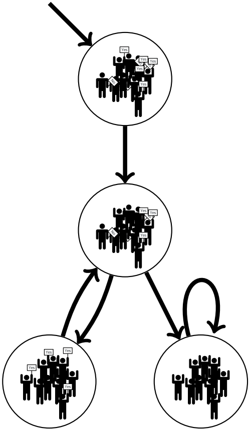
Computational Power Proof

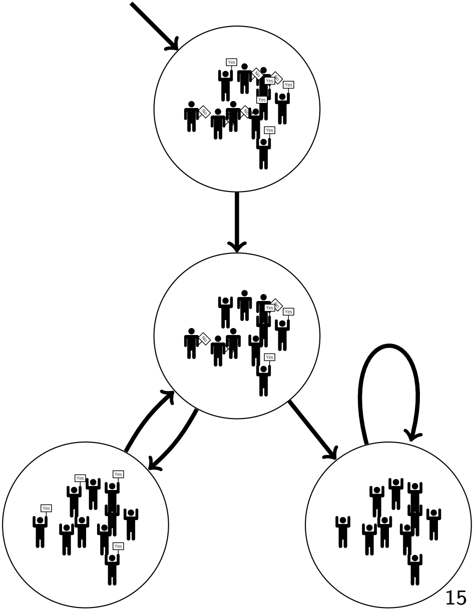
literature: boolean combination of modulo and threshold predicates
here: find “good” simulation for each population protocol

D. Gorla 2010 — *Towards a unified approach to encodability and separation results for process calculi*
↳ criteria for encodings to be good

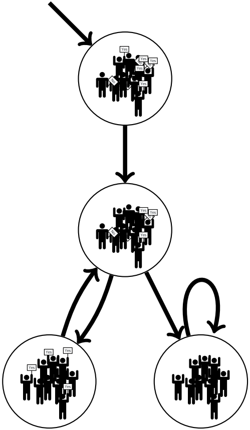
Original configurations can be translated to configurations of the “good” simulation such that the following criteria hold.

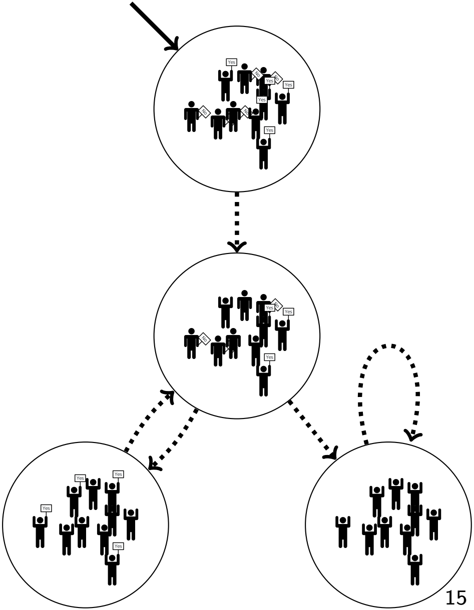
Translation Criteria



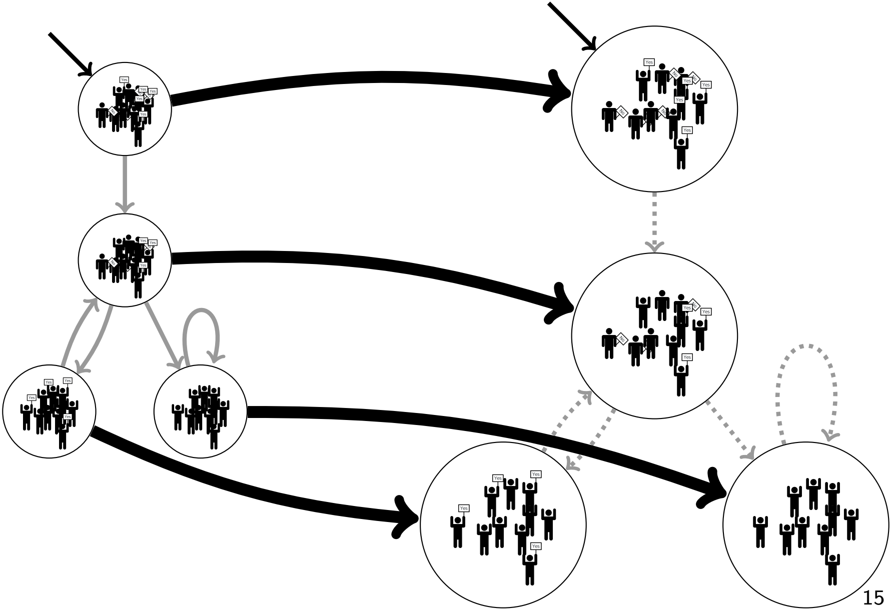


Translation Criteria

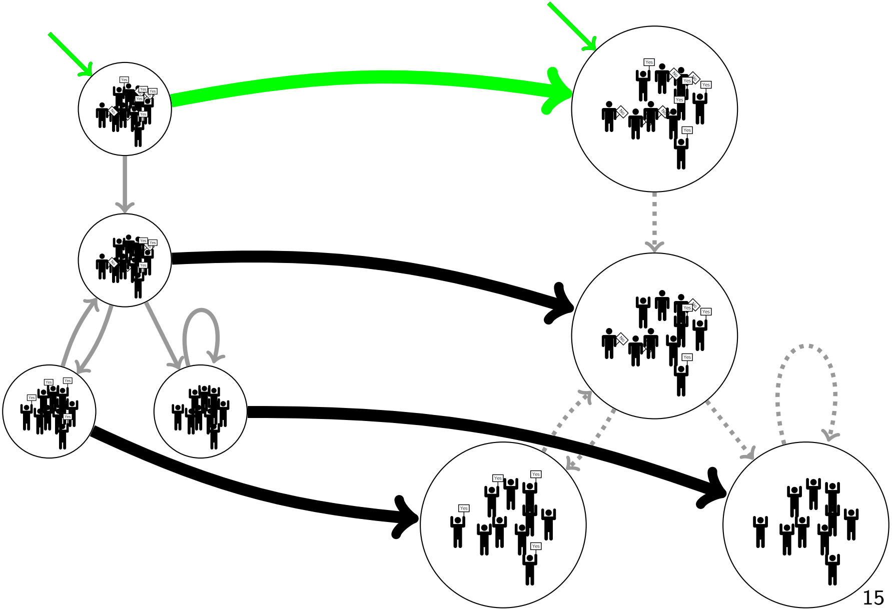




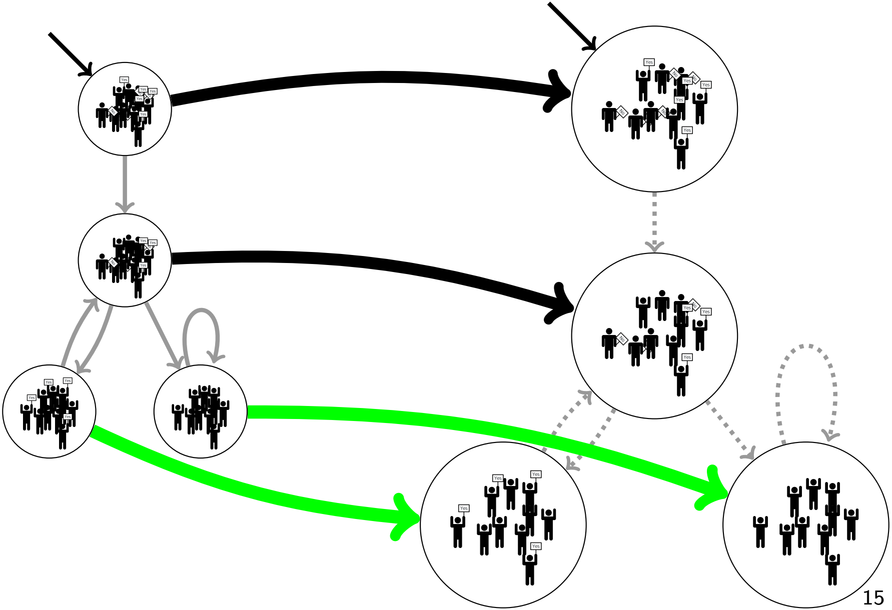
Translation Criteria



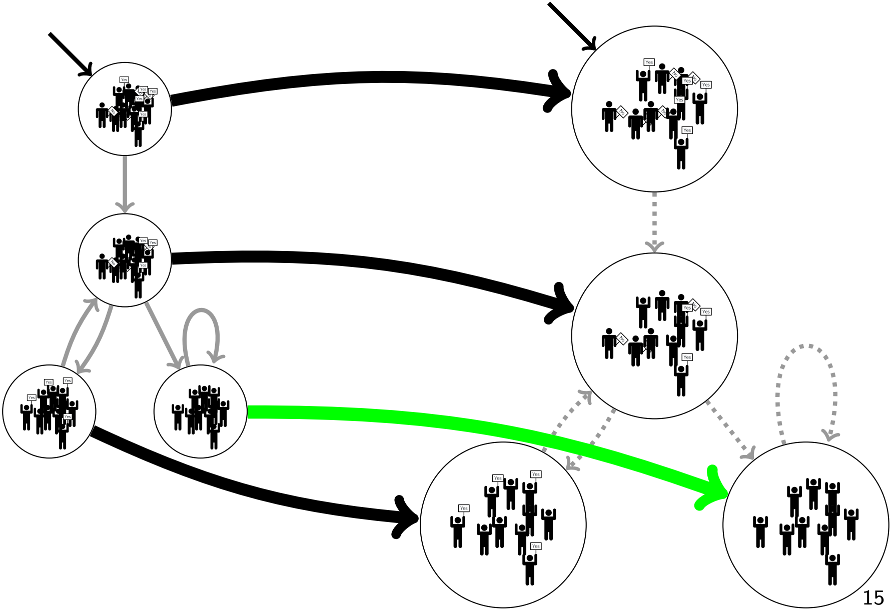
Translation Criteria - Input Correspondence



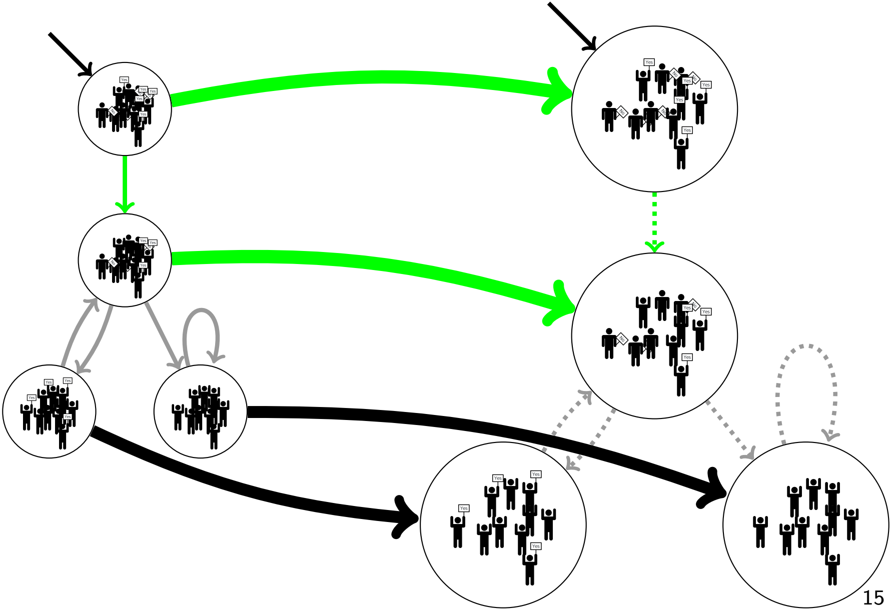
Translation Criteria - Output Correspondence



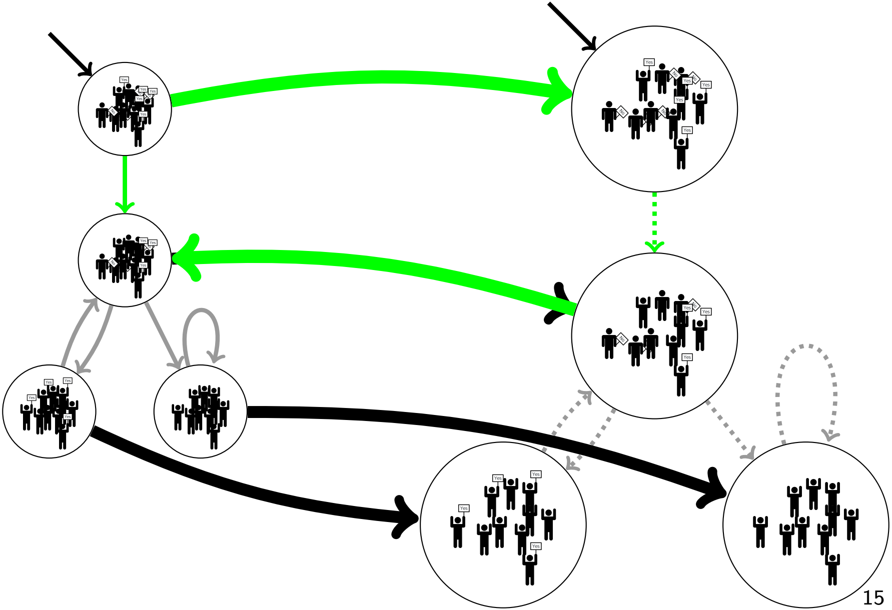
Translation Criteria - Output Stability Preservation



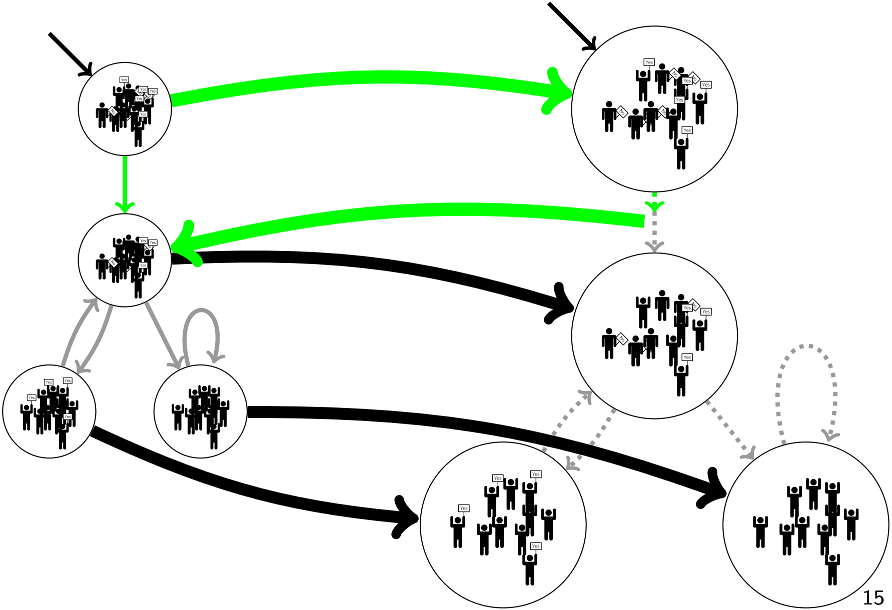
Translation Criteria - Operational Correspondence [Gorla]



Translation Criteria - Operational Correspondence [Gorla]



Translation Criteria - Operational Correspondence [Gorla]

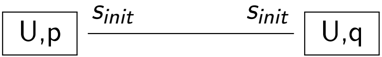


Simulation of PP in IOMPP

For each $(p, q) \to (p', q')$ in δ we get 4+1 steps in δ' as follows:

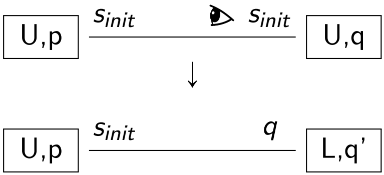
Simulation of PP in IOMPP

For each $(p, q) \to (p', q')$ in δ we get 4+1 steps in δ' as follows:



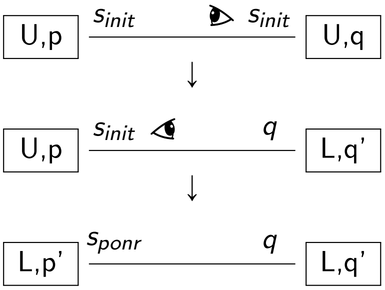
Simulation of PP in IOMPP

For each $(p, q) \to (p', q')$ in δ we get 4+1 steps in δ' as follows:



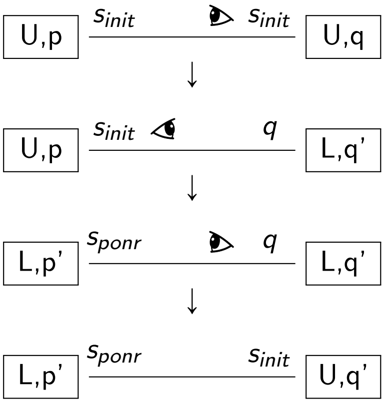
Simulation of PP in IOMPP

For each $(p, q) \rightarrow (p', q')$ in δ we get 4+1 steps in δ' as follows:



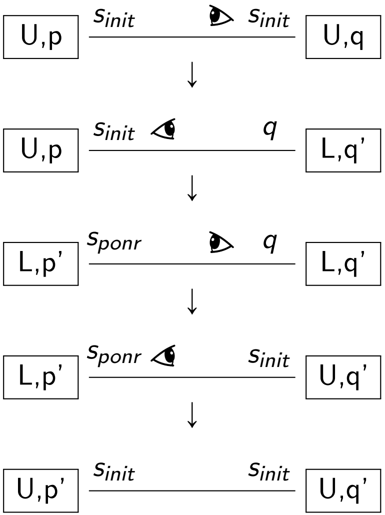
Simulation of PP in IOMPP

For each $(p, q) \to (p', q')$ in δ we get 4+1 steps in δ' as follows:



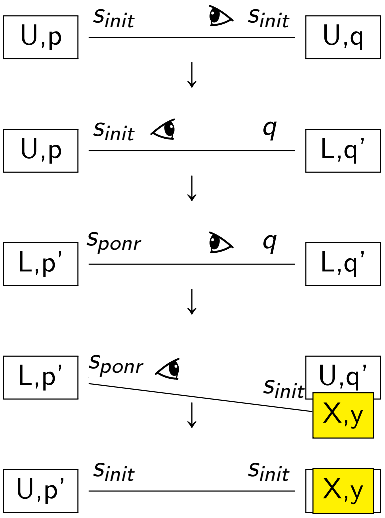
Simulation of PP in IOMPP

For each $(p, q) \to (p', q')$ in δ we get 4+1 steps in δ' as follows:



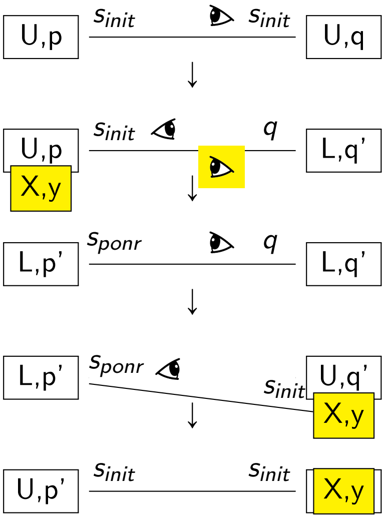
Simulation of PP in IOMPP

For each $(p, q) \to (p', q')$ in δ we get 4+1 steps in δ' as follows:



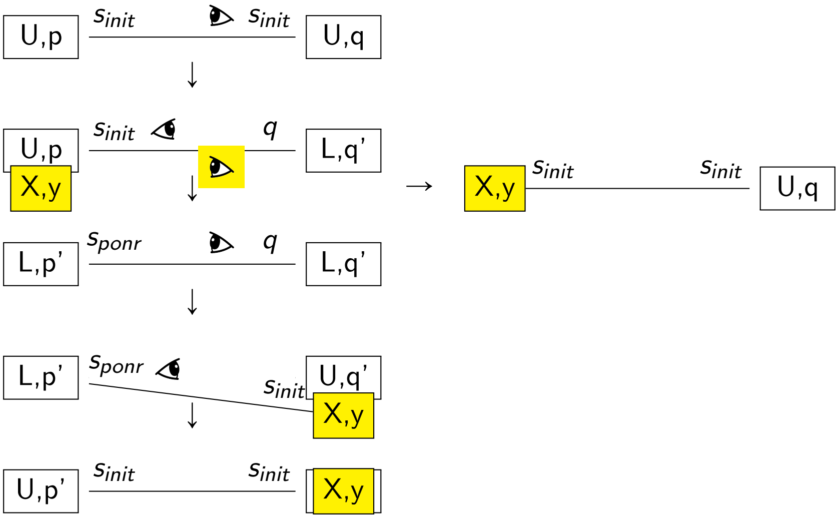
Simulation of PP in IOMPP

For each $(p, q) \to (p', q')$ in δ we get 4+1 steps in δ' as follows:



Simulation of PP in IOMPP

For each $(p, q) \to (p', q')$ in δ we get 4+1 steps in δ' as follows:



Benefits

several *possibly* desirable attributes of an algorithm carry over, e.g.:

- livelock freedom

Benefits

several *possibly* desirable attributes of an algorithm carry over, e.g.:

- livelock freedom
- supported communication structures

Benefits

several *possibly* desirable attributes of an algorithm carry over, e.g.:

- livelock freedom
- supported communication structures
- existence of key configurations / transitions

Benefits

several *possibly* desirable attributes of an algorithm carry over, e.g.:

- livelock freedom
- supported communication structures
- existence of key configurations / transitions
- (failure resistance)

Future Work I — IOMPP

- criteria for a “good” simulation

Future Work I — IOMPP

- criteria for a “good” simulation
- simulation for MPP in IOMPP

Future Work I — IOMPP

- criteria for a "good" simulation
- simulation for MPP in IOMPP
  $\Rightarrow$ IO does not weaken MPP

Future Work I — IOMPP

- criteria for a "good" simulation
- simulation for MPP in IOMPP
  $\Rightarrow$ IO does not weaken MPP
- study preserved attributes

Future Work I — IOMPP

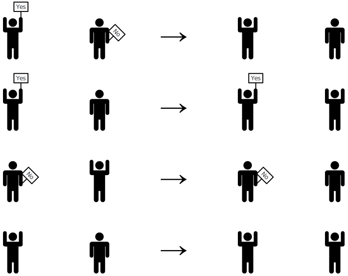
- criteria for a "good" simulation
- simulation for MPP in IOMPP
  $\Rightarrow$ IO does not weaken MPP
- study preserved attributes
- study PP with no calculated predicate

Future Work II — Well-Specified Transitions

Can we (sometimes) decide that a PP is well-specified by looking at the transitions?

or: Can we define designing guidlines for transitions that will, if followed, guarantee a well-specified PP?

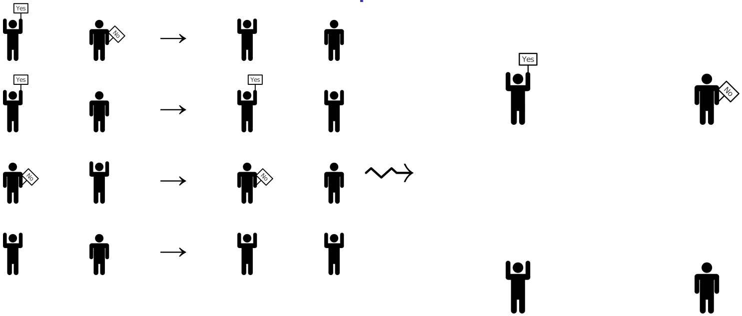
Future Work II — Well-Specified Transitions



Can we (sometimes) decide that a PP is well-specified
by looking at the transitions?

or: Can we define designing guidlines for transitions that will, if followed, guarantee a well-specified PP?

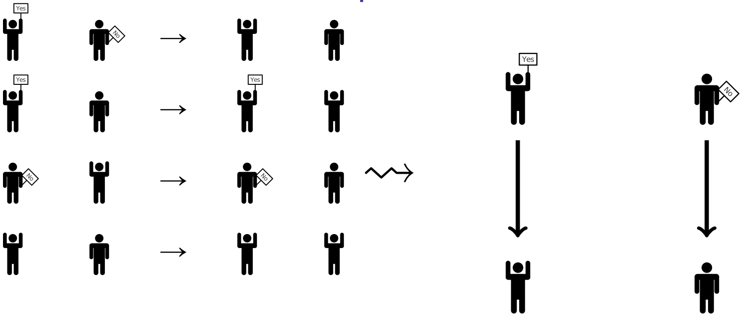
Future Work II — Well-Specified Transitions



Can we (sometimes) decide that a PP is well-specified
by looking at the transitions?

or: Can we define designing guidlines for transitions that will, if followed, guarantee a well-specified PP?

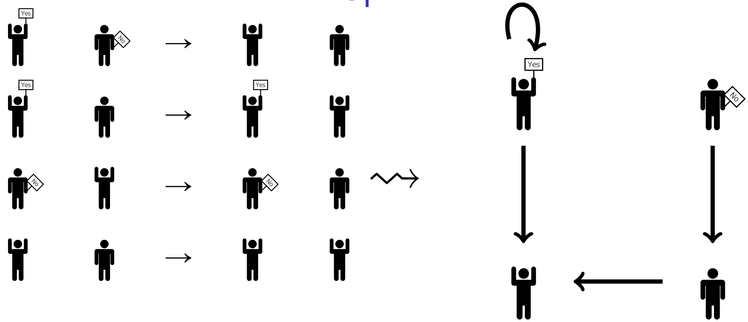
Future Work II — Well-Specified Transitions



Can we (sometimes) decide that a PP is well-specified
by looking at the transitions?

or: Can we define designing guidlines for transitions that will, if followed, guarantee a well-specified PP?

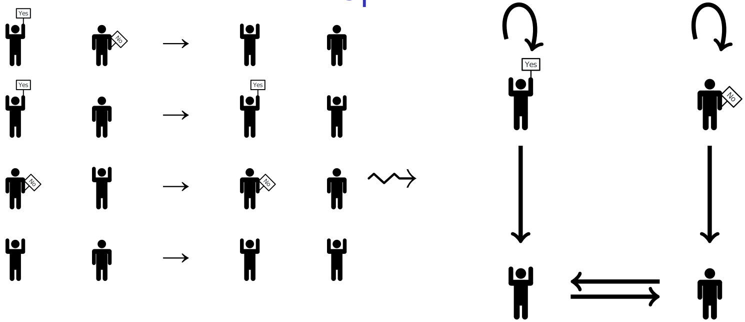
Future Work II — Well-Specified Transitions



Can we (sometimes) decide that a PP is well-specified by looking at the transitions?

or: Can we define designing guidlines for transitions that will, if followed, guarantee a well-specified PP?

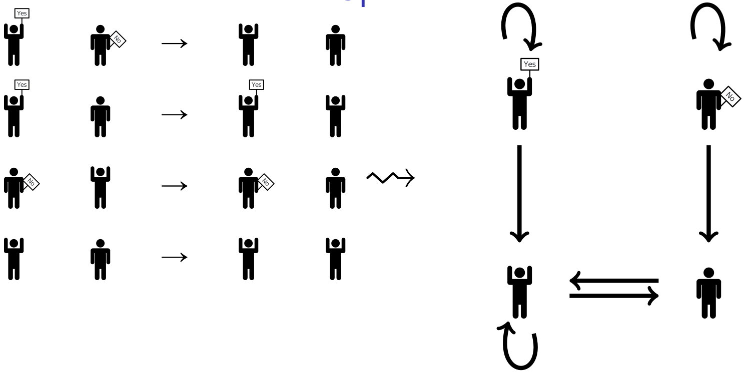
Future Work II — Well-Specified Transitions



Can we (sometimes) decide that a PP is well-specified by looking at the transitions?

or: Can we define designing guidlines for transitions that will, if followed, guarantee a well-specified PP?

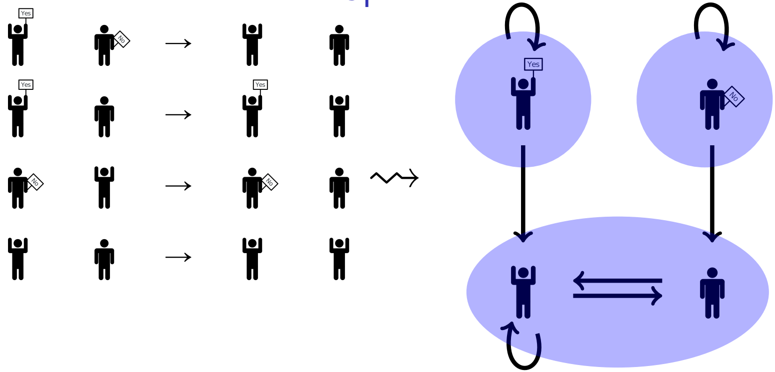
Future Work II — Well-Specified Transitions



Can we (sometimes) decide that a PP is well-specified by looking at the transitions?

or: Can we define designing guidlines for transitions that will, if followed, guarantee a well-specified PP?

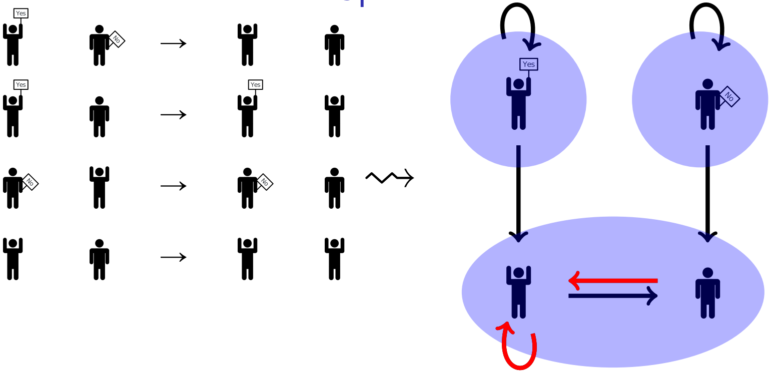
Future Work II — Well-Specified Transitions



Can we (sometimes) decide that a PP is well-specified by looking at the transitions?

or: Can we define designing guidlines for transitions that will, if followed, guarantee a well-specified PP?

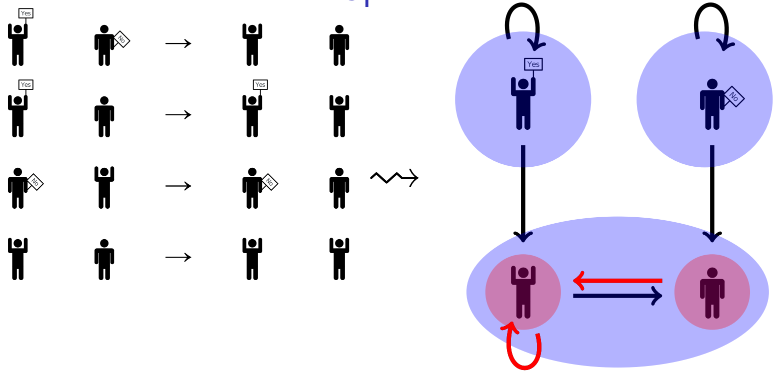
Future Work II — Well-Specified Transitions



Can we (sometimes) decide that a PP is well-specified by looking at the transitions?

or: Can we define designing guidlines for transitions that will, if followed, guarantee a well-specified PP?

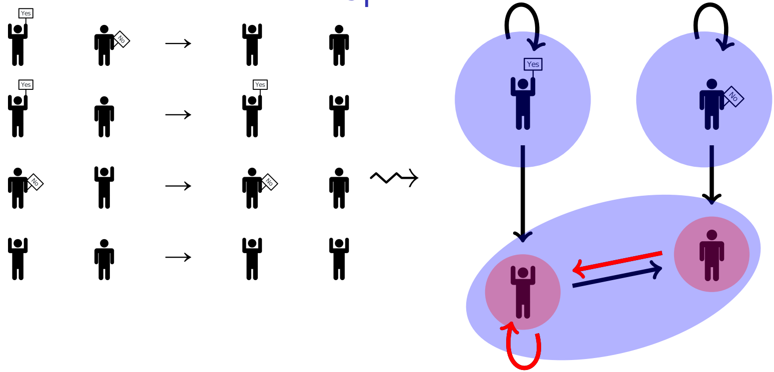
Future Work II — Well-Specified Transitions



Can we (sometimes) decide that a PP is well-specified
by looking at the transitions?

or: Can we define designing guidlines for transitions that will, if followed, guarantee a well-specified PP?

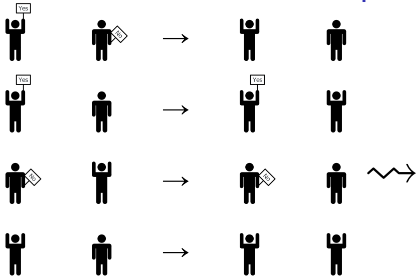
Future Work II — Well-Specified Transitions

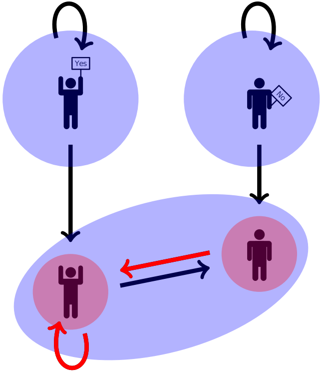


Can we (sometimes) decide that a PP is well-specified by looking at the transitions?

or: Can we define designing guidlines for transitions that will, if followed, guarantee a well-specified PP?

Future Work II — Well-Specified Transitions





Transitions may be:

Identity:

$(p, q) \rightarrow_{\delta_P} (p, q)$ where $O(p) = O(q)$

Layer-Weakening:

$(p, q) \rightarrow_{\delta_P} (p', q')$ where $q' \ll q$ and $p' \ll= p$

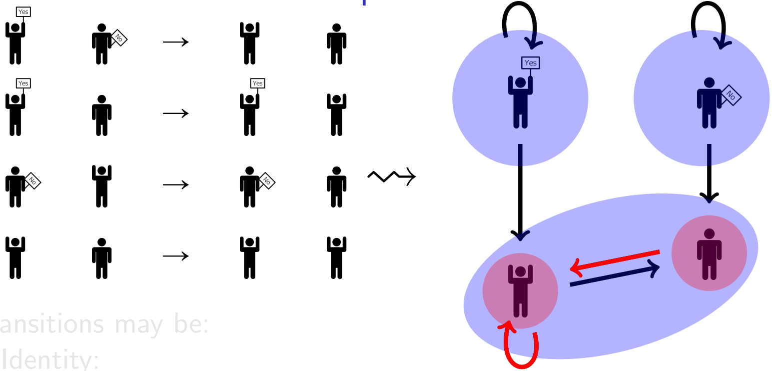
Sub-Layer-Weakening:

$(p, q) \rightarrow_{\delta_P} (p', q')$ where $p == q$ and $q' < q$ and $p' <= p$

Higher-Layer-Id-Convincing:

$(p, q) \rightarrow_{\delta_P} (p, q')$ where $q \ll p$ and $O(p) = O(q')$

Future Work II — Well-Specified Transitions



Transitions may be:

- Identity:
  - $(p, q) \rightarrow_{\delta_P} (p, q)$ where $O(p) = O(q)$
- Layer-Weakening:
  - $(p, q) \rightarrow_{\delta_P} (p',$
- Sub-Layer-Weake
  - $(p, q) \rightarrow_{\delta_P} (p', q')$ where $p == q$ and $q' < q$ and $p' <= p$
- Higher-Layer-Id-Convincing:
  - $(p, q) \rightarrow_{\delta_P} (p, q')$ where $q \ll p$ and $O(p) = O(q')$

$\Rightarrow$ “Termination” / Convergence ?
$\not\Rightarrow$ predetermined output

Future Work II — Well-Specified Transitions

From Blondin, Esparza, Kučera - CONCUR 2018:

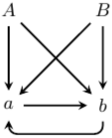
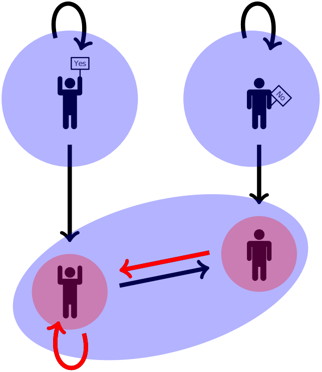


Figure 2 Transformation graphs of Example 8.



Transitions may be:

- Identity:
  $(p, q) \rightarrow_{\delta_P} (p, q)$ where $O(p) = O(q)$
- Layer-Weakening:
  $(p, q) \rightarrow_{\delta_P} (p'$, …
- Sub-Layer-Weake…
  $(p, q) \rightarrow_{\delta_P} (p', q')$ where $p == q$ and $q' < q$ and $p' <= p$
- Higher-Layer-Id-Convincing:
  $(p, q) \rightarrow_{\delta_P} (p, q')$ where $q \ll p$ and $O(p) = O(q')$

$\Rightarrow$ "Termination" / Convergence ?
$\not\Rightarrow$ predetermined output

Future Work III — Crash Failure Resistance

Crash and transient failures via split-simulate-majority vote

Delporte-Gallet et al. (2006) - *When birds die: Making population protocols fault-tolerant.*

Future Work III — Crash Failure Resistance

Crash and transient failures via split-simulate-majority vote

Delporte-Gallet et al. (2006) - *When birds die: Making population protocols fault-tolerant.*

simple threshold k (with c crashes):

$(\#A \geqslant k)$

x	y	$\to$	x+y	0
k	y	$\to$	k	k

Future Work III — Crash Failure Resistance

Crash and transient failures via split-simulate-majority vote

Delporte-Gallet et al. (2006) - *When birds die: Making population protocols fault-tolerant.*

simple threshold k (with c crashes):

$(\#A \geqslant k)$

$$\begin{array}{cccccc} x & y & \rightarrow & x+y & 0 \\ k & y & \rightarrow & k & k \end{array} \quad \text{vs.} \quad \begin{array}{cccccc} x & x & \rightarrow & x & x+1 \\ k & y & \rightarrow & k & k \end{array}$$

Future Work III — Crash Failure Resistance

Crash and transient failures via split-simulate-majority vote

Delporte-Gallet et al. (2006) - *When birds die: Making population protocols fault-tolerant.*

simple threshold k (with c crashes):

$(\#A \geqslant k)$

x	y	→	x+y	0
k	y	→	k	k

vs.

x	x	→	x	x+1
k	y	→	k	k

$\#A \geqslant k + c$

Future Work III — Crash Failure Resistance

Crash and transient failures via split-simulate-majority vote

Delporte-Gallet et al. (2006) - *When birds die: Making population protocols fault-tolerant.*

simple threshold k (with c crashes):

$(\#A \geqslant k)$

x	y	→	x+y	0
k	y	→	k	k

vs.

x	x	→	x	x+1
k	y	→	k	k

$\#A \geqslant k + c \cdot \quad k$

$\#A \geqslant k + c$

Future Work III — Crash Failure Resistance

Crash and transient failures via split-simulate-majority vote

Delporte-Gallet et al. (2006) - *When birds die: Making population protocols fault-tolerant.*

simple threshold k (with c crashes):

$(\#A \geqslant k)$

x	y	→	x+y	0			x	x	→	x	x+1
k	y	→	k	k	vs.		k	y	→	k	k

$\#A \geqslant k + c \cdot 2 \cdot (k-1)$ $\#A \geqslant k + c$

Future Work

IOMPP

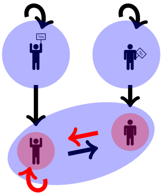
- simulation for MPP in IOMPP

  $\Rightarrow$ IO does not weaken MPP

Translation Criteria

- criteria for a "good" simulation
- study preserved attributes
- study PP with no predicate

Well-Specified Transitions



Transitions may be:

Identity

Layer-Weakening

Sub-Layer-Weakening

Higher-Layer-Id-Convincing

Crash Failure Resistance

- find resistant protocols
- other failures
- other failure handling strategies