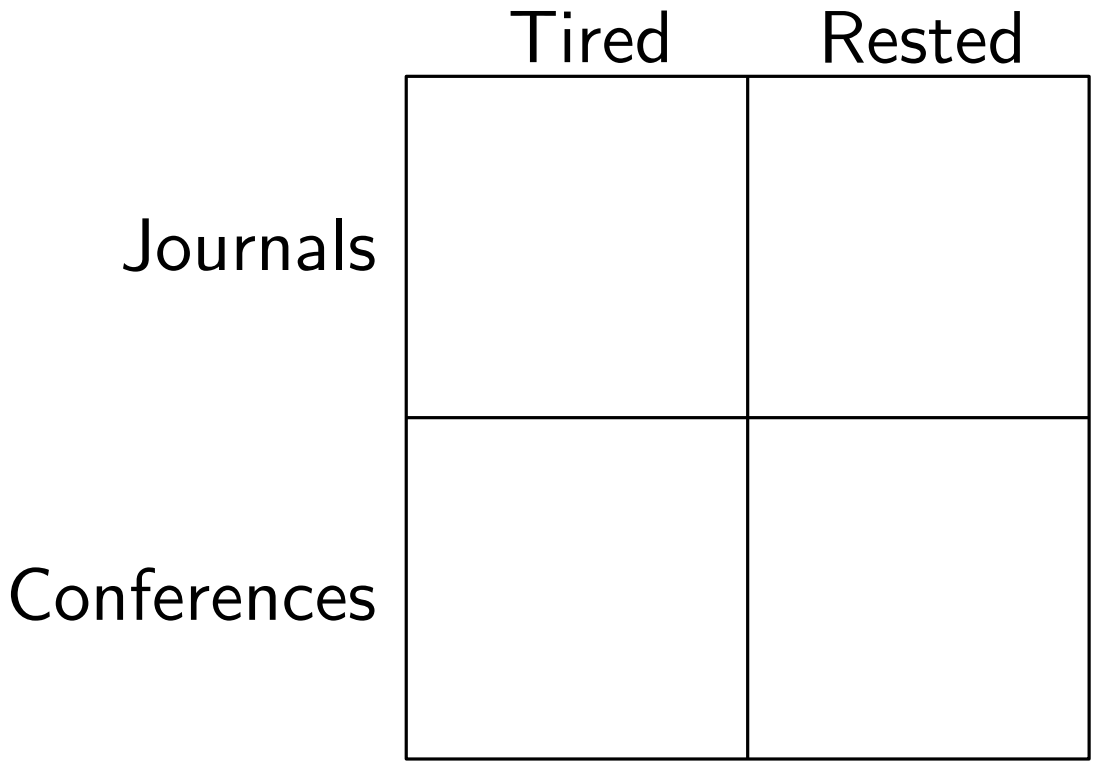
Population Protocols and Predicates

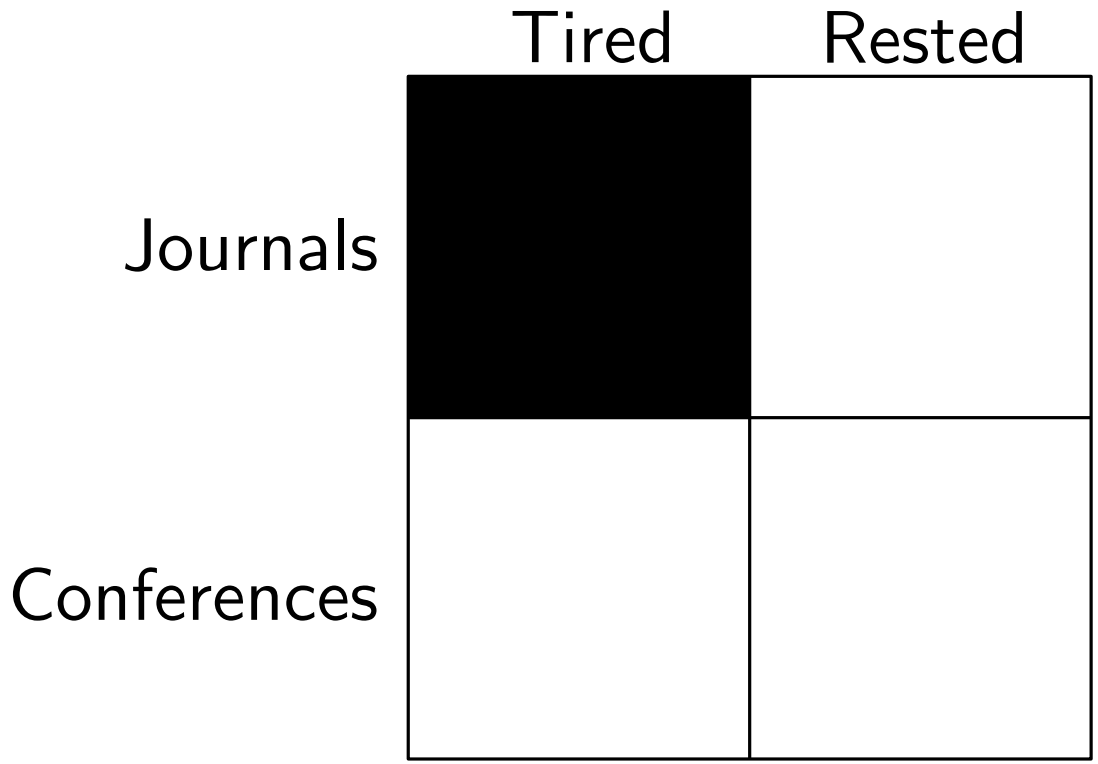
Pierre Ganty IMDEA Software Institute



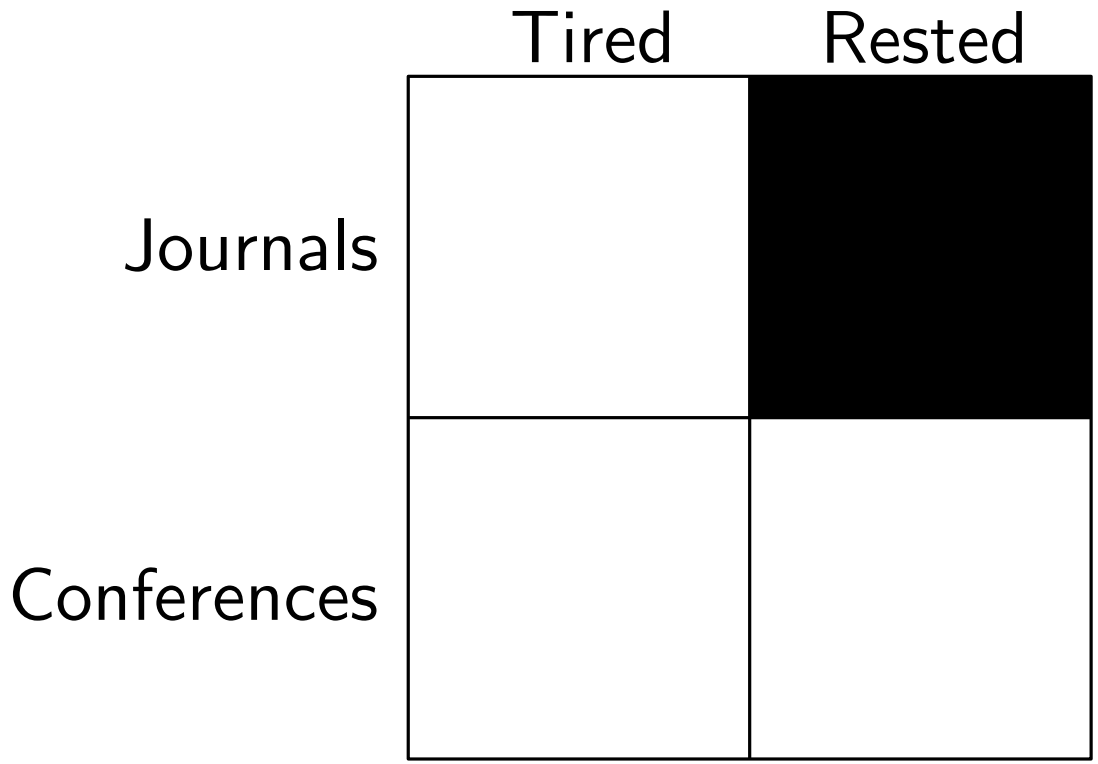
The computer science debate: Journals vs Conferences



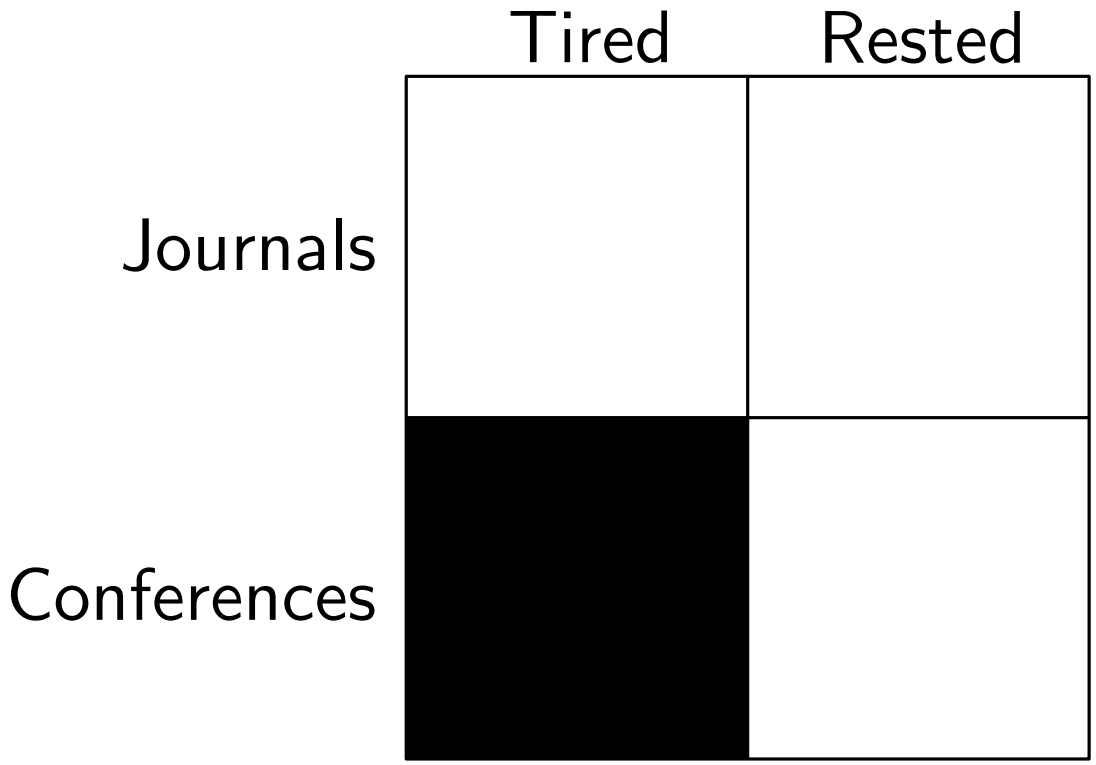
The computer science debate: Journals vs Conferences



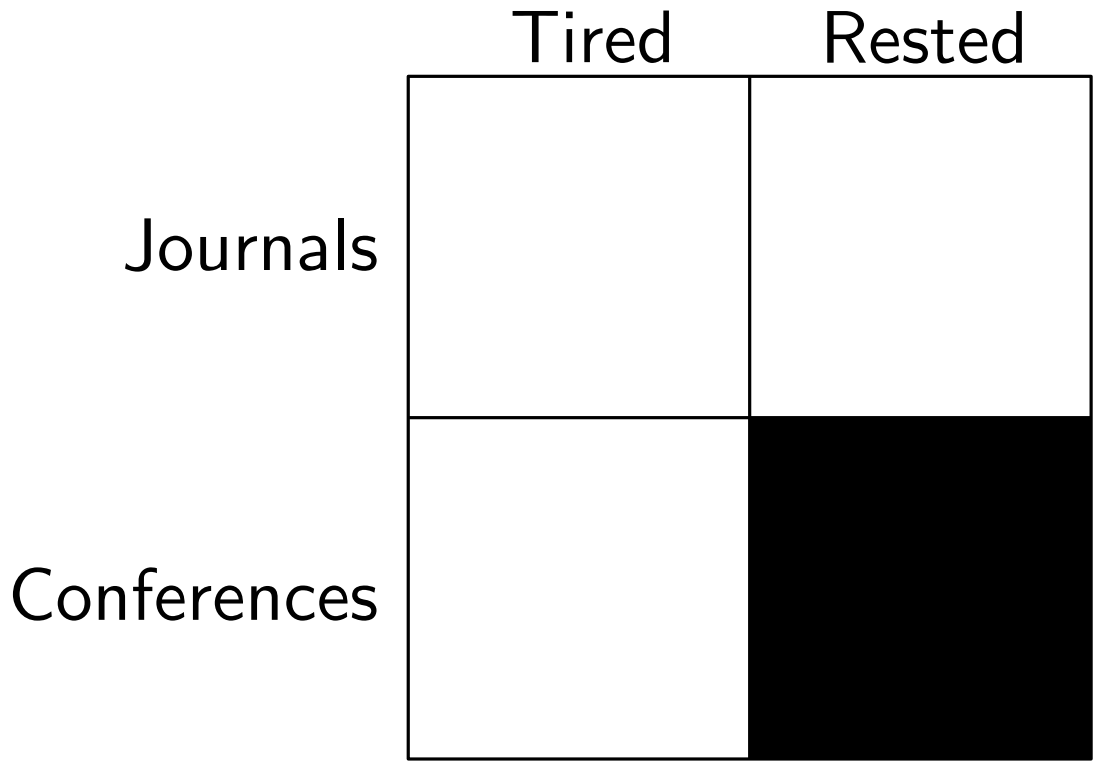
The computer science debate: Journals vs Conferences



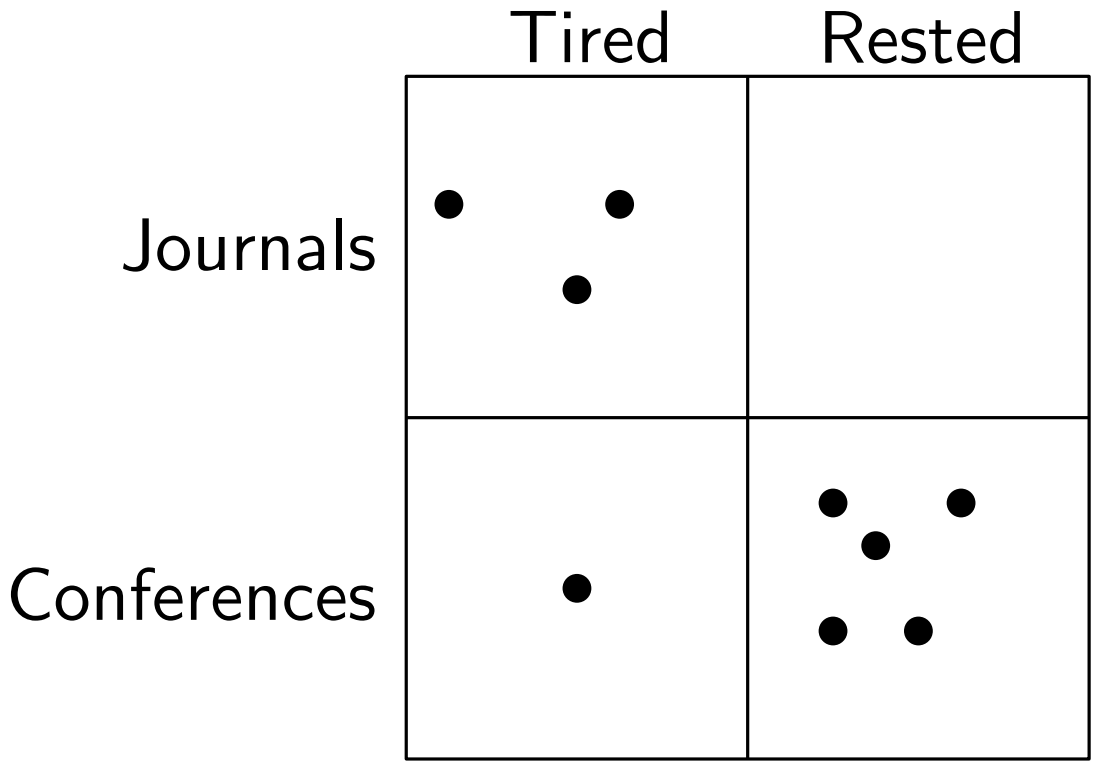
The computer science debate: Journals vs Conferences



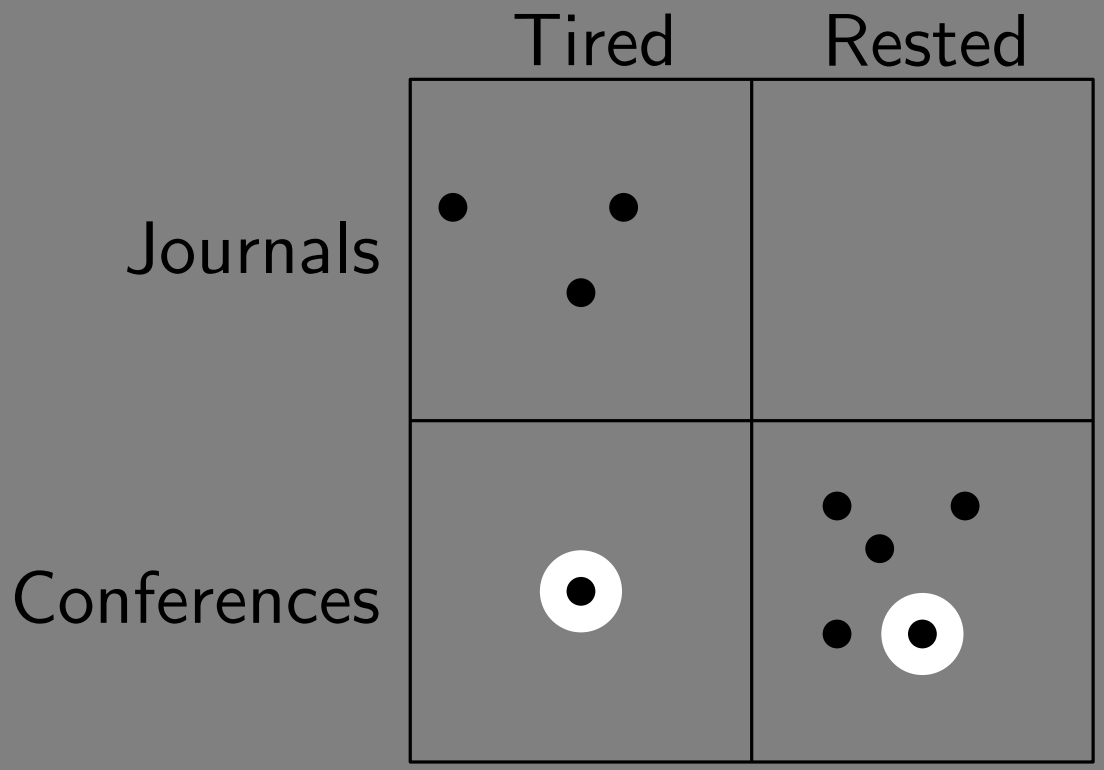
The computer science debate: Journals vs Conferences



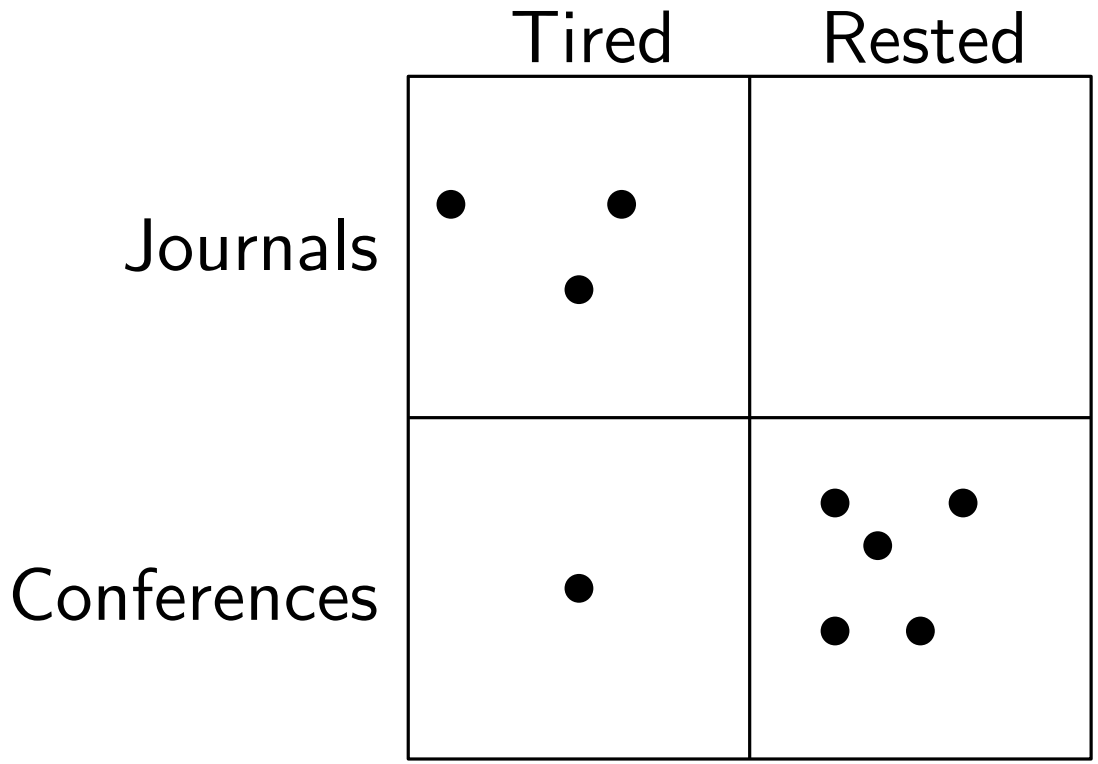
The computer science debate: Journals vs Conferences



The computer science debate: Journals vs Conferences

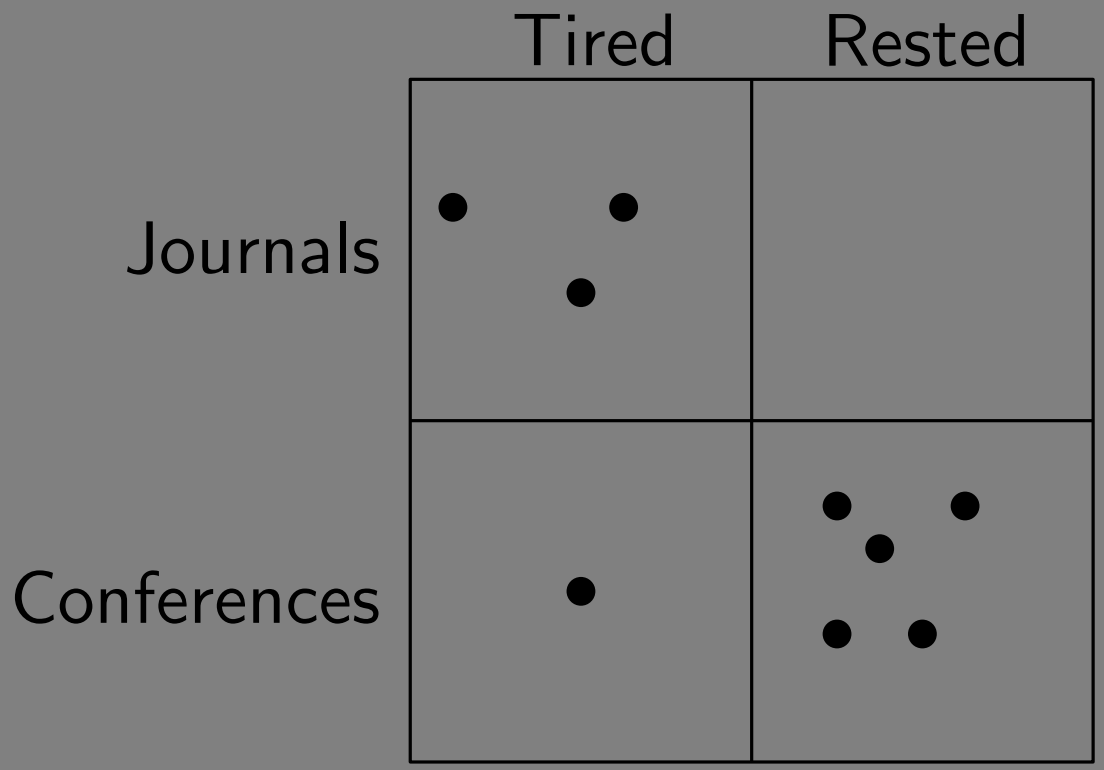


The computer science debate: Journals vs Conferences



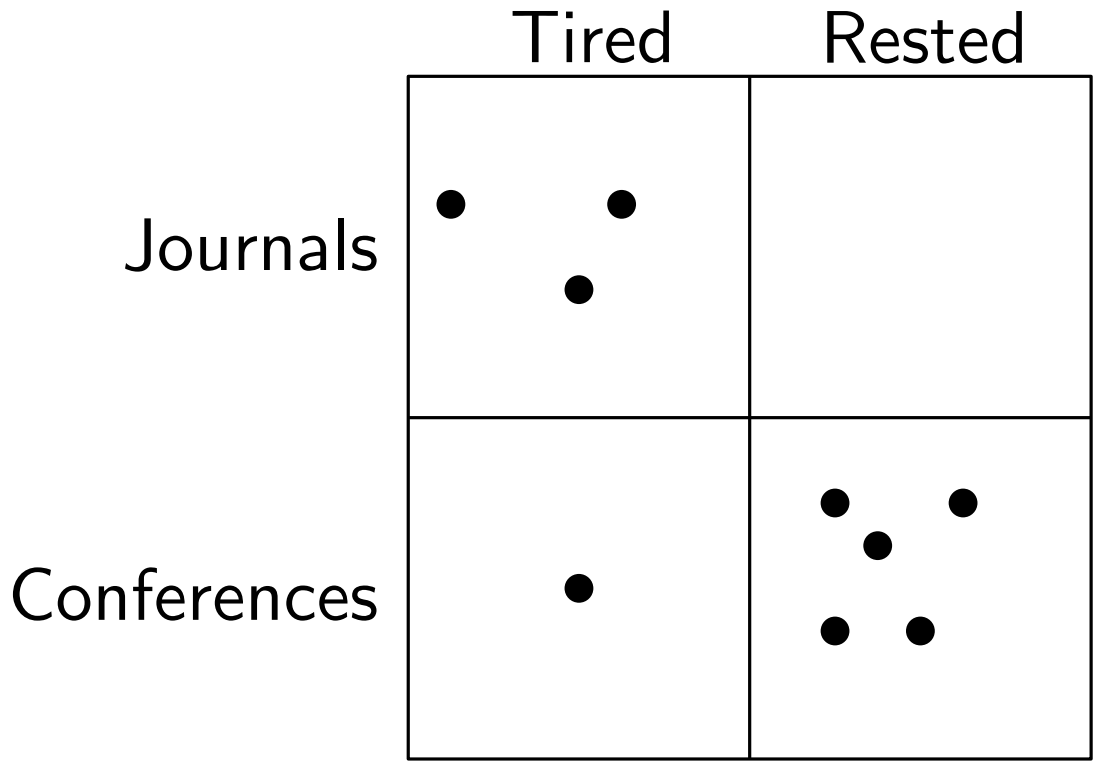
- If two thinking the same debate then they remain the same
- Rested scientists convince tired ones of anything
- If two feeling the same debate then the one for conferences convince the one for journals, and both feel tired afterwards

The computer science debate: Journals vs Conferences



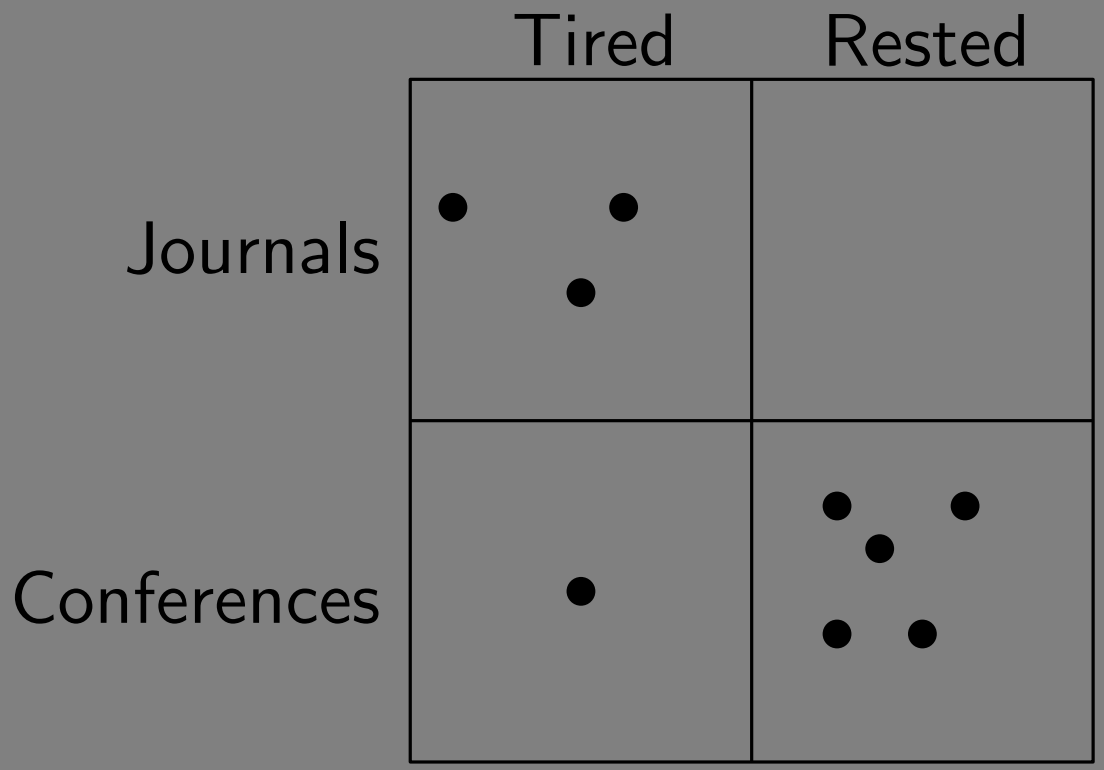
- If two thinking the same debate then they remain the same
- Rested scientists convince tired ones of anything
- If two feeling the same debate then the one for conferences convince the one for journals, and both feel tired afterwards

The computer science debate: Journals vs Conferences



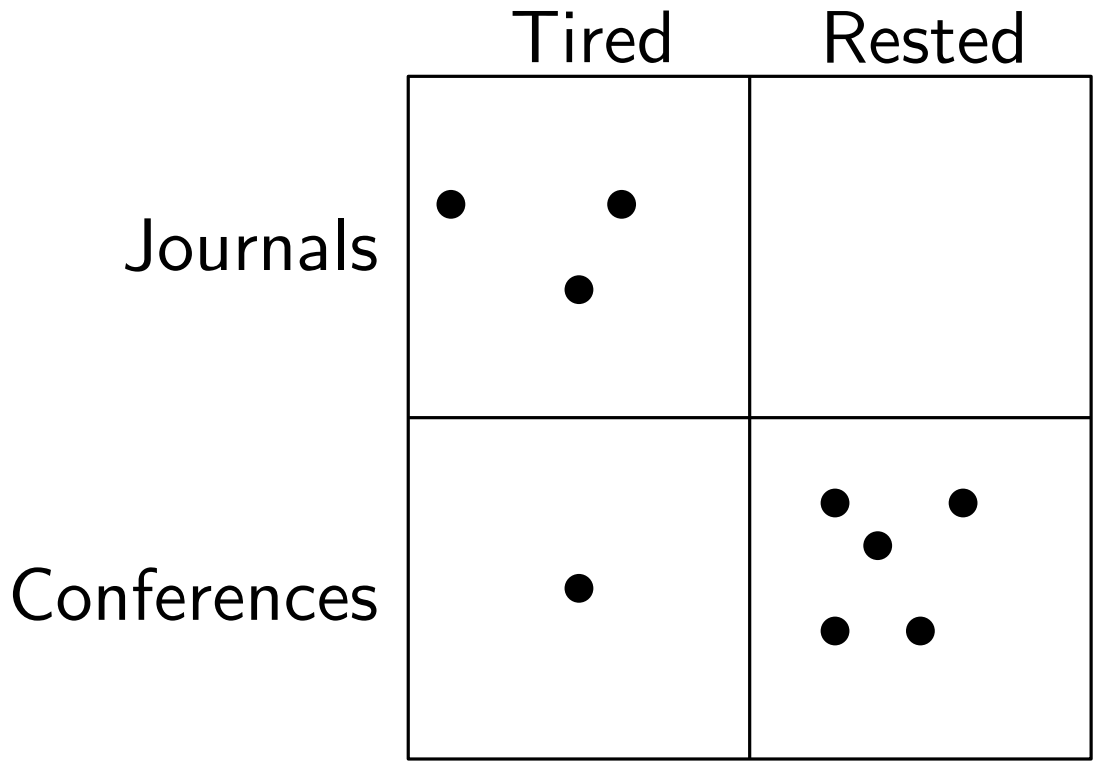
- $(\boxplus, \boxplus) \mapsto (\boxplus, \boxplus)$
- $(\boxplus, \boxplus) \mapsto (\boxplus, \boxplus)$

The computer science debate: Journals vs Conferences



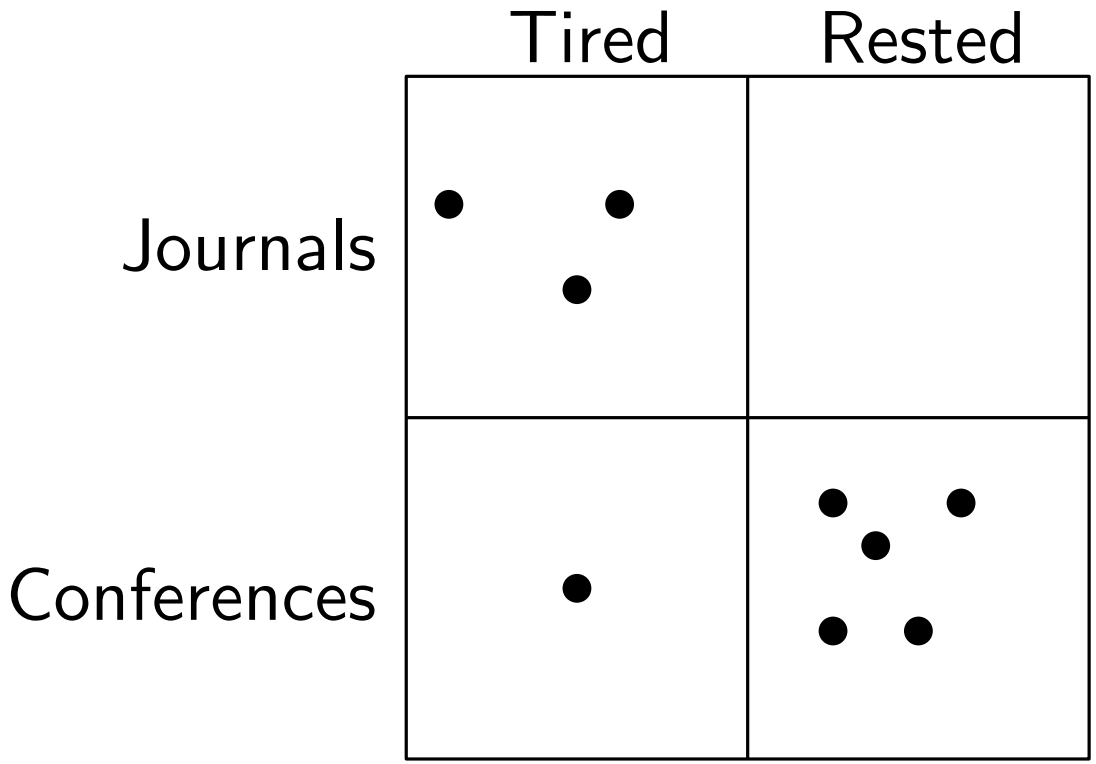
- If two thinking the same debate then they remain the same
- Rested scientists convince tired ones of anything
- If two feeling the same debate then the one for conferences convince the one for journals, and both feel tired afterwards

The computer science debate: Journals vs Conferences



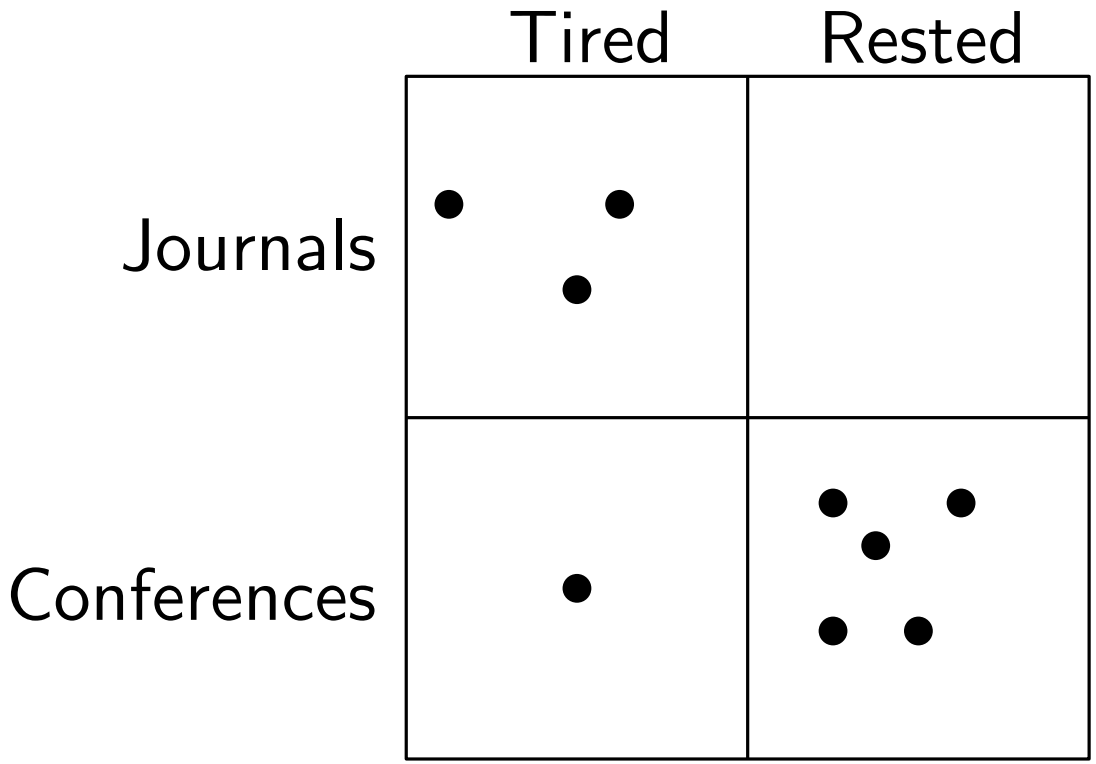
- $\left(\boxplus, \boxplus \right) \mapsto \left(\boxplus, \boxplus \right)$
- $\left(\boxplus, \boxplus \right) \mapsto \left(\boxplus, \boxplus \right)$

The computer science debate: Journals vs Conferences



What is the result of endless debates?

The computer science debate: Journals vs Conferences



¿initial distribution?
¿scheduler policy?

What is the result of endless debates?

Initial distribution

	Tired	Rested
Journals	**0**	
Conferences	**0**	

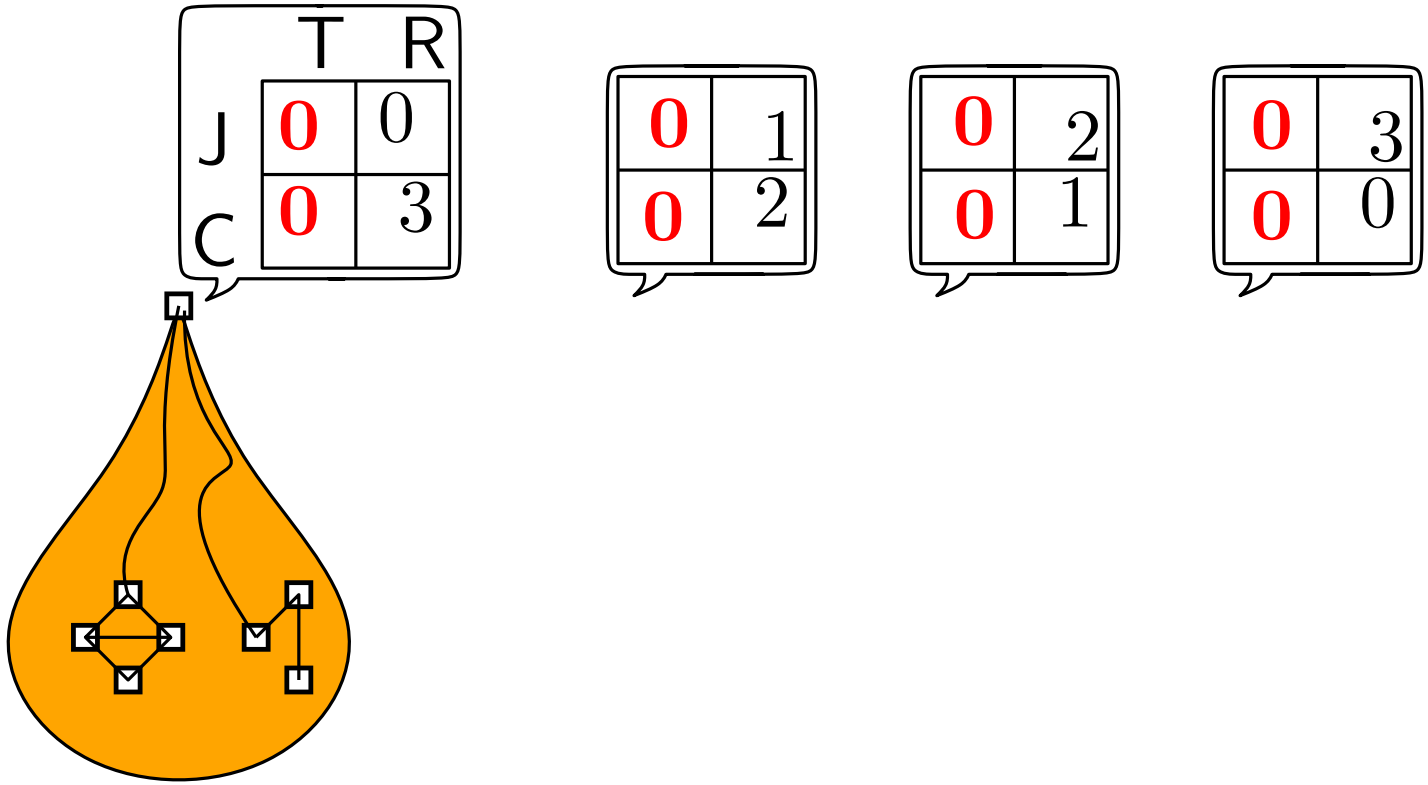
Initial distributions & reachability graphs

Take a population of 3 rested computer scientists

	T	R
J	**0**	0
C	**0**	3

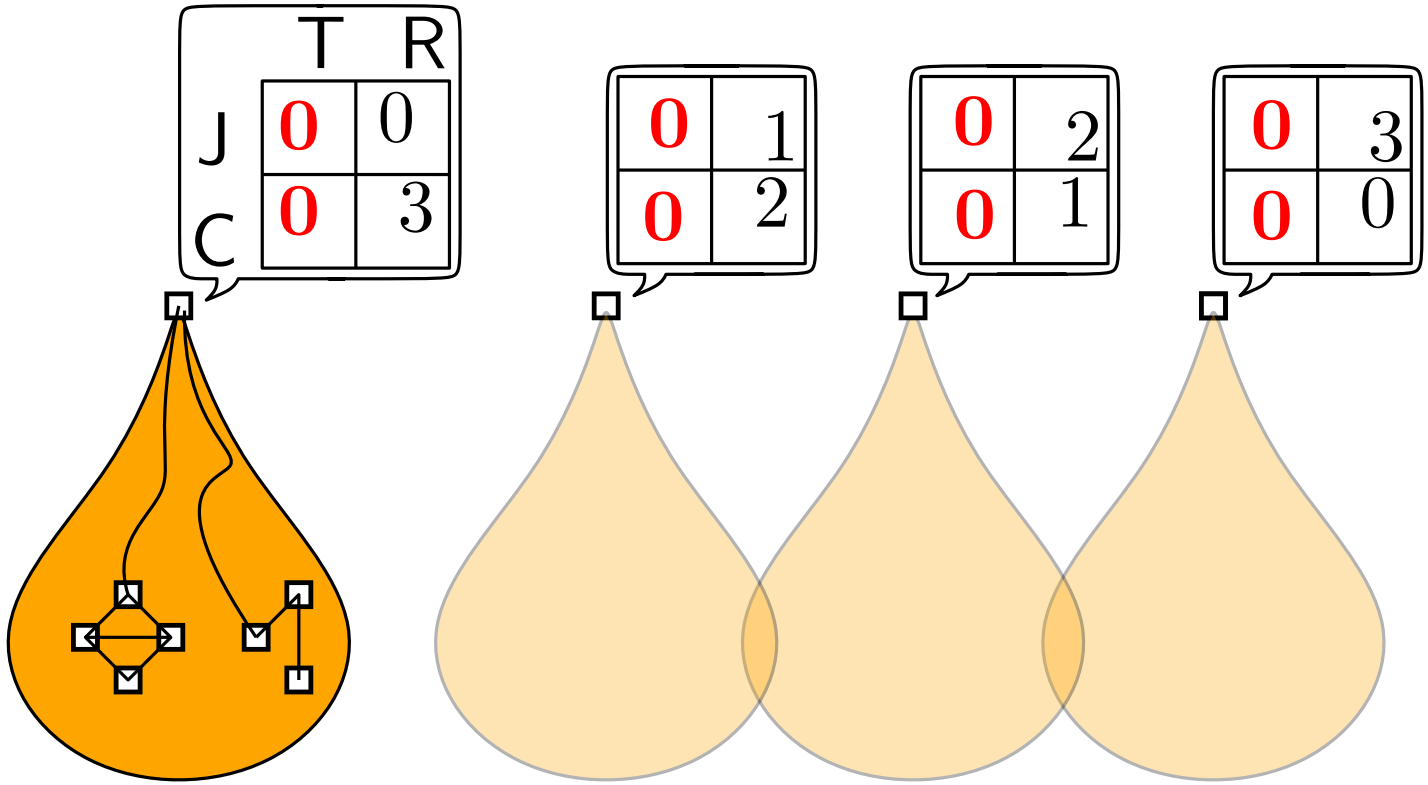
Initial distributions & reachability graphs

Take a population of 3 rested computer scientists



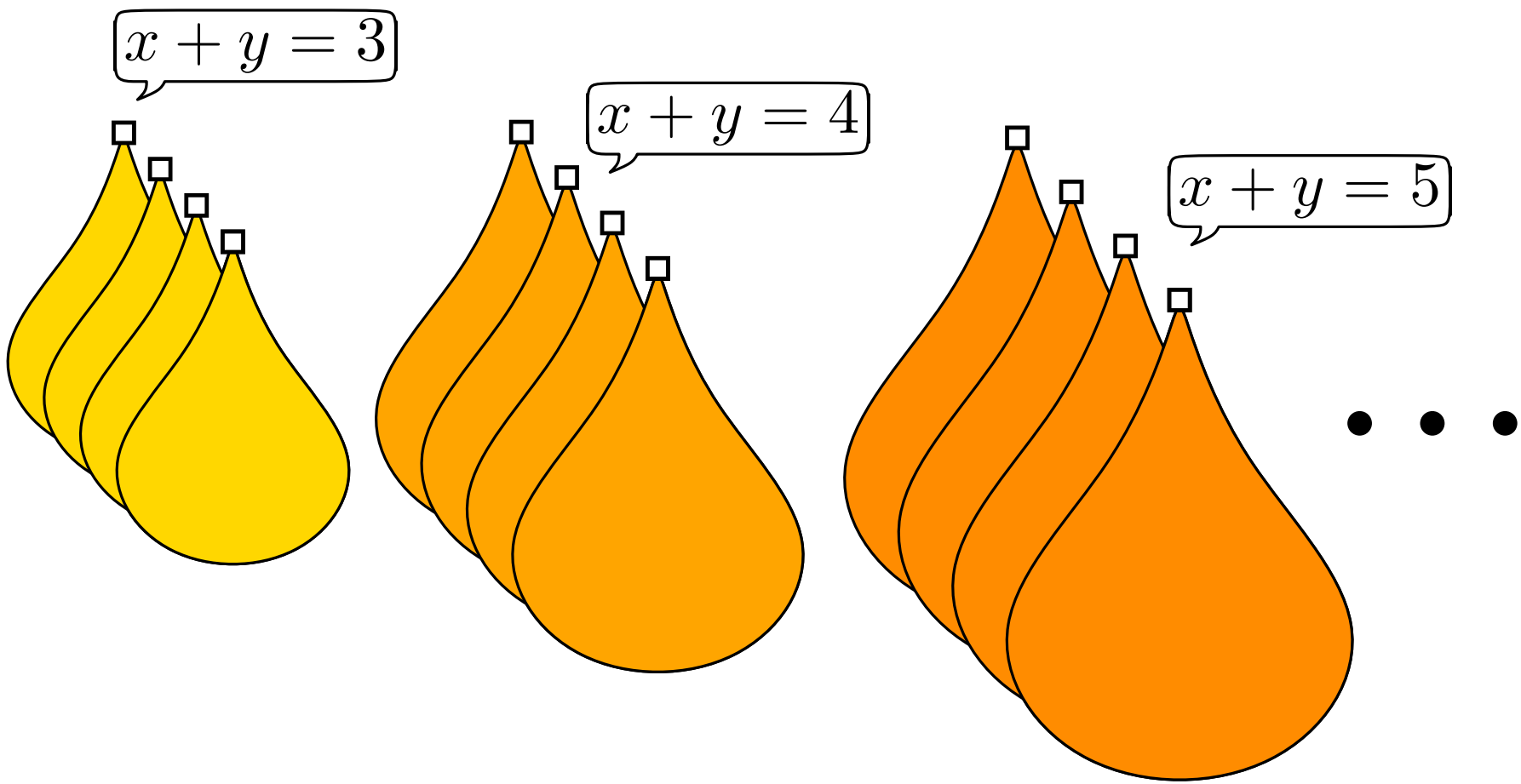
Initial distributions & reachability graphs

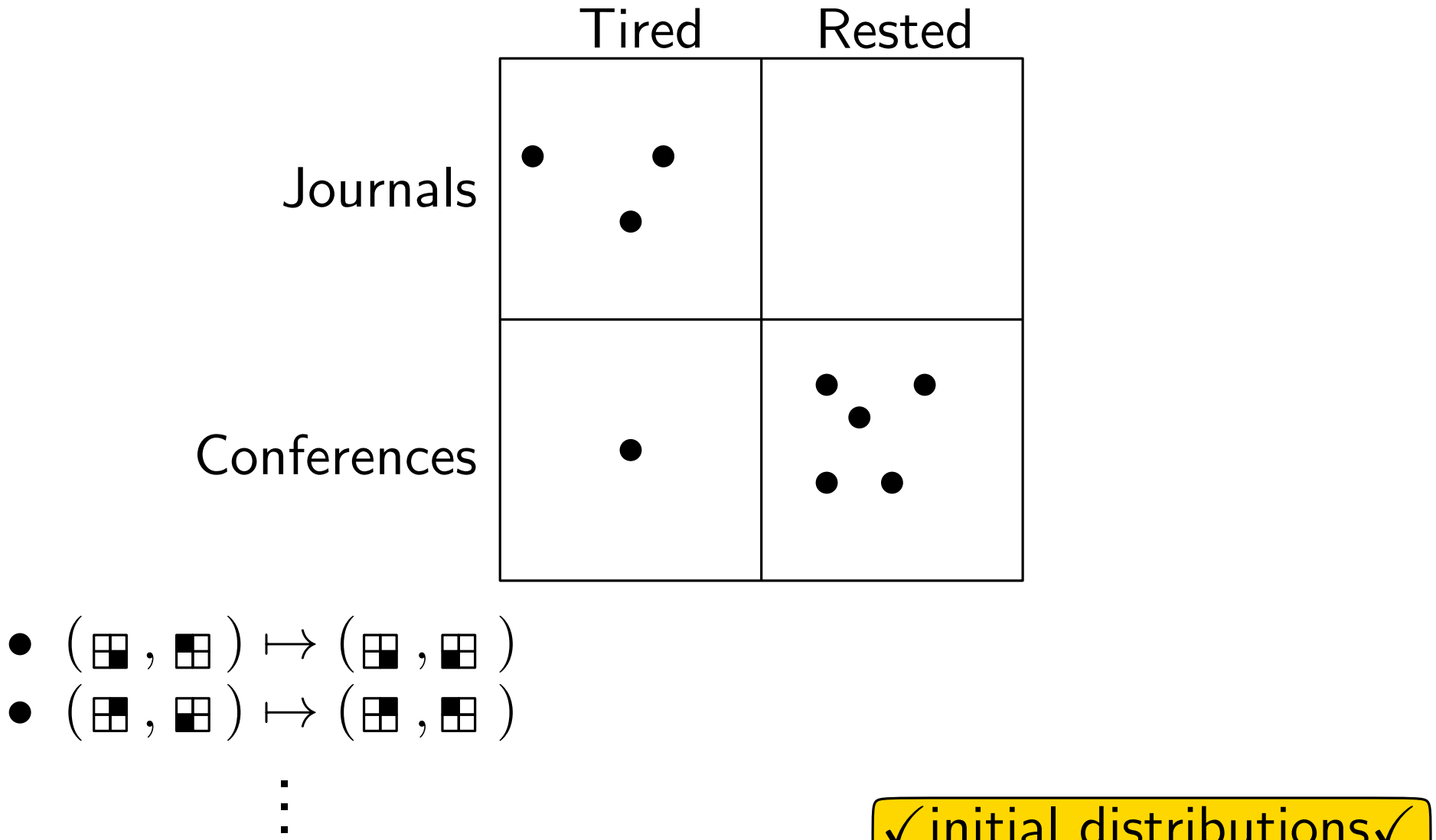
Take a population of 3 rested computer scientists



PP define infinitely many graphs

	T	R
J	**0**	x
C	**0**	y





- $\left(\boxplus\!\blacksquare\,,\,\blacksquare\!\boxplus\right) \mapsto \left(\boxplus\!\blacksquare\,,\boxplus\!\blacksquare\right)$
- $\left(\boxplus\!\blacksquare\,,\,\blacksquare\!\boxplus\right) \mapsto \left(\boxplus\!\blacksquare\,,\boxplus\!\blacksquare\right)$

$\vdots$

✓initial distributions✓
¿scheduler policy?

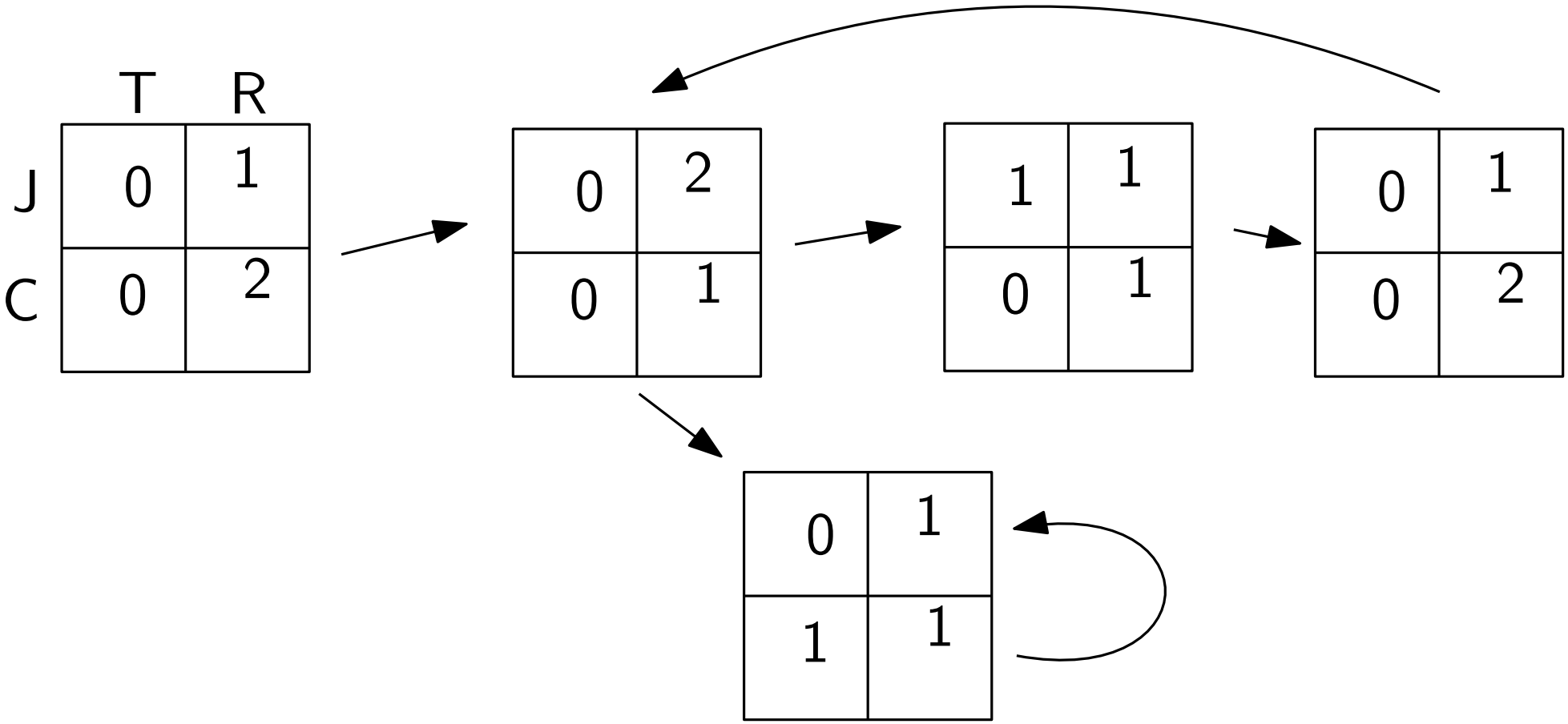
What is the result of endless debates?

Scheduler policy: fair

"if C appears infinitely often in a fair execution, then every step enabled at C is taken infinitely often in the execution"

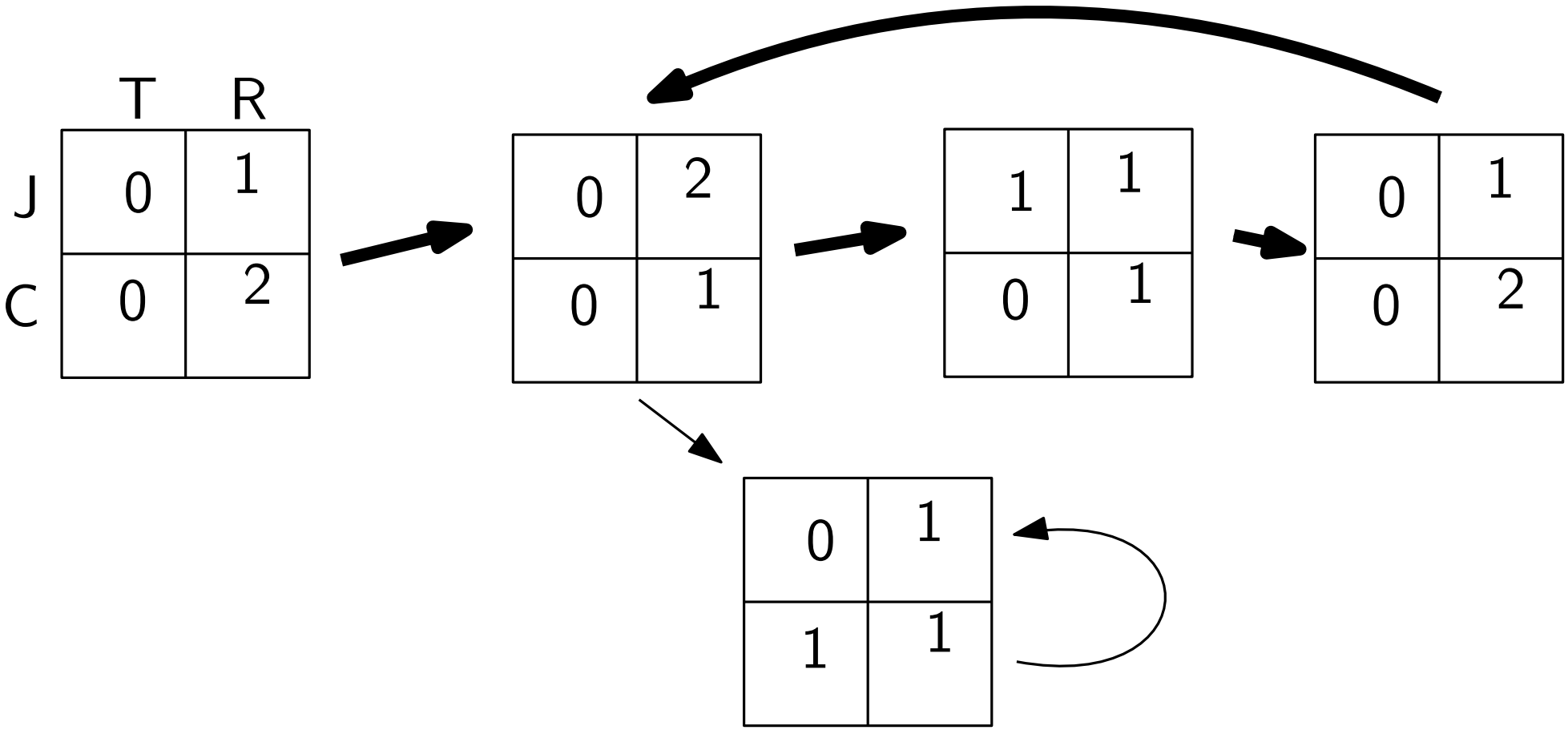
Scheduler policy: fair

"if C appears infinitely often in a fair execution, then every step enabled at C is taken infinitely often in the execution"



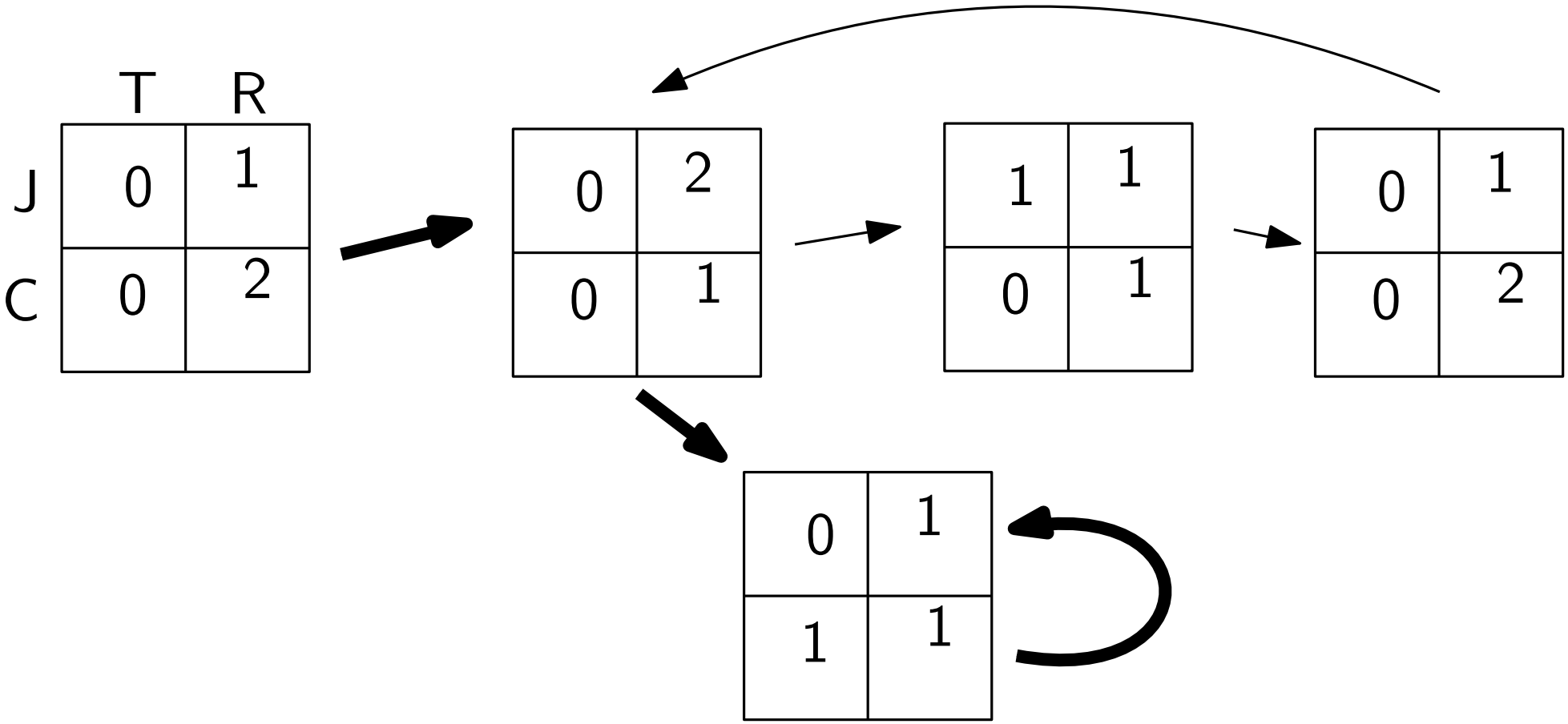
Scheduler policy: fair

"if C appears infinitely often in a fair execution, then every step enabled at C is taken infinitely often in the execution"



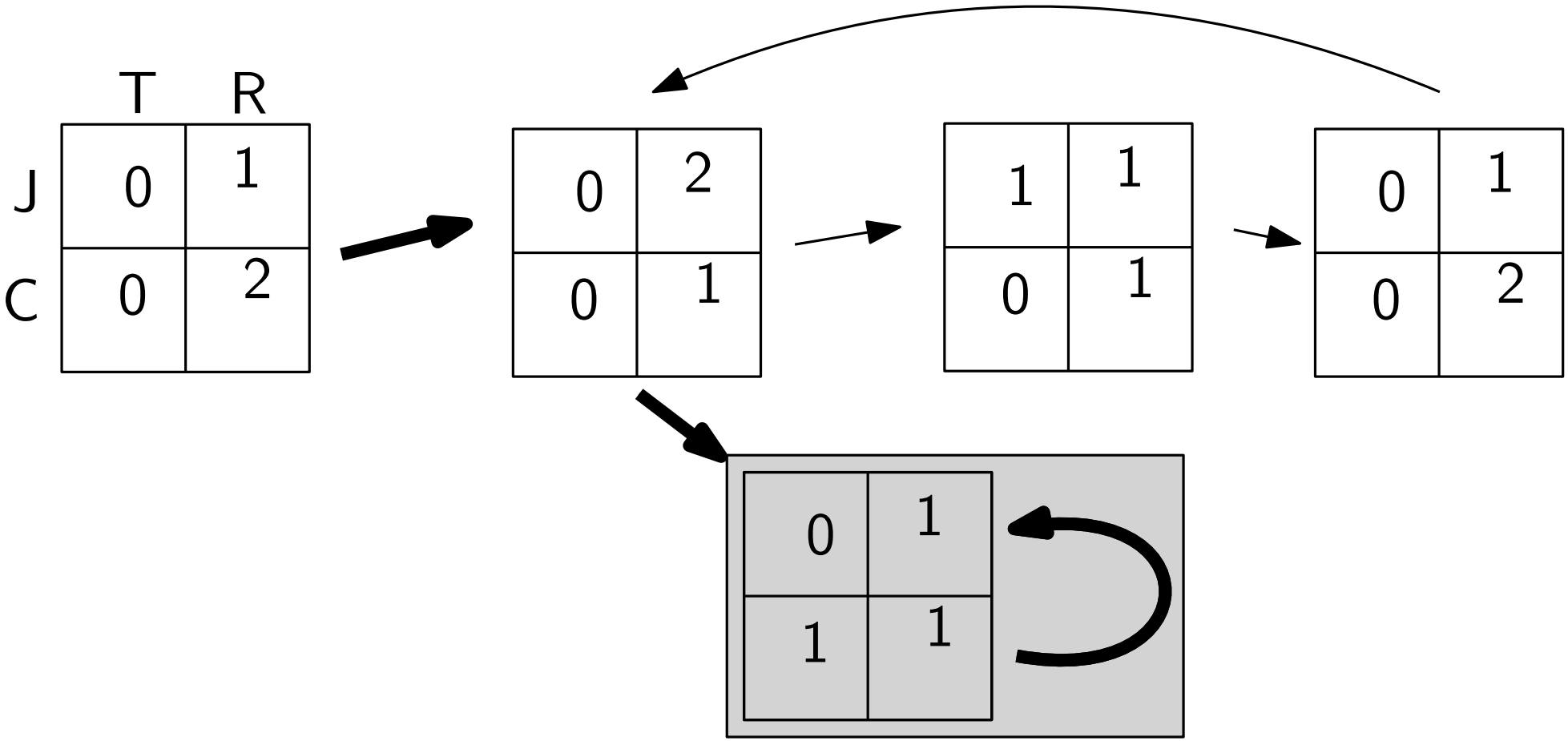
Scheduler policy: fair

"if C appears infinitely often in a fair execution, then every step enabled at C is taken infinitely often in the execution"



Scheduler policy: fair

"if C appears infinitely often in a fair execution, then every step enabled at C is taken infinitely often in the execution"



- infinite executions end up in Strongly Connected Components
- **fair** executions end up in a bottom SCC visiting **every** state

Under a fair scheduler

If for each initial distribution

1. the agents reach a stable consensus

2. the outcome only depends on the initial distribution

then

PP maps initial distributions onto consensus values

Under a fair scheduler

If for each initial distribution

1. the agents reach a stable consensus

2. the outcome only depends on the initial distribution

then

J or C

PP maps initial distributions onto consensus values

Under a fair scheduler

If for each initial distribution

1. the agents reach a stable consensus

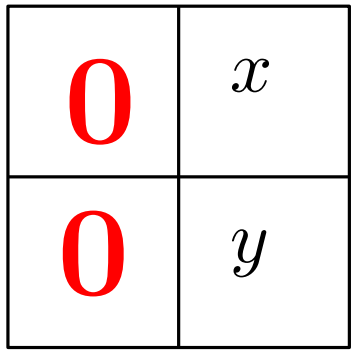
2. the outcome only depends on the initial distribution

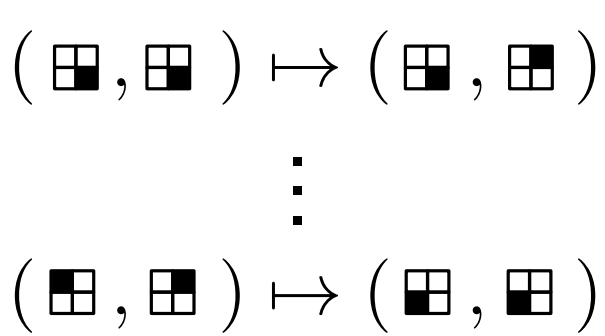
then

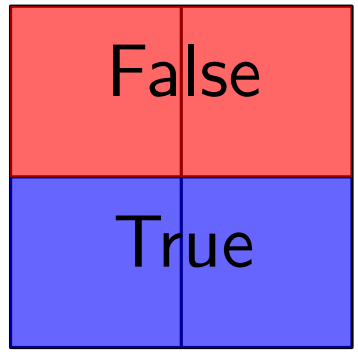
True or False

PP maps initial distributions onto consensus values

$$\varphi(x, y) \mapsto \{\mathsf{True}, \mathsf{False}\}$$







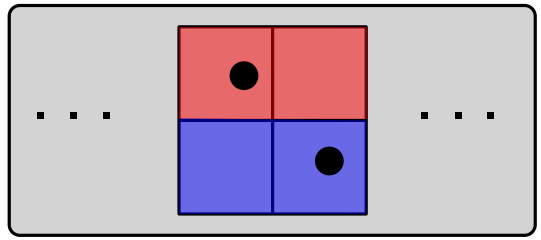
Does my PP compute a predicate?

Does my PP compute this predicate?

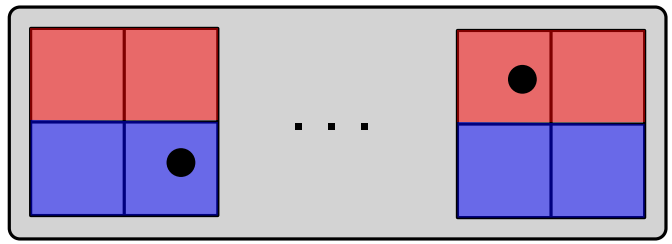
What predicate does my PP compute?

Does my PP compute a predicate?

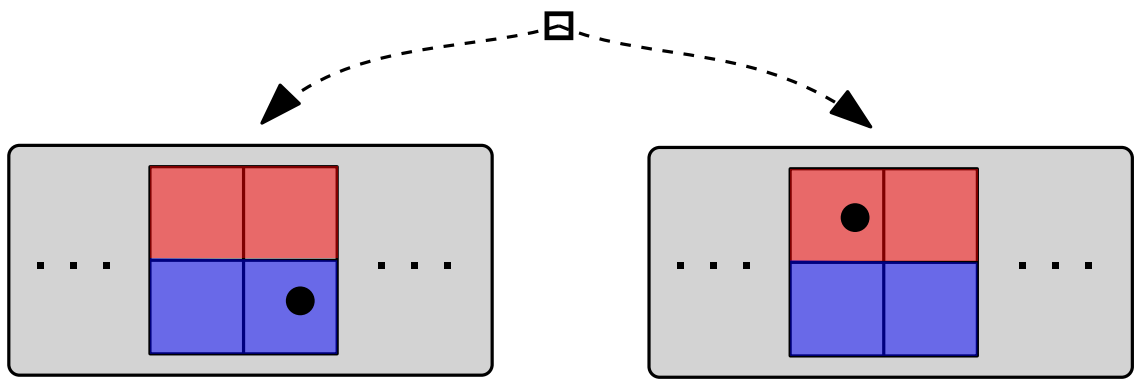
1. the agents reach a stable consensus under a fair scheduler
   - broken when, in a reachable bottom SCC,



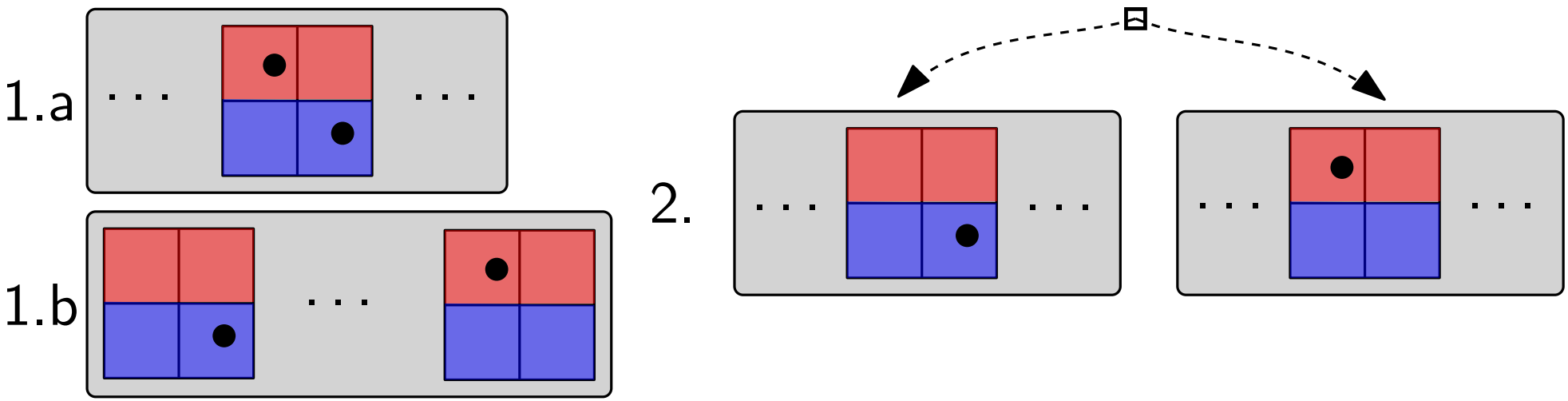
   - broken when, in a reachable bottom SCC,



2. the outcome only depends on the initial distribution
   - broken when, two bottom SCC are such that



PP does not compute a predicate iff either condition holds:



Equivalently, there exist configurations c_0, c_1, c_2 such that

c_0 is initial

c_1, c_2 belongs to bottom SCC $\wedge\ c_1(\blacksquare) > 0 \wedge\ c_2(\blacksquare) > 0$

c_1, c_2 are reachable from c_0

Decision procedure

c_0 is initial

c_1, c_2 belongs to bottom SCC$\wedge$ $c_1($$) > 0 \wedge$ $c_2($$) > 0$

c_1, c_2 are reachable from c_0

Decision procedure

c_0 is initial

c_1, c_2 belongs to bottom SCC$\wedge$ $c_1($$) > 0 \wedge$ $c_2($$) > 0$

c_1, c_2 are reachable from c_0

Decision procedure

c_0 is initial

c_1, c_2 belongs to bottom SCC$\wedge$ $c_1($$) > 0 \wedge$ $c_2($$) > 0$

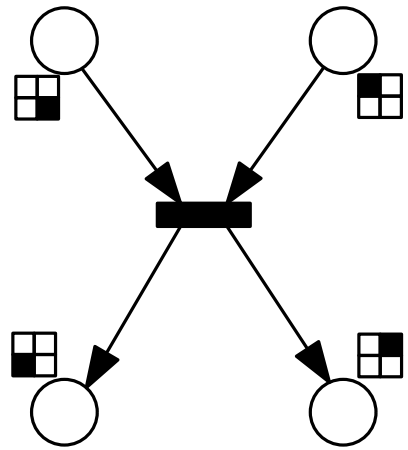
c_1, c_2 are reachable from c_0

Decision procedure

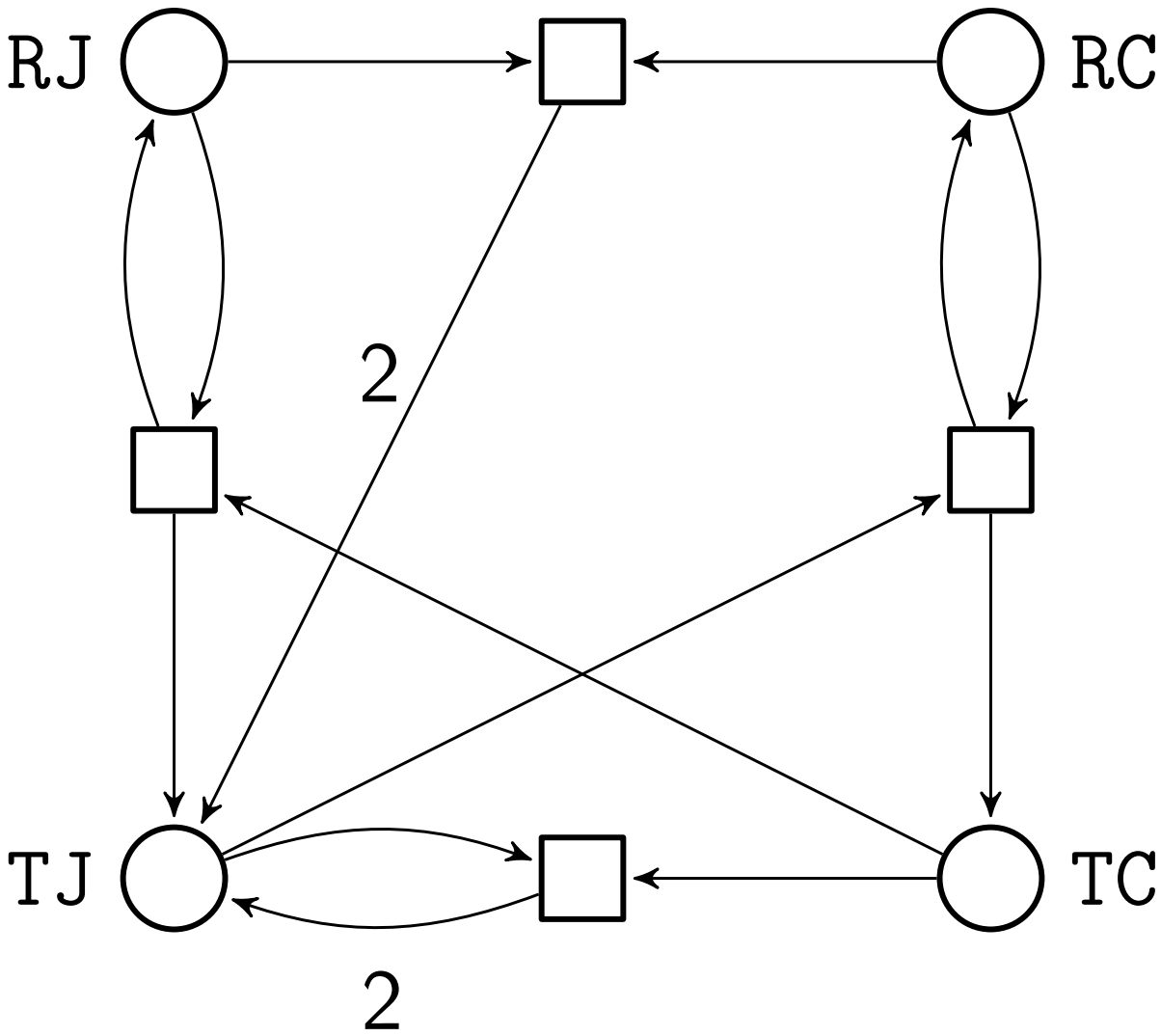
c_0 is initial

c_1, c_2 belongs to bottom SCC$\wedge$ $c_1($$) > 0 \wedge$ $c_2($$) > 0$

c_1, c_2 are reachable from c_0

Population Protocols	Petri nets
State	Place
Configuration	Marking
$(\boxplus, \boxplus) \mapsto (\boxplus, \boxplus)$	
PP	PN with infinitely many initial markings

Petrification of the debating scientists



c_0 is initial

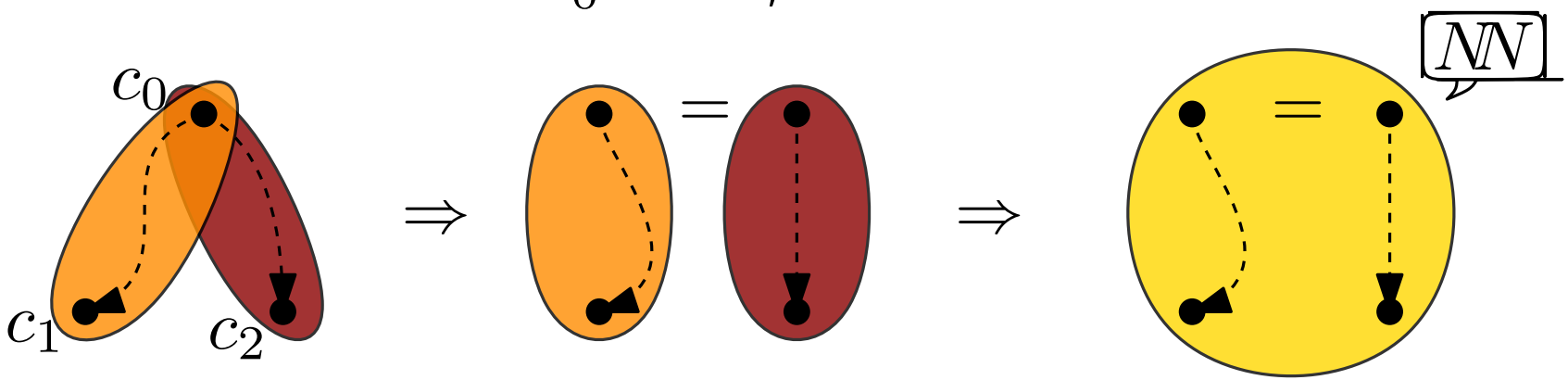
c_1, c_2 belongs to bottom SCC$\wedge$ $c_1($$) > 0 \wedge$ $c_2($$) > 0$

c_1, c_2 are reachable from c_0 in N, the Petrified PP

c_0 is initial

c_1, c_2 belongs to bottom SCC$\wedge$ $c_1($  $) > 0 \wedge$ $c_2($  $) > 0$

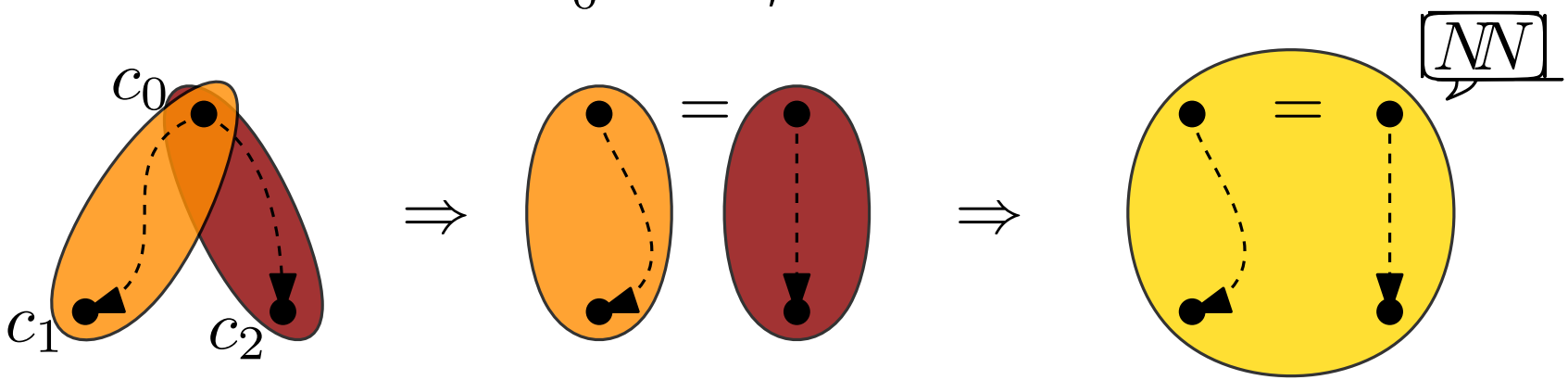
c_1, c_2 are reachable from c_0 in N, the Petrified PP



c_0 is initial

c_1, c_2 belongs to bottom SCC$\wedge$ $c_1($  $) > 0 \wedge$ $c_2($  $) > 0$

c_1, c_2 are reachable from c_0 in N, the Petrified PP

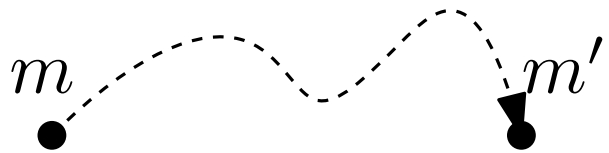


$A(C_0, C_0') \stackrel{\text{def}}{=} Init_N(C_0) \wedge Init_N(C_0') \wedge C_0 = C_0'$

$B(C_1, C_2) \stackrel{\text{def}}{=} \Omega_N(C_1) \wedge \Omega_N(C_2) \wedge C_1($  $) > 0 \wedge C_2($  $) > 0$

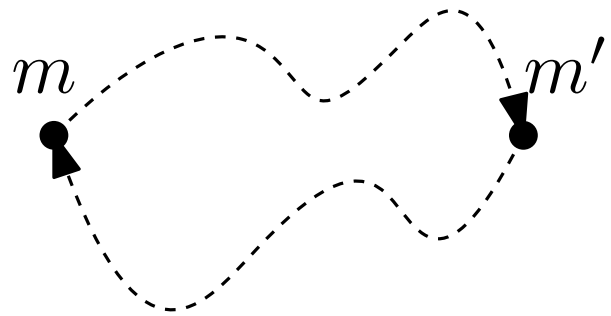
Check reachability from A to B in $\mathbb{NN}$

How to compute Ω?



The reachability relation is not Presburger definable

How to compute Ω?



The reachability relation is not Presburger definable

But, the mutual reachability relation is effectively Presburger

that is, a Presburger formula $MR(m, m')$ is computable

Finally

$$A(C_0, C_0') \stackrel{\text{def}}{=} Init_N(C_0) \wedge Init_N(C_0') \wedge C_0 = C_0'$$

$$B(C_1, C_2) \stackrel{\text{def}}{=} \Omega_N(C_1) \wedge \Omega_N(C_2) \wedge C_1(\blacksquare) > 0 \wedge C_2(\blacksquare) > 0$$

Check reachability from A to B in $\mathbb{NN}$

$$\Omega_N(C) \stackrel{\text{def}}{=} \forall C', C'' \colon (MR_N(C, C') \wedge C' \rightarrow C'') \Rightarrow MR_N(C, C'')$$

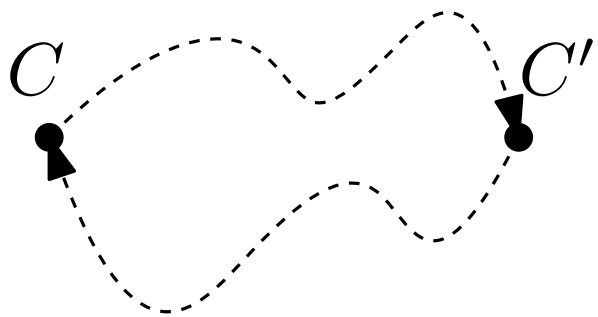
Finally

$$A(C_0, C_0') \stackrel{\text{def}}{=} Init_N(C_0) \wedge Init_N(C_0') \wedge C_0 = C_0'$$

$$B(C_1, C_2) \stackrel{\text{def}}{=} \Omega_N(C_1) \wedge \Omega_N(C_2) \wedge C_1(\blacksquare) > 0 \wedge C_2(\blacksquare) > 0$$

Check reachability from A to B in $\mathbb{NN}$

$$\Omega_N(C) \stackrel{\text{def}}{=} \forall C', C'' : (MR_N(C, C') \wedge C' \rightarrow C'') \Rightarrow MR_N(C, C'')$$



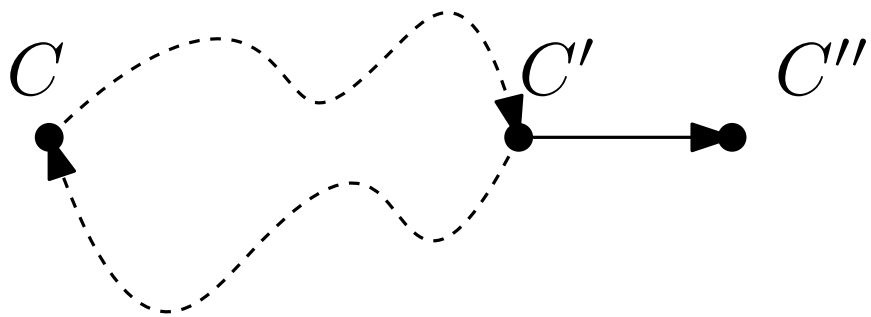
Finally

$$A(C_0, C_0') \stackrel{\text{def}}{=} Init_N(C_0) \wedge Init_N(C_0') \wedge C_0 = C_0'$$

$$B(C_1, C_2) \stackrel{\text{def}}{=} \Omega_N(C_1) \wedge \Omega_N(C_2) \wedge C_1(\blacksquare) > 0 \wedge C_2(\blacksquare) > 0$$

Check reachability from A to B in $\mathbb{NN}$

$$\Omega_N(C) \stackrel{\text{def}}{=} \forall C', C'' : (MR_N(C, C') \wedge C' \rightarrow C'') \Rightarrow MR_N(C, C'')$$



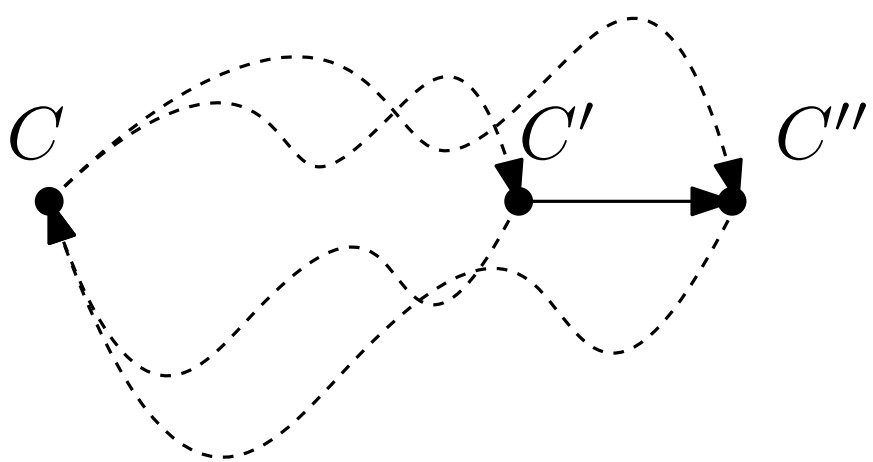
Finally

$$A(C_0, C_0') \stackrel{\text{def}}{=} Init_N(C_0) \wedge Init_N(C_0') \wedge C_0 = C_0'$$

$$B(C_1, C_2) \stackrel{\text{def}}{=} \Omega_N(C_1) \wedge \Omega_N(C_2) \wedge C_1(\blacksquare) > 0 \wedge C_2(\blacksquare) > 0$$

Check reachability from A to B in $\mathbb{NN}$

$$\Omega_N(C) \stackrel{\text{def}}{=} \forall C', C'' : (MR_N(C, C') \wedge C' \rightarrow C'') \Rightarrow MR_N(C, C'')$$



Does my PP compute this predicate?

What predicate does my PP compute?

What predicate can PPs compute?

PP compute exactly the Presburger predicates [Angluin et al.,'04]

What predicate can PPs compute?

PP compute exactly the Presburger predicates [Angluin et al.,'04]

Not **less** than Presburger is "easy": treshold, $\equiv_m$, $\wedge, \vee, \neg$

Not **more** Presburger is "involved"

Does my PP compute this (Presburger) predicate?

What (Presburger) predicate does my PP compute?

decidable

Does my PP compute this (Presburger) predicate?

effectively computable

What (Presburger) predicate does my PP compute?

Does my PP compute (Presburger) predicate Π?

Does my PP compute (Presburger) predicate Π?

It does **not** compute Π when:

- for a valuation $v_1, \ldots, v_n$ such that $\Pi(v_1, \ldots, v_n) =$ 

$Init_N(C_0)$ reaches $C_1($  $) > 0 \wedge \Omega_N(C_1)$ where

$$C_0(x_i) = \begin{cases} v_i & \text{if } 1 \leq i \leq n \\ \mathbf{0} & \text{elsewhere} \end{cases}$$

Does my PP compute (Presburger) predicate Π?

It does **not** compute Π when:

- for a valuation $v_1, \ldots, v_n$ such that $\Pi(v_1, \ldots, v_n) =$ 

$Init_N(C_0)$ reaches $C_1($  $) > 0 \wedge \Omega_N(C_1)$ where

$$C_0(x_i) = \begin{cases} v_i & \text{if } 1 \leq i \leq n \\ \mathbf{0} & \text{elsewhere} \end{cases}$$

OR

- for a valuation $v_1, \ldots, v_n$ such that $\Pi(v_1, \ldots, v_n) =$ 

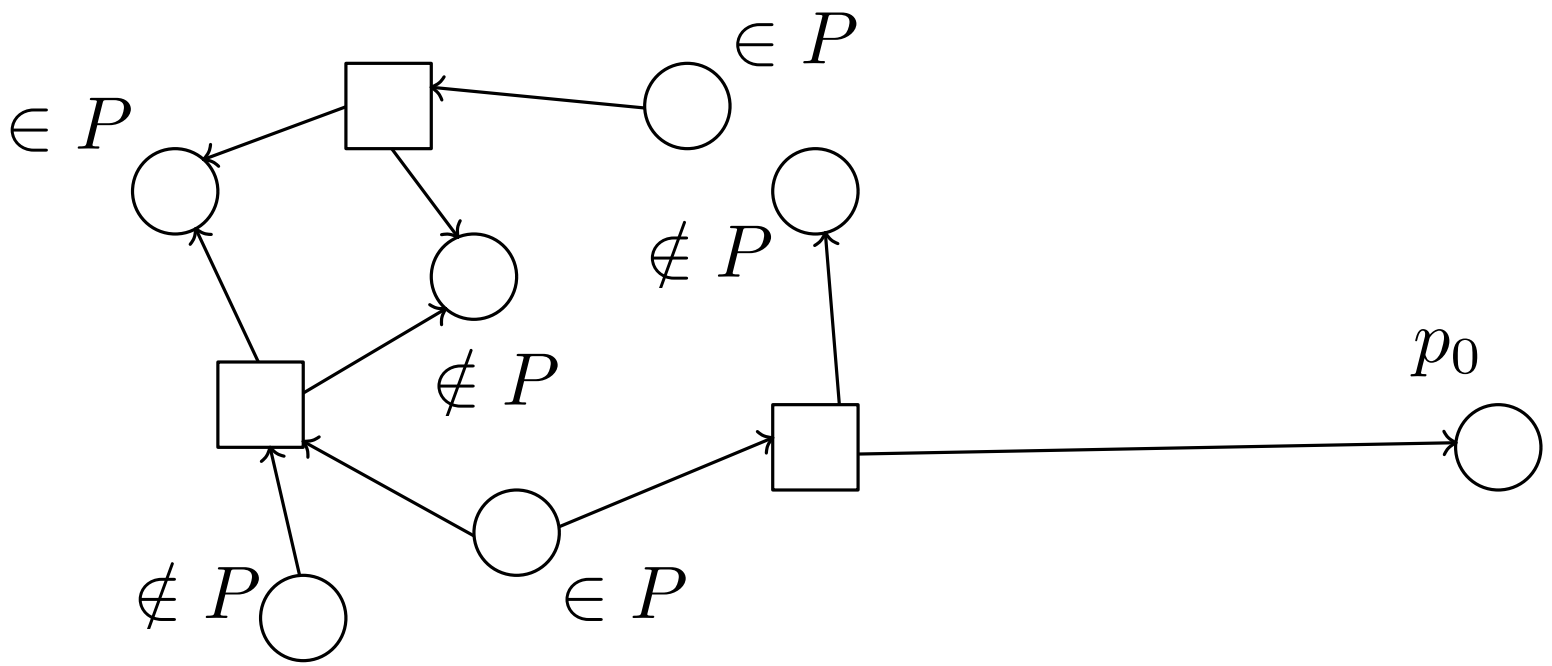
$Init_N(C_0)$ reaches $C_1($  $) > 0 \wedge \Omega_N(C_1)$ where

$$C_0(x_i) = \begin{cases} v_i & \text{if } 1 \leq i \leq n \\ \mathbf{0} & \text{elsewhere} \end{cases}$$

“Does my PP compute a predicate?” is PN-reachability hard

"Does my PP compute a predicate?" is PN-reachability hard

Does $\mathbf{p_0} \in Reach(N, M)$ for some M s.t. $M(p_0 \cup P) = 0$?



N "looks like" a PP: transitions have 1 or 2 input/output arcs

p_0 is "write only": no transition consumes from p_0

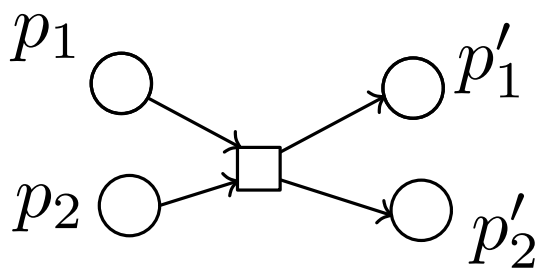
PN-reachability hardness (con'td)

PP states $=$ {places of N} $\cup$ $\{Budget\}$ $\cup$ $\{zombie\}$

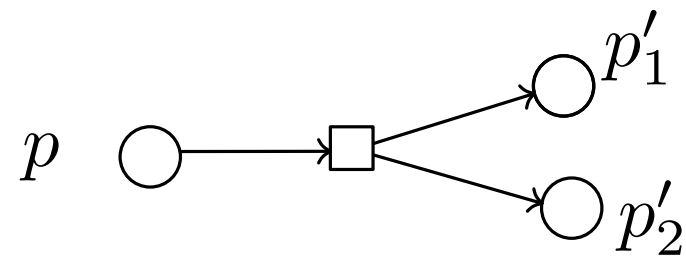
PN-reachability hardness (con'td)

PP states = {places of N} $\cup$ {$Budget$} $\cup$ {$zombie$}

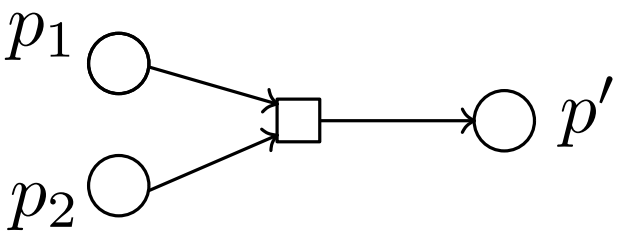
PP transitions:



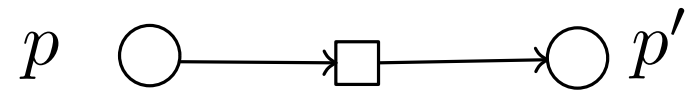
$(p_1, p_2) \mapsto (p'_1, p'_2)$



$(p, Budget) \mapsto (p'_1, p'_2)$



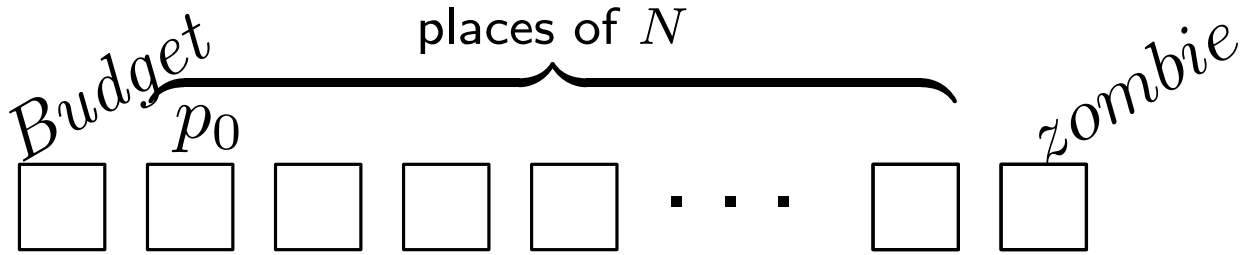
$(p_1, p_2) \mapsto (p', Budget)$



$(p, Budget) \mapsto (p', Budget)$

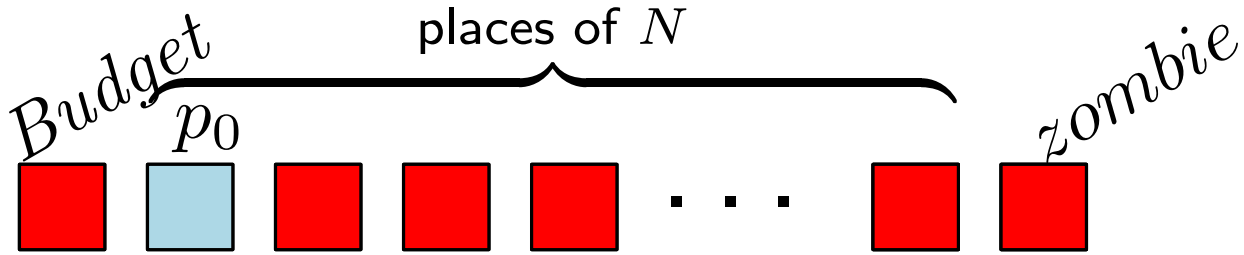
From PN reachability to "does my PP compute a predicate?"

- $\mathbf{p_0} \in Reach(N, M)$ for *some* marking M



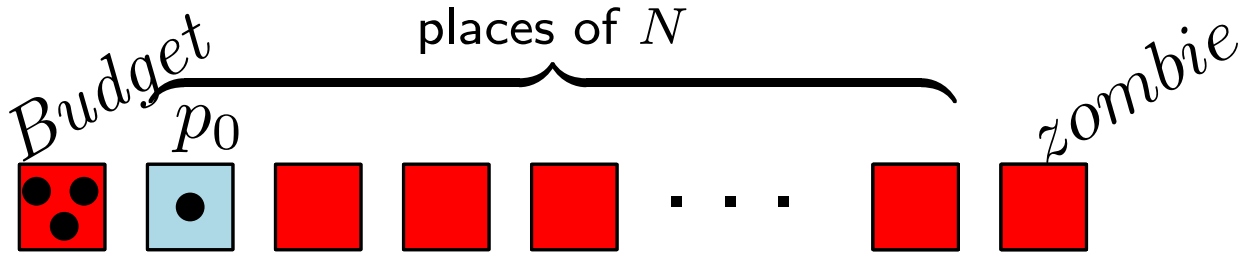
From PN reachability to "does my PP compute a predicate?"

- $\mathbf{p_0} \in Reach(N, M)$ for *some* marking M



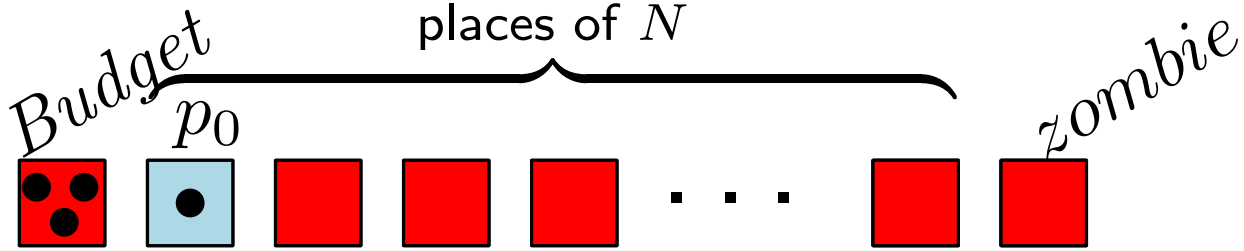
From PN reachability to "does my PP compute a predicate?"

- $\mathbf{p_0} \in Reach(N, M)$ for *some* marking M

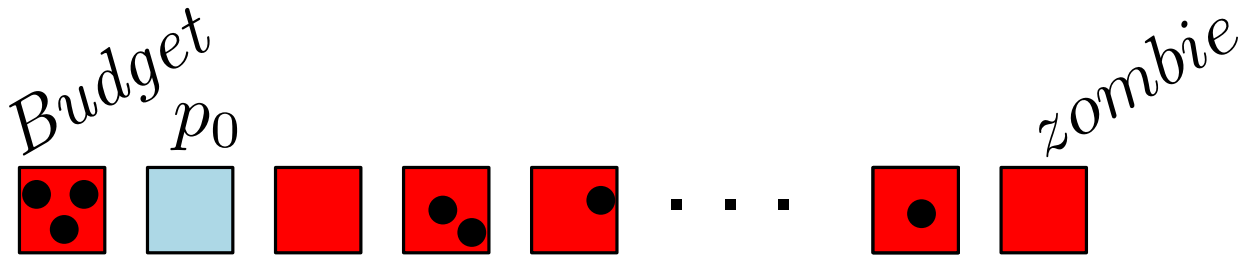


From PN reachability to "does my PP compute a predicate?"

- $\mathbf{p_0} \in Reach(N, M)$ for *some* marking M

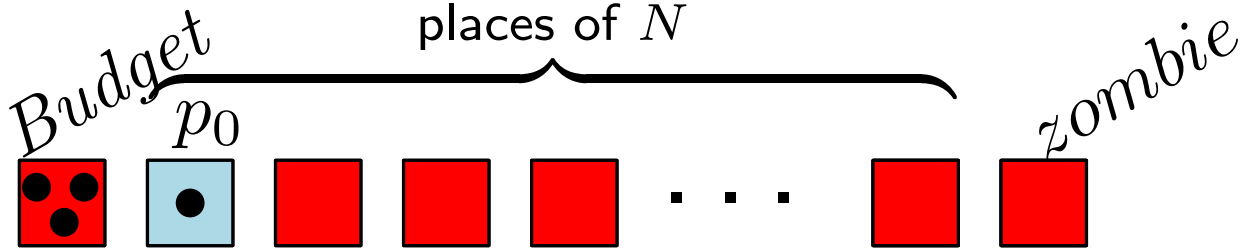


- $\mathbf{p_0} \in Reach(N, M)$ for **no** marking M

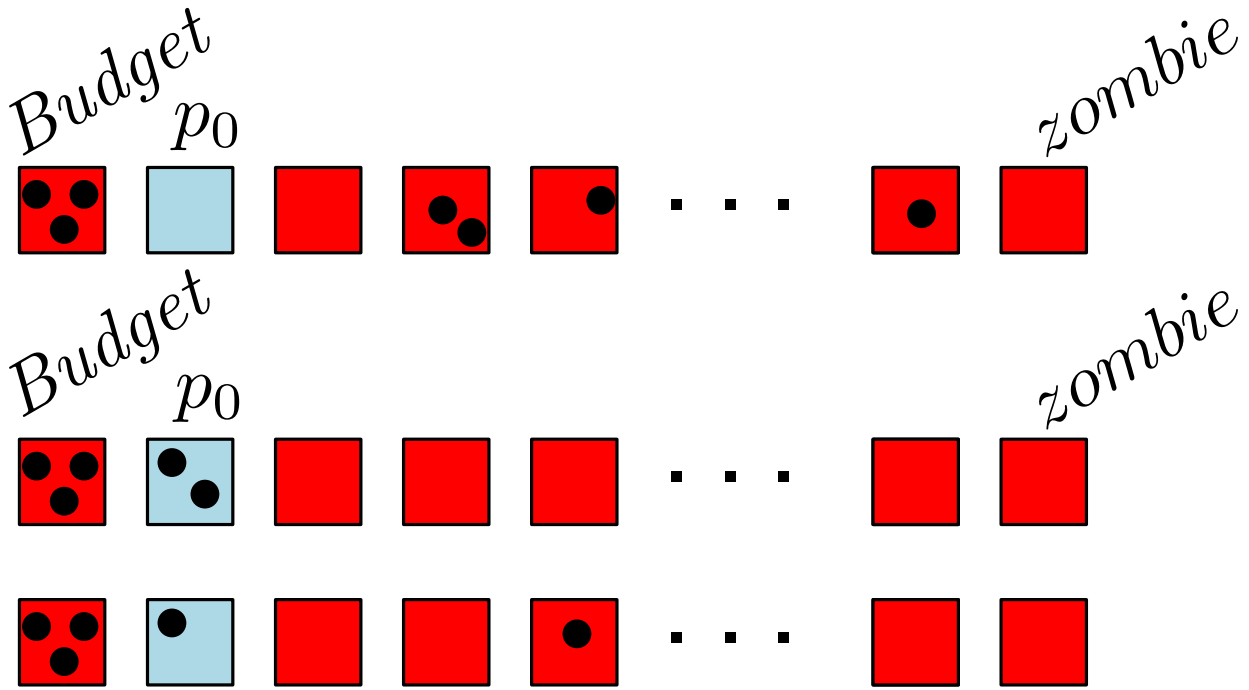


From PN reachability to "does my PP compute a predicate?"

- $\mathbf{p_0} \in Reach(N, M)$ for *some* marking M

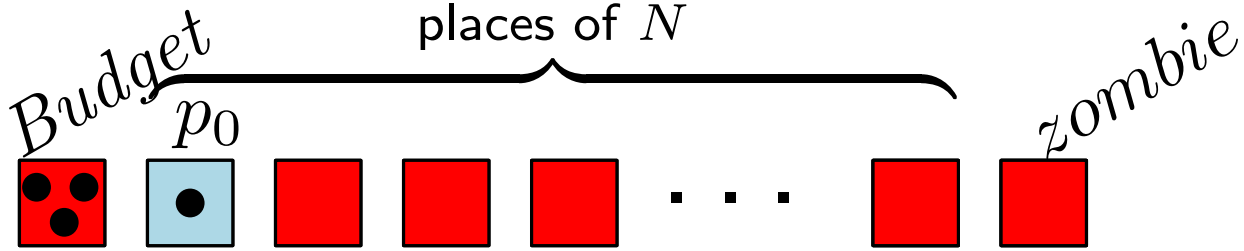


- $\mathbf{p_0} \in Reach(N, M)$ for **no** marking M

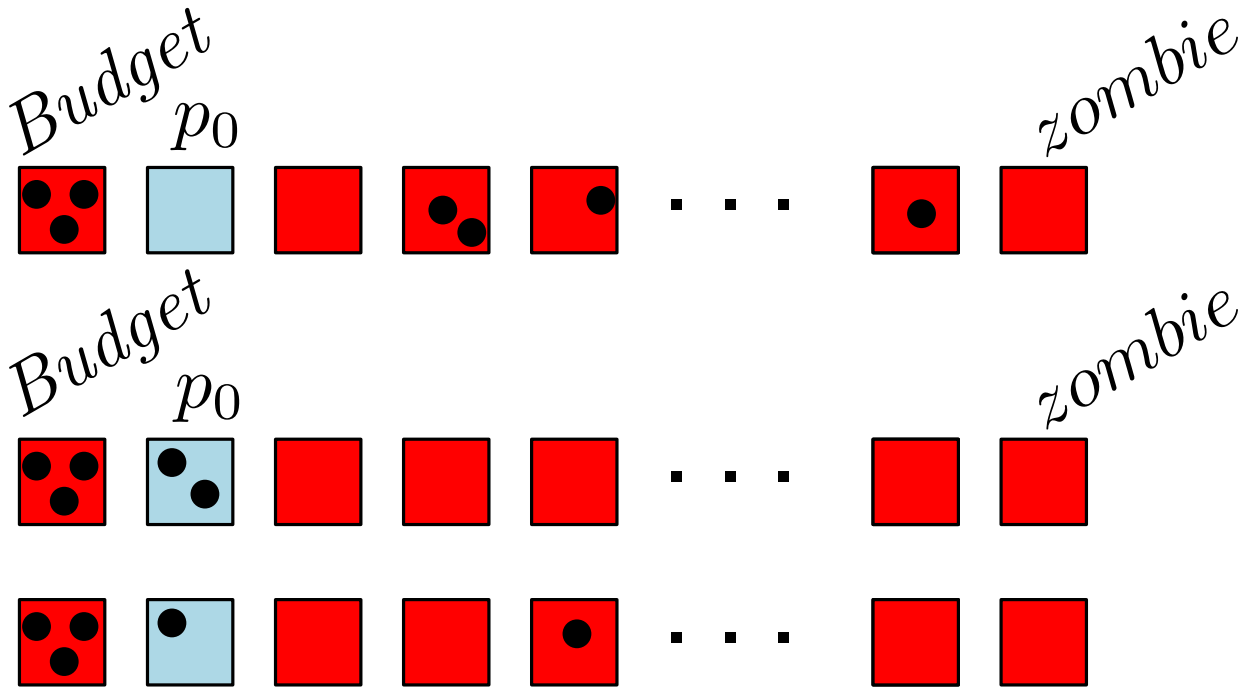


From PN reachability to "does my PP compute a predicate?"

- $\mathbf{p_0} \in Reach(N, M)$ for *some* marking M



- $\mathbf{p_0} \in Reach(N, M)$ for **no** marking M



$(p_0, X) \mapsto (p_0, zombie) \quad \forall X \in$ places of N

$(zombie, Y) \mapsto (zombie, zombie) \quad \forall Y$

From general PN-reachability to "our" PN-reachability

Does $M \in Reach(N, M_0)$?

From general PN-reachability to "our" PN-reachability

Does $M \in Reach(N, M_0)$?

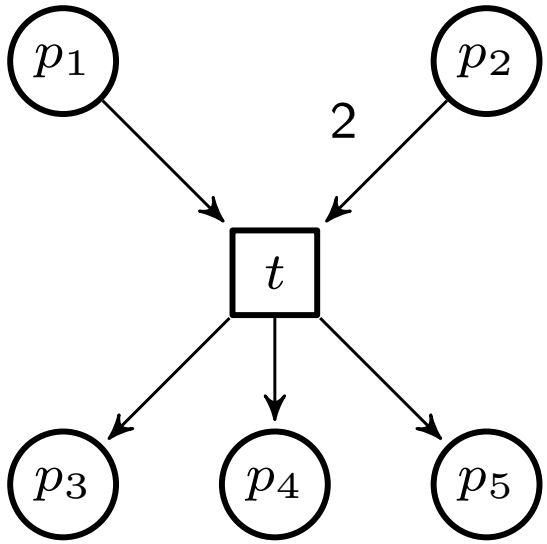
Does $M \in Reach(N_0, M_0)$ from some M with $M(\hat{p}) = 0$?

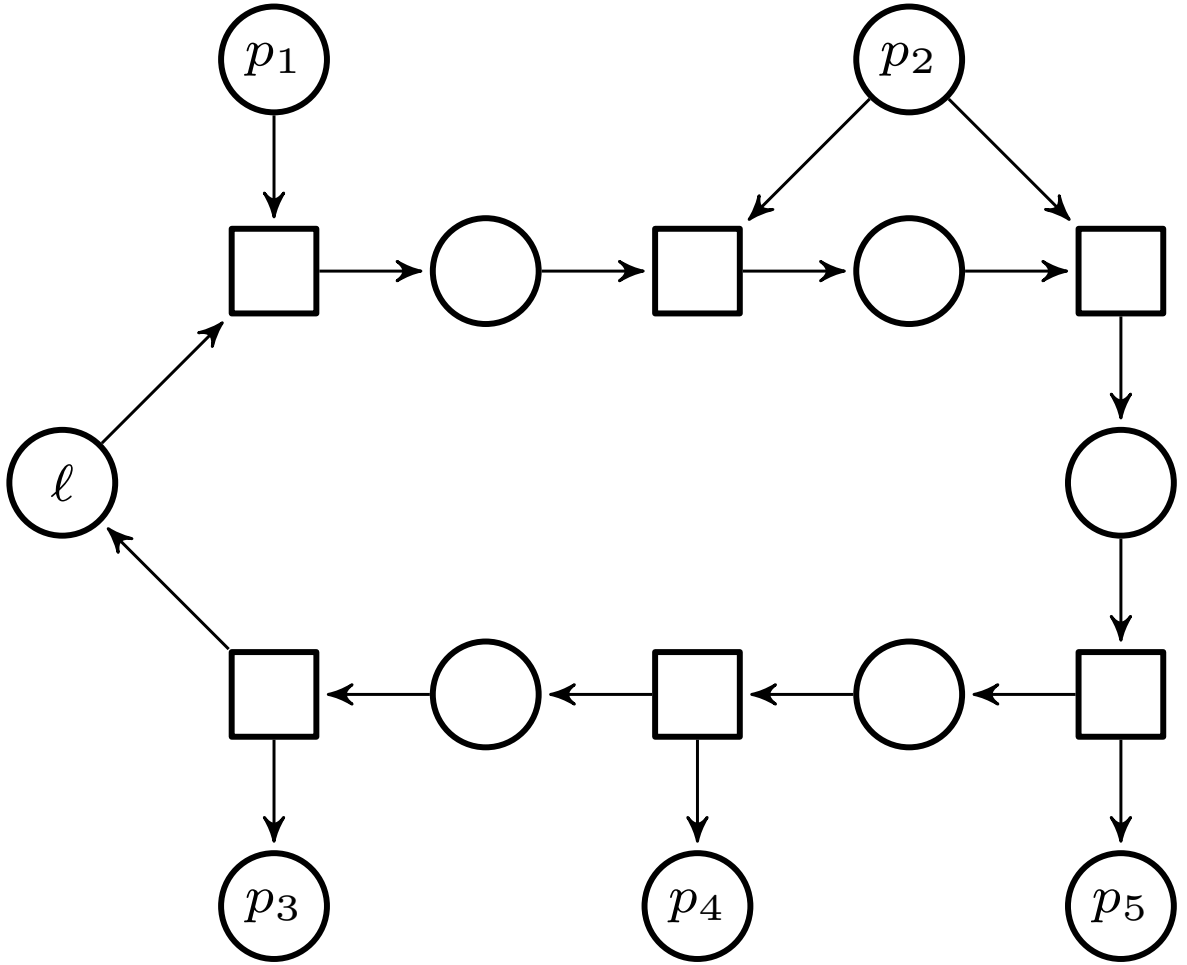
From general PN-reachability to "our" PN-reachability

Does $M \in Reach(N, M_0)$?

Does $M \in Reach(N_0, M_0)$ from some M with $M(\hat{p}) = 0$?

Does $M \in Reach(N_1, M_0 + \boldsymbol{\ell})$ for some M s.t. $M(\hat{p}) = 0$ and $M(P_{aux}) = 0$ and where N_1 is PP like?


p_1
p_2
2
t
p_3
p_4
p_5


p_1
p_2
ℓ
p_3
p_4
p_5

From general PN-reachability to "our" PN-reachability

Does $M \in Reach(N, M_0)$?

Does $M \in Reach(N_0, M_0)$ from some M with $M(\hat{p}) = 0$?

Does $M \in Reach(N_1, M_0 + \boldsymbol{\ell})$ for some M s.t. $M(\hat{p}) = 0$ and $M(P_{aux}) = 0$ and where N_1 is PP like?

From general PN-reachability to "our" PN-reachability

Does $M \in Reach(N, M_0)$?

Does $M \in Reach(N_0, M_0)$ from some M with $M(\hat{p}) = 0$?

Does $M \in Reach(N_1, M_0 + \boldsymbol{\ell})$ for some M s.t. $M(\hat{p}) = 0$ and $M(P_{aux}) = 0$ and where N_1 is PP like?

Does $M \in Reach(N_2, \mathbf{p_0})$ for a M s.t. $M(\{\hat{p}, p_0\} \cup P_{aux}) = 0$?

From general PN-reachability to "our" PN-reachability

Does $M \in Reach(N, M_0)$?

Does $M \in Reach(N_0, M_0)$ from some M with $M(\hat{p}) = 0$?

Does $M \in Reach(N_1, M_0 + \boldsymbol{\ell})$ for some M s.t. $M(\hat{p}) = 0$ and $M(P_{aux}) = 0$ and where N_1 is PP like?

Does $M \in Reach(N_2, \mathbf{p_0})$ for a M s.t. $M(\{\hat{p}, p_0\} \cup P_{aux}) = 0$?

Does $\mathbf{p_0} \in Reach(N_3, M)$ for a M s.t. $M(\{\hat{p}, p_0\} \cup P_{aux}) = 0$?

Population Protocols

Theoretical model with reasonable motivations

- distributed computations among sensor nodes
- "soups" of chemical molecules
- people in social networks
- ...

Does my PP compute a predicate?

Does my PP compute this predicate?

What predicate does my PP compute?

Credits: Javier Esparza  , Rupak Majumdar  , Jerome Leroux 