Parameterized Analysis of Immediate Observation Petri Nets

Chana Weil-Kennedy

joint work with Javier Esparza and Mikhail Raskin

European Research Council

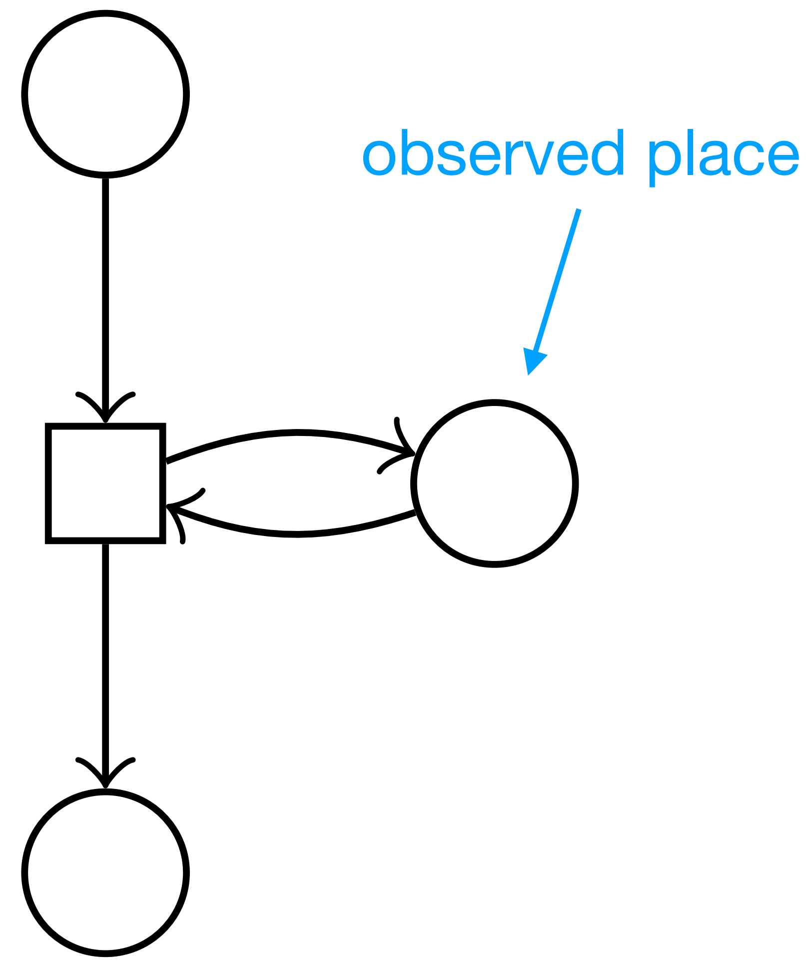
Established by the European Commission

The project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 787367

Immediate Observation Petri Nets

All transitions have

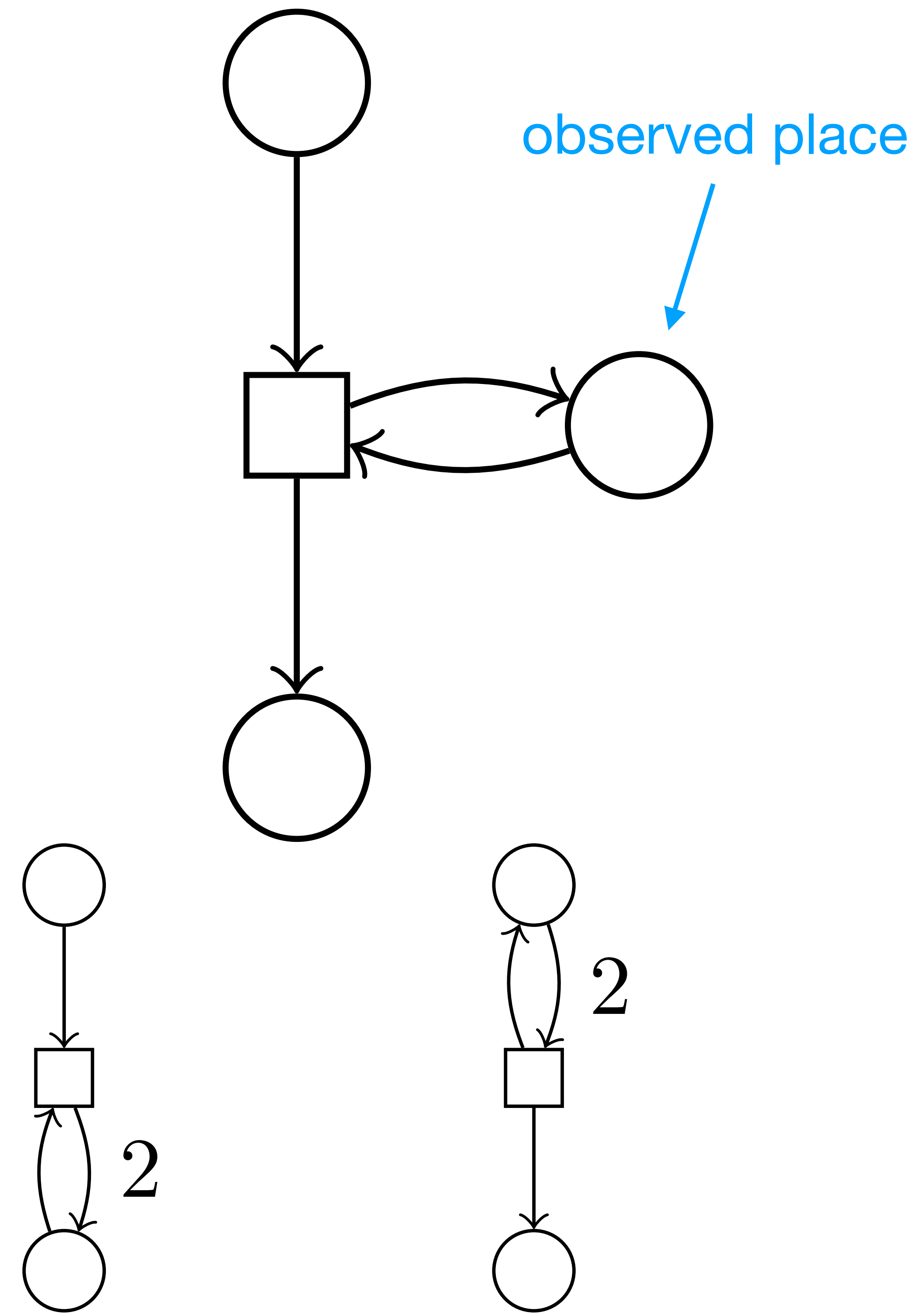
- one input place and one output place, and
- one observed place that is both input and output.



Immediate Observation Petri Nets

All transitions have

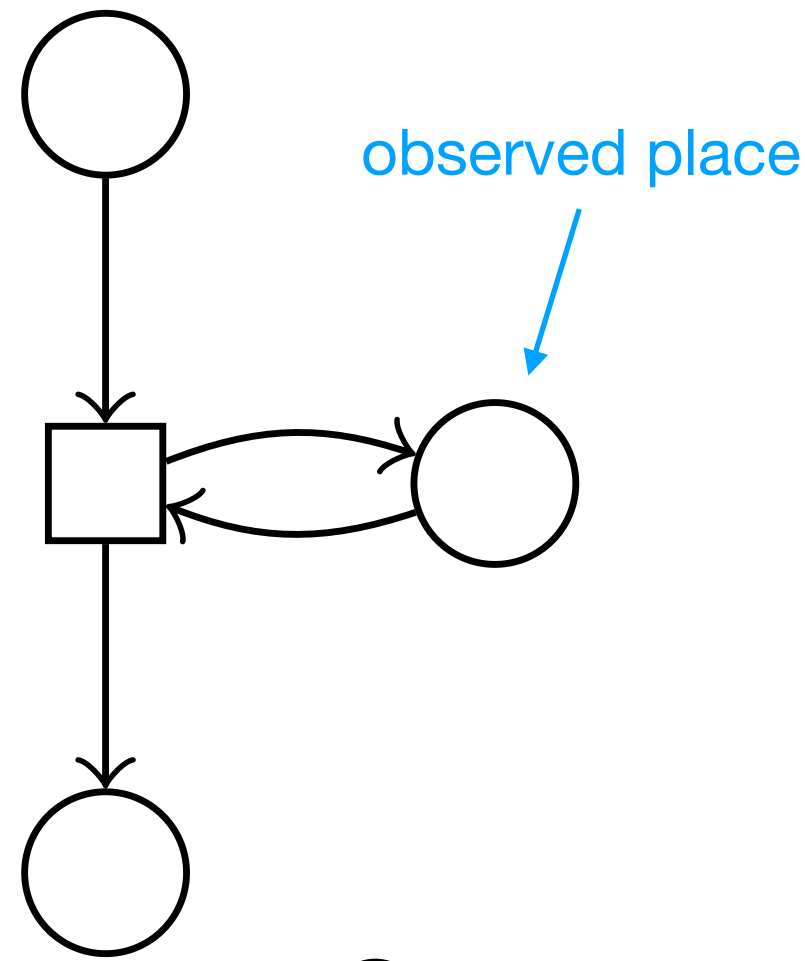
- one input place and one output place, and
- one observed place that is both input and output.



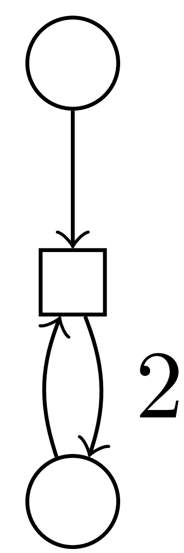
Immediate Observation Petri Nets

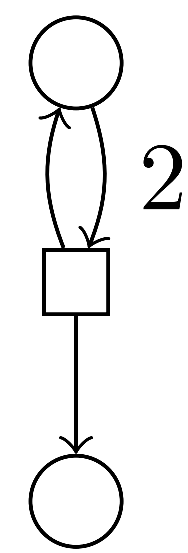
All transitions have

- one input place and one output place, and
- one observed place that is both input and output.

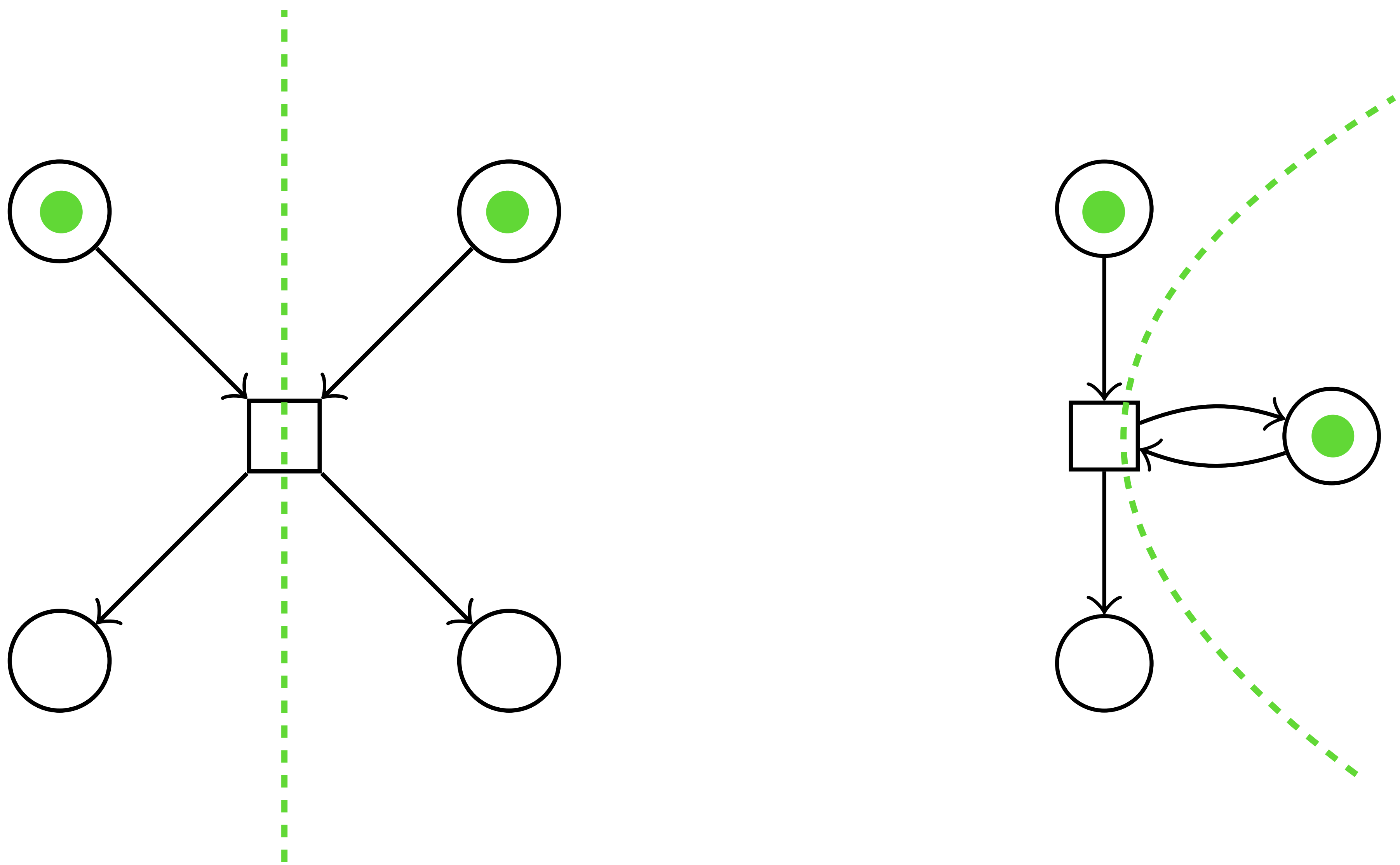


IO nets are **conservative**.

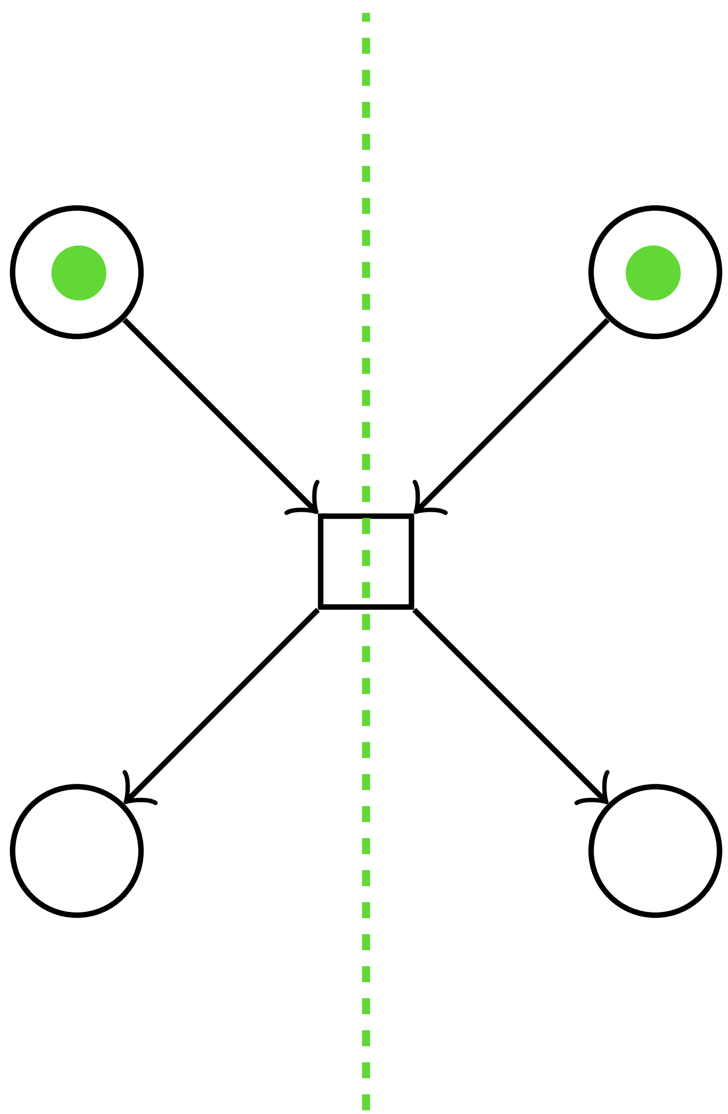




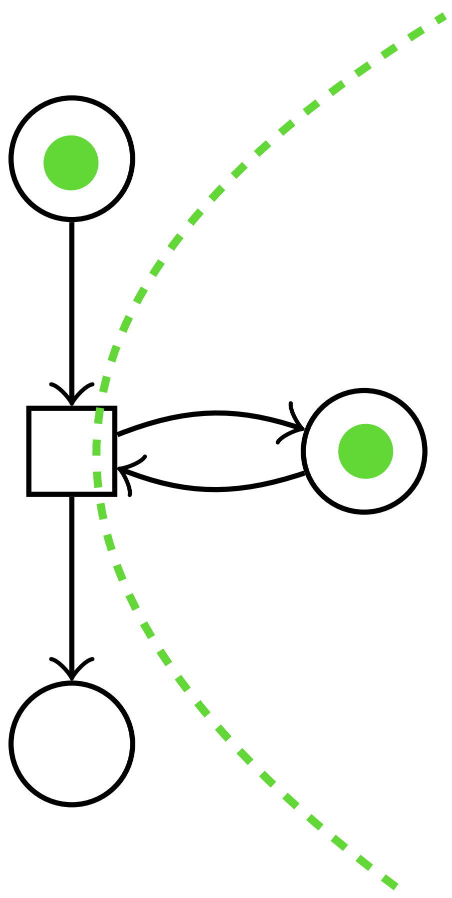
Why “Immediate Observation”?



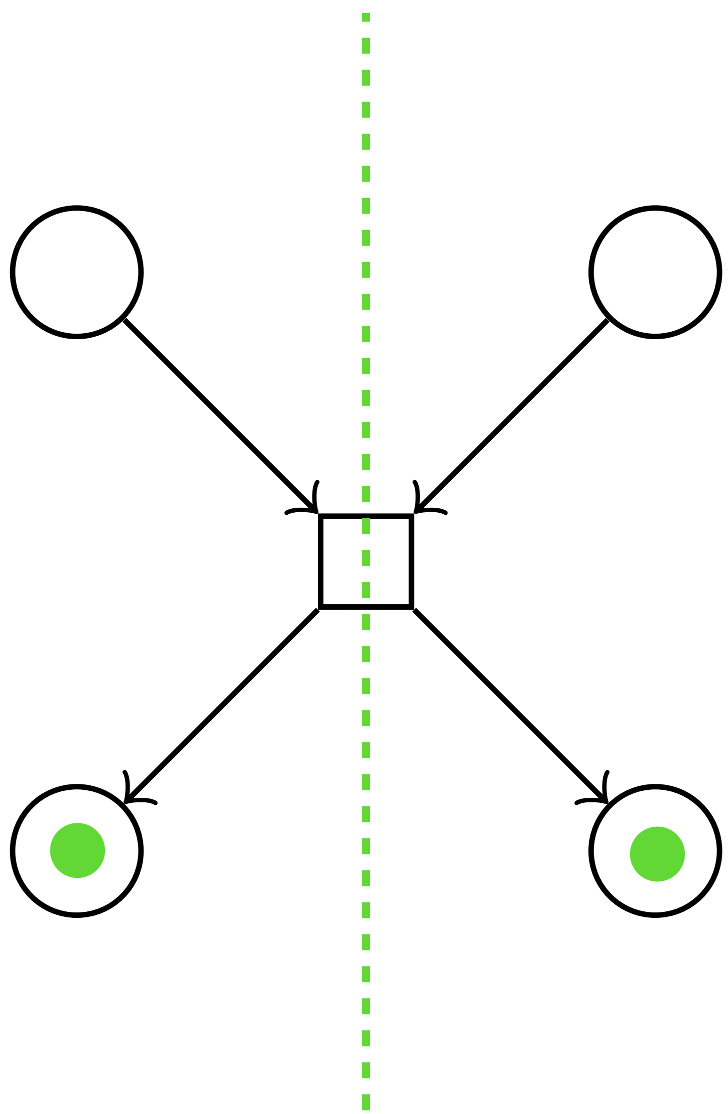
Why “Immediate Observation”?



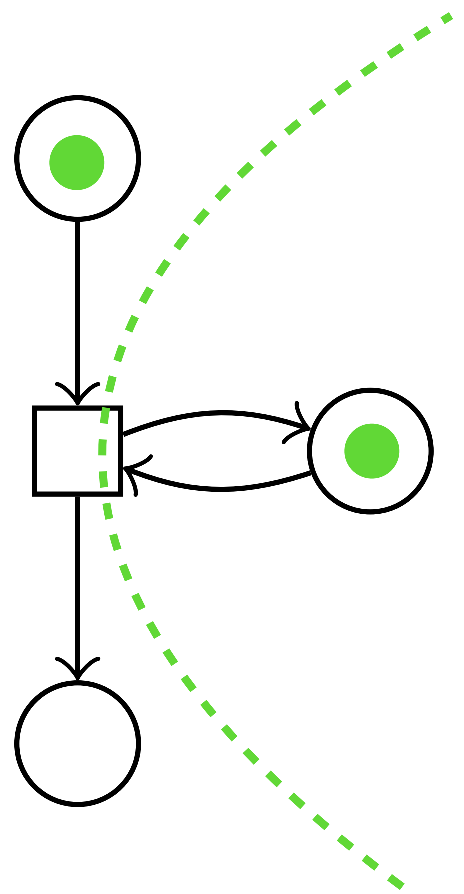
Processes interact and change
their states simultaneously



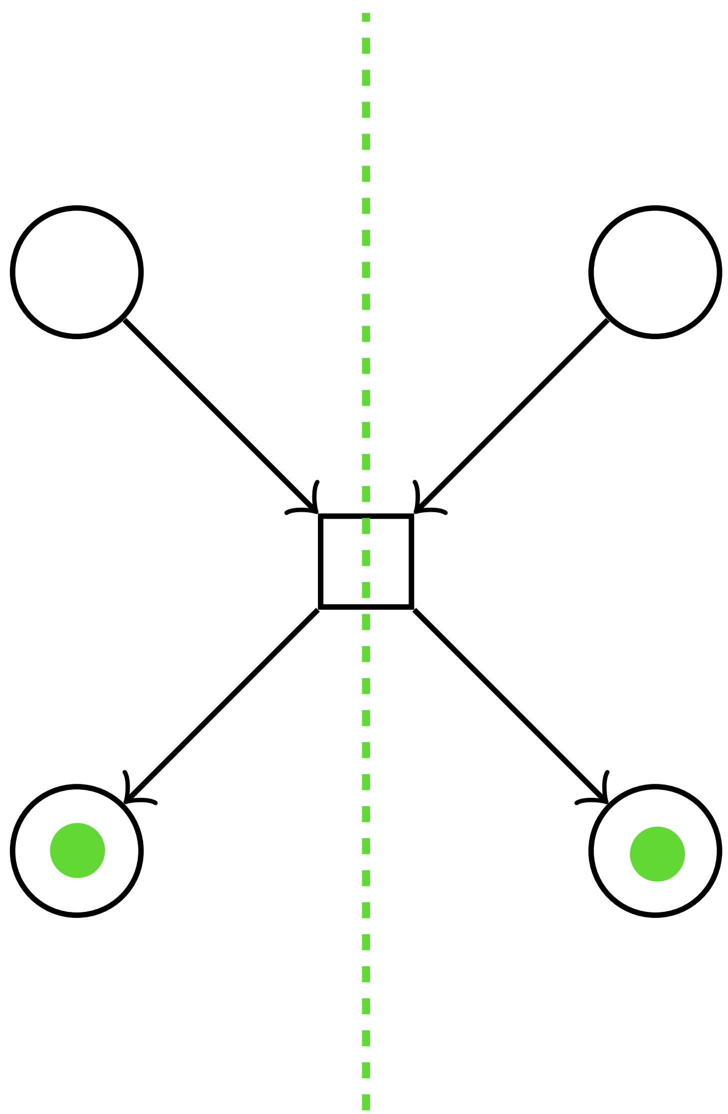
Why “Immediate Observation”?



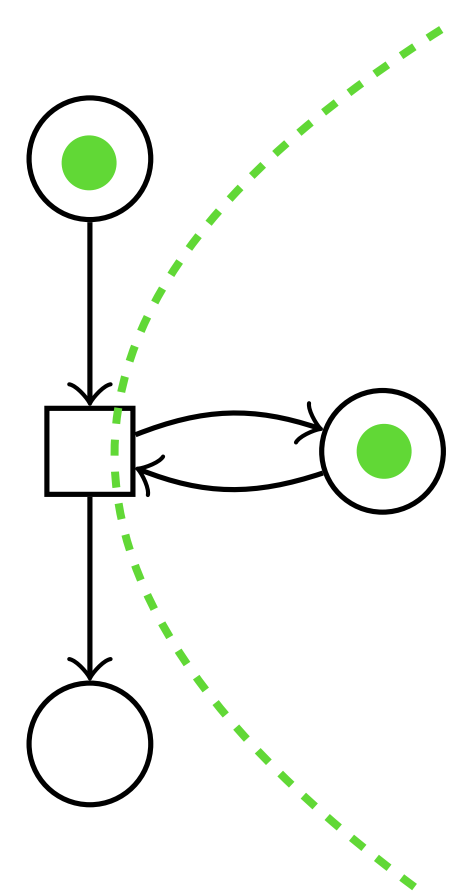
Processes interact and change
their states simultaneously



Why “Immediate Observation”?

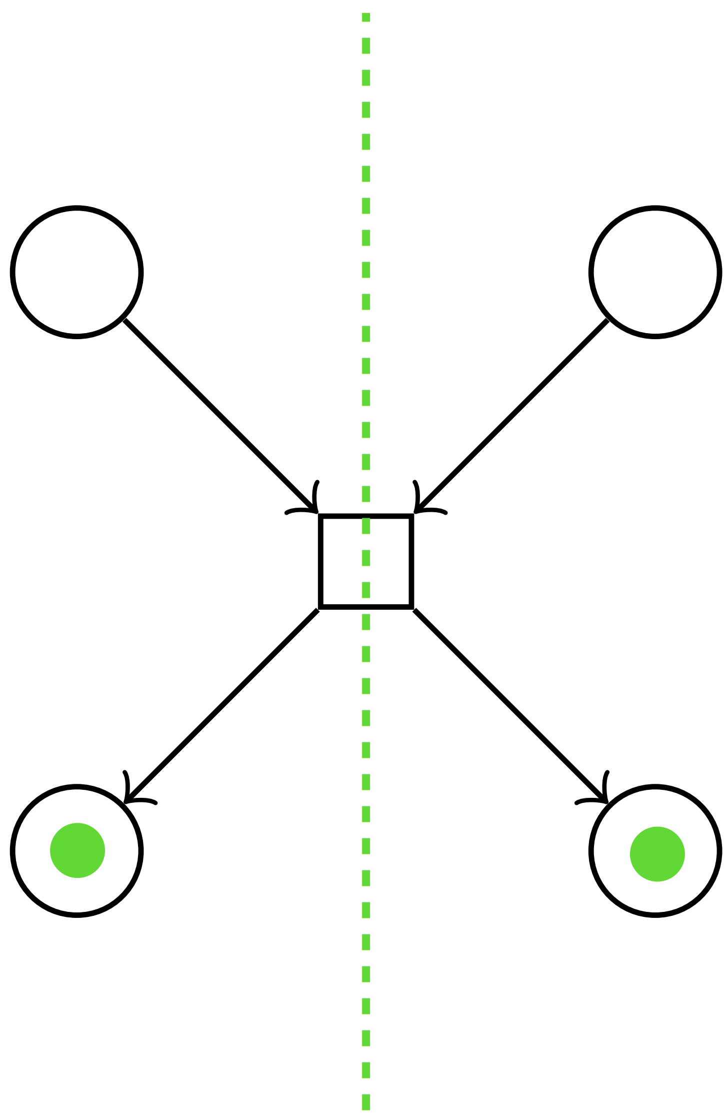


Processes interact and change
their states simultaneously

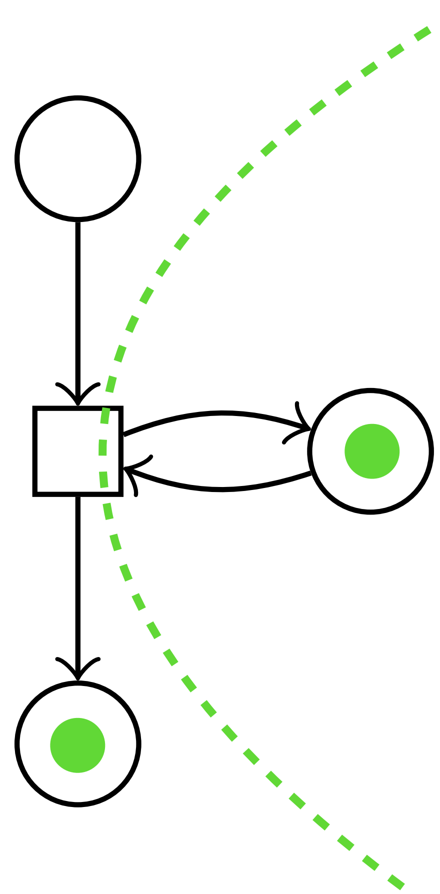


One process **observes** the state of another
and
changes its own state **immediately**
(in one atomic action)

Why “Immediate Observation”?



Processes interact and change their states simultaneously



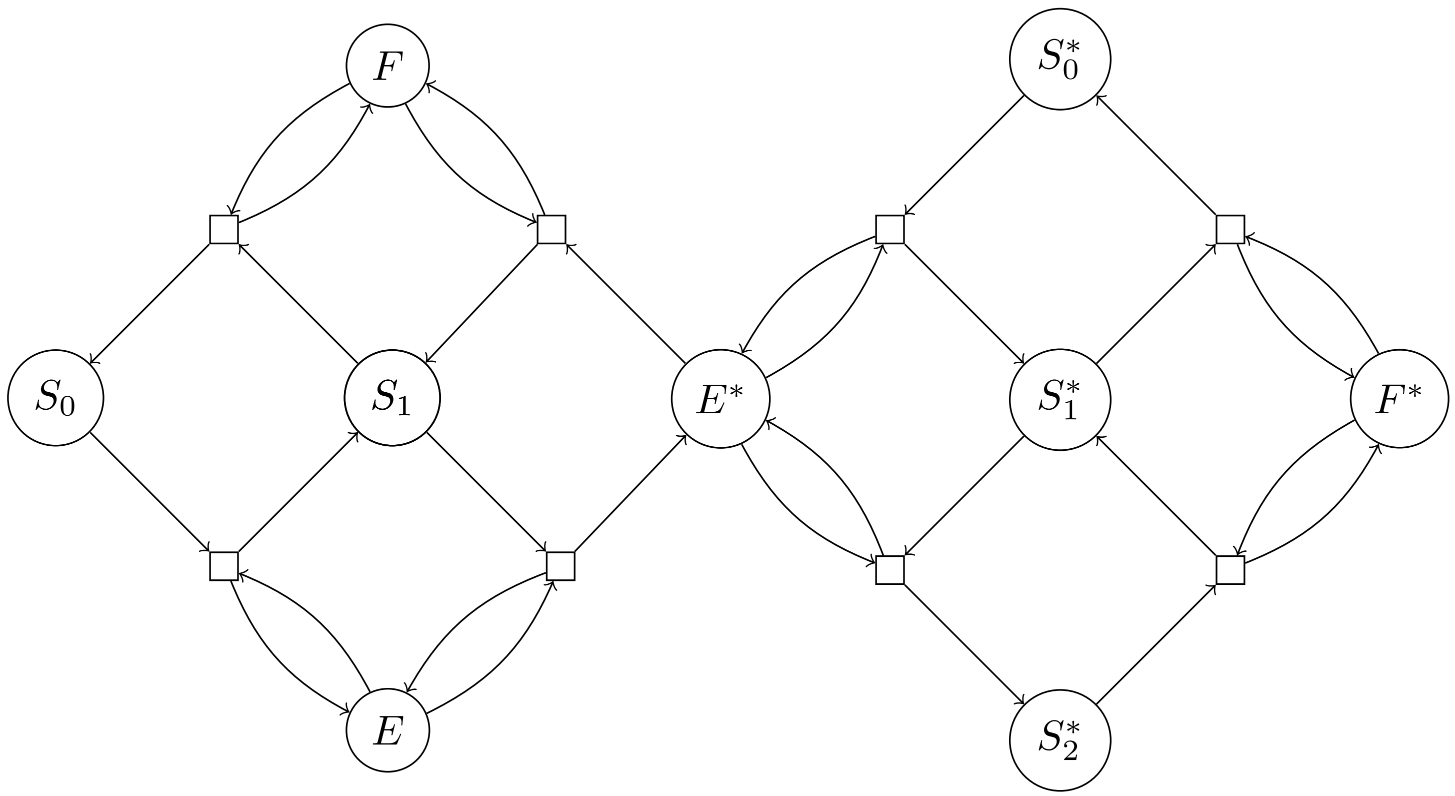
One process **observes** the state of another
and
changes its own state **immediately**
(in one atomic action)

Application: Chemical Reaction Networks

enzyme → E, reactant → R, product → P

Enzymatic catalysis: $E + R \longrightarrow E + P$

$$E + S_0 \rightarrow E + S_1 \rightarrow E + E^*$$
$$F + E^* \rightarrow F + S_1 \rightarrow F + S_0$$
$$E^* + S_0^* \rightarrow E^* + S_1^* \rightarrow E^* + S_2^*$$
$$F^* + S_2^* \rightarrow F^* + S_1^* \rightarrow F^* + S_0^*$$



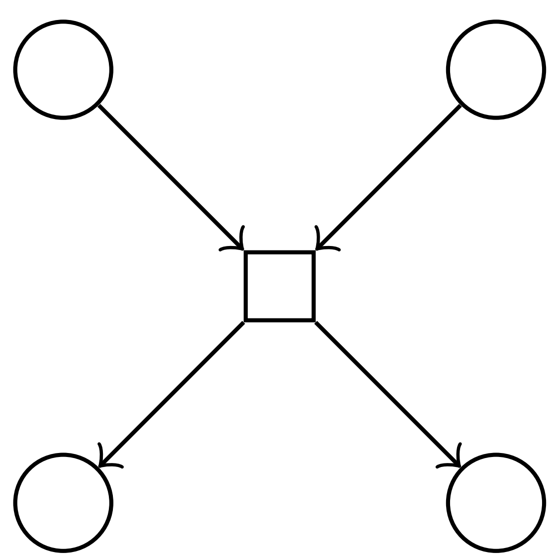
[A Petri net approach to the study of persistence in chemical reaction networks, Angeli et al., ’06]

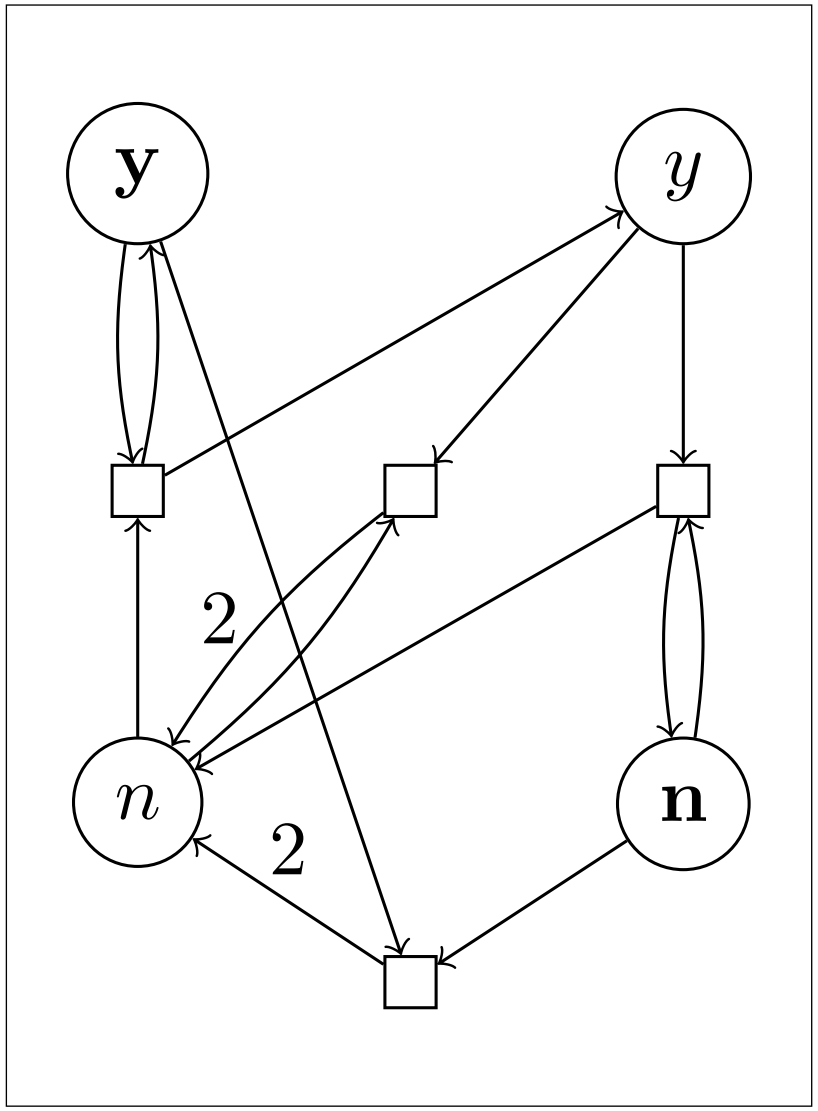
Application: Population Protocols

[Angluin et al.*, '04]*

Distributed computing model where identical finite-state mobile agents jointly compute a function.

Agents communicate through *rendez-vous*.





[The computational power of population protocols, Angluin et al.,'06]

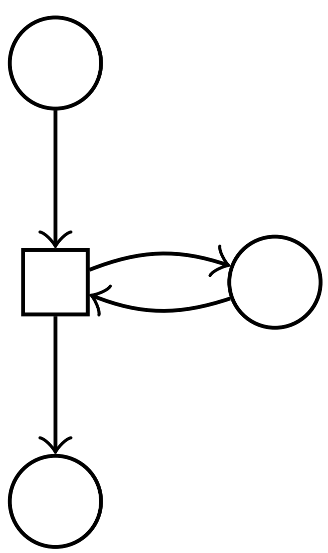
Application: Population Protocols

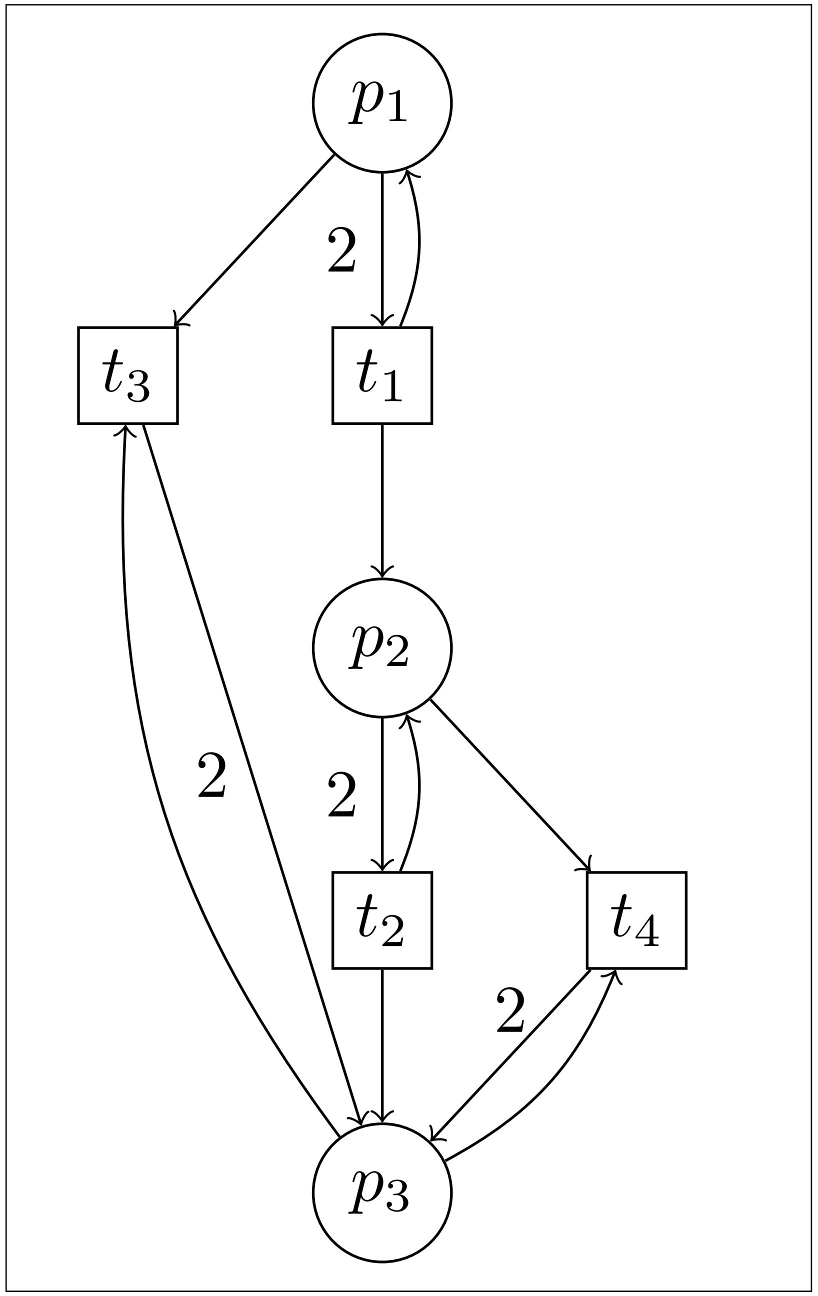
Distributed computing model for identical finite-state mobile agents.

Immediate observation population protocols:

An agent *observes* an other agent's state and updates its own based on this information.

Introduced to model sensor networks.

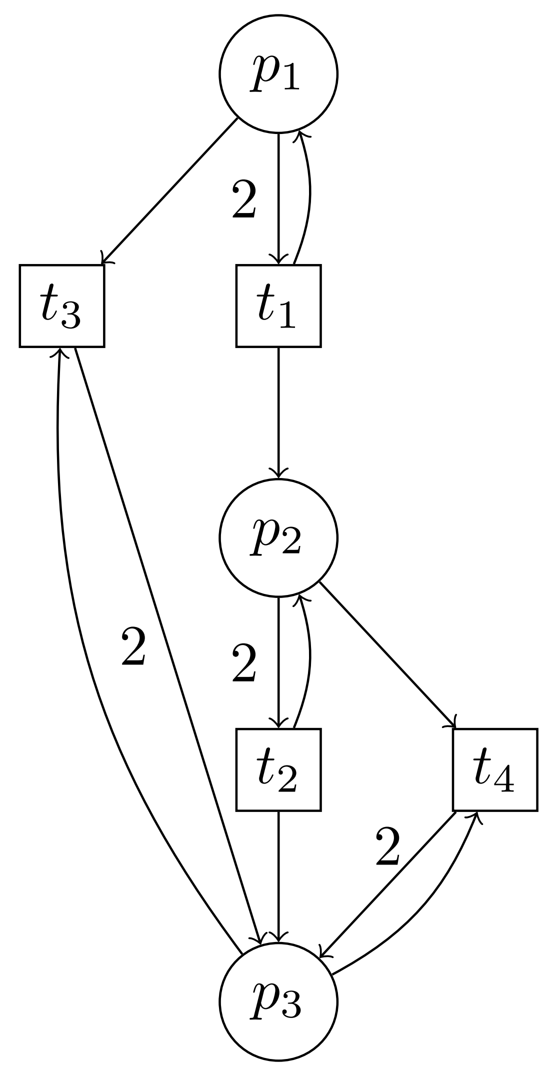




[The computational power of population protocols, Angluin et al.,'06]

Parameterized Problems

In these application domains we are interested in *parameterized* problems.

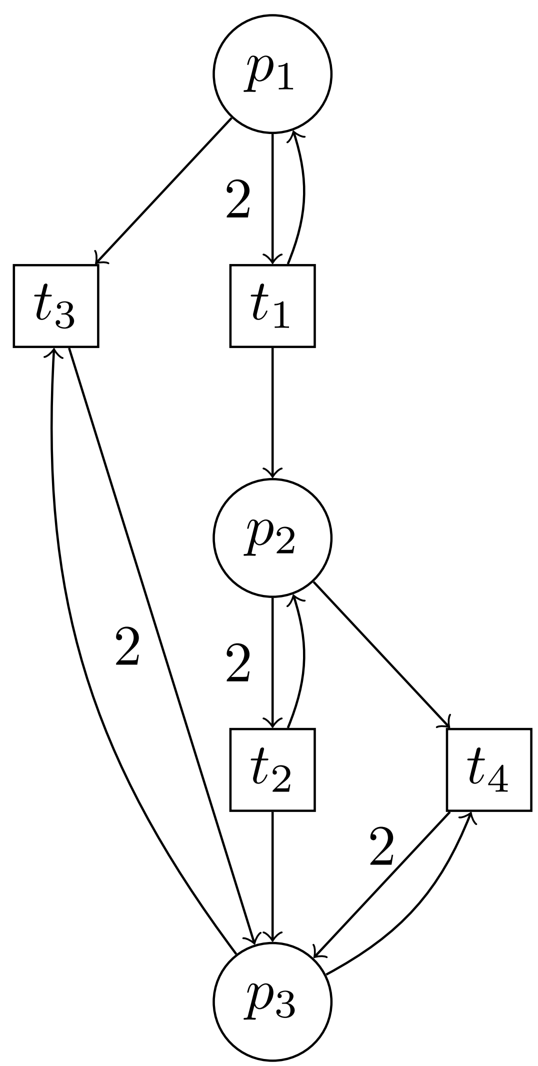


[Angluin et al.*,'06]*

Parameterized Problems

In these application domains we are interested in *parameterized* problems.

- Goal of a protocol: compute a function $f : \mathbb{N}^k \to \{0,1\}$
- Protocol: Petri net N. Input: initial marking M_0.
- Correctness: for every initial marking M_0, the Petri net (N, M_0) "computes" $f(M_0)$.

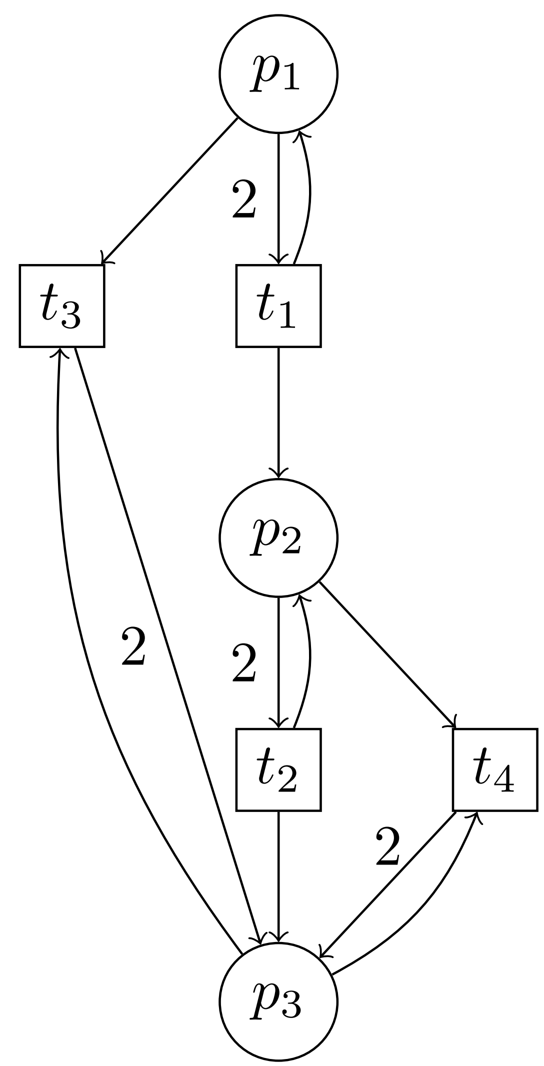


[Angluin et al.*,'06]*

Parameterized Problems

In these application domains we are interested in *parameterized* problems.

- Goal of a protocol: compute a function $f : \mathbb{N}^k \rightarrow \{0,1\}$
- Protocol: Petri net N. Input: initial marking M_0.
- Correctness: for every initial marking M_0, the Petri net (N, M_0) "computes" $f(M_0)$.



[Angluin et al.*,'06]*

Initial markings: $M_0^{(n)} = n \cdot p_1$

parameter

This protocol is correct if and only if for every initial marking $M_0^{(n)}$:

- $n \geq 3 \Rightarrow$ all markings reachable from $M_0^{(n)}$ can reach the marking with all tokens in p_3.
- $n < 3 \Rightarrow$ there is no reachable marking with a token in p_3.

Counting Constraints

We consider infinite sets of markings defined by counting constraints.

number of tokens in p_2

lower bound

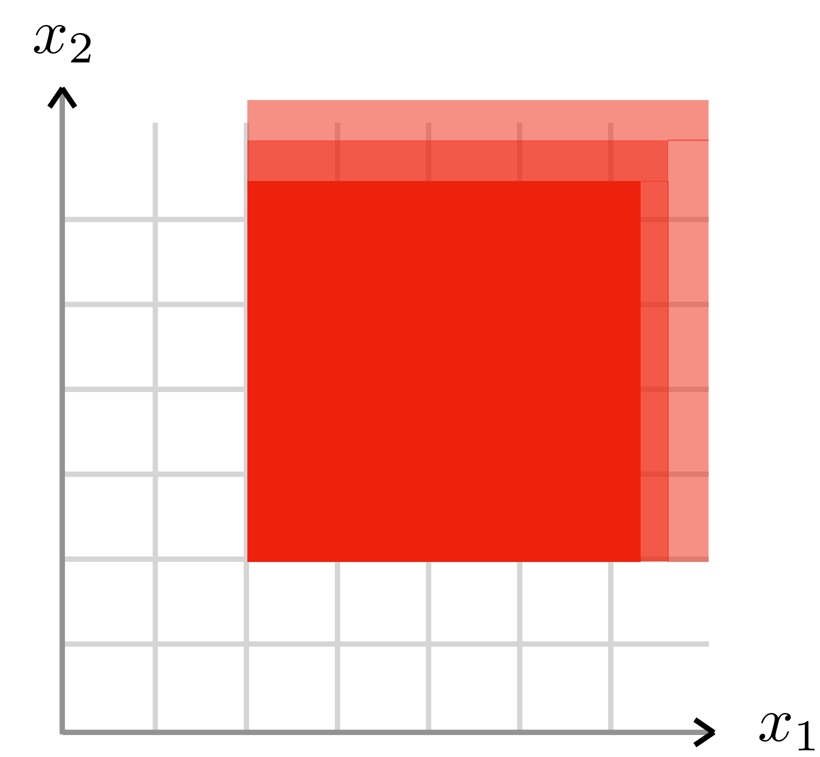
upper bound

- An expression $2 \leq x_2 \leq 5$ is an *atomic bound.*

Counting Constraints

We consider infinite sets of markings defined by counting constraints.

e.g. in a Petri net with two places p_1 and p_2



$2 \leq x_1 \leq \infty \wedge$
$2 \leq x_2 \leq \infty$

number of tokens in p_2

lower bound

upper bound

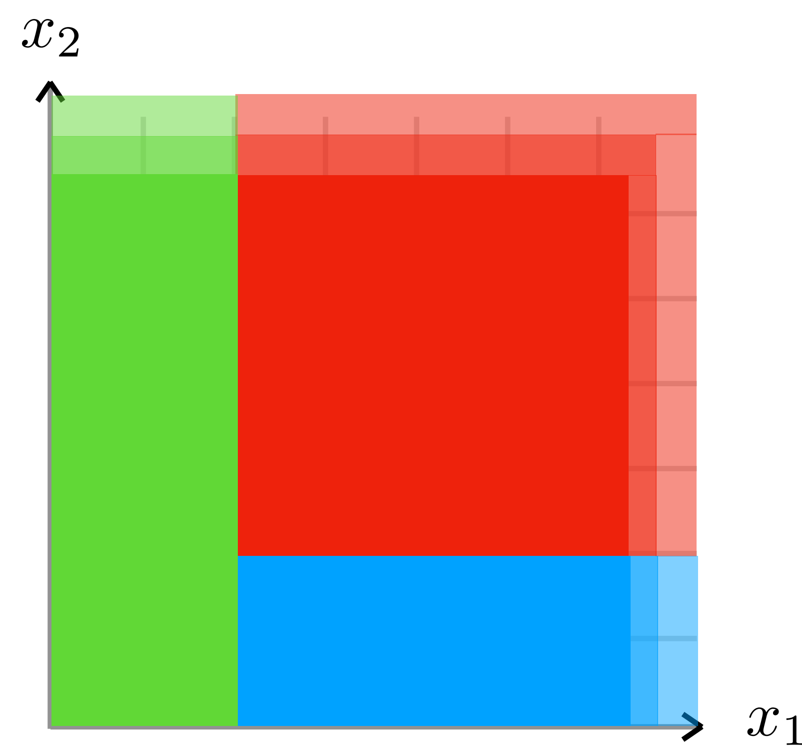
- An expression $2 \leq x_2 \leq 5$ is an *atomic bound.*

- *Counting constraints* are boolean combinations of atomic bounds.

Counting Constraints

We consider infinite sets of markings defined by counting constraints.

e.g. in a Petri net with two places p_1 and p_2



$$\neg \begin{array}{l} 2 \leq x_1 \leq \infty \wedge \\ 2 \leq x_2 \leq \infty \end{array}$$

$$\begin{array}{l} 2 \leq x_1 \leq \infty \wedge \\ 0 \leq x_2 \leq 2 \end{array} \vee \begin{array}{l} 0 \leq x_1 \leq 2 \wedge \\ 0 \leq x_2 \leq \infty \end{array}$$

number of tokens in p_2

lower bound

upper bound

- An expression $2 \leq x_2 \leq 5$ is an *atomic bound.*
- *Counting constraints* are boolean combinations of atomic bounds.

Parameterized Reachability and Coverability

INPUT: An IO net N, and two sets of markings S and S' described by **counting constraints**.

Parameterized Reachability

QUESTION: Are there markings $M \in S$ and $M' \in S'$ such that M' is **reachable** from M in net N ?

Parameterized Reachability and Coverability

INPUT: An IO net *N*, and two sets of markings *S* and *S'* described by **counting constraints**.

Parameterized Reachability

QUESTION: Are there markings $M \in S$ and $M' \in S'$ such that M' is **reachable** from *M* in net *N* ?

Parameterized Coverability

QUESTION: Are there markings $M \in S$ and $M' \in S'$ such that M' is **coverable** by *M* in net *N* ?

Results

	Conservative Petri nets	**Immediate Observation nets**
Reachability	PSPACE-complete	
Coverability	PSPACE-complete	
Param-Reachability		
Param-Coverability		

Results

	Conservative Petri nets	**Immediate Observation nets**
Reachability	PSPACE-complete	**PSPACE-complete**
Coverability	PSPACE-complete	**PSPACE-complete**
Param-Reachability		
Param-Coverability		

Results

	Conservative Petri nets	**Immediate Observation nets**
Reachability	PSPACE-complete	**PSPACE-complete**
Coverability	PSPACE-complete	**PSPACE-complete**
Param-Reachability	TOWER-hard (non-elementary)	
Param-Coverability	EXPSPACE-hard	

Results

	Conservative Petri nets	**Immediate Observation nets**
Reachability	PSPACE-complete	**PSPACE-complete**
Coverability	PSPACE-complete	**PSPACE-complete**
Param-Reachability	TOWER-hard (non-elementary)	**PSPACE-complete**
Param-Coverability	EXPSPACE-hard	**PSPACE-complete**

Results

	Conservative Petri nets	**Immediate Observation nets**
Reachability	PSPACE-complete	**PSPACE-complete**
Coverability	PSPACE-complete	**PSPACE-complete**
Param-Reachability	TOWER-hard (non-elementary)	**PSPACE-complete**
Param-Coverability	EXPSPACE-hard	**PSPACE-complete**

Compared to arbitrary conservative Petri nets, IO nets

- don't lose expressivity, and
- do much better for parameterized problems: deciding for infinitely many markings is not harder than for a single marking!

Application: Correctness of IOPP

[Angluin, Aspnes, Diamada, Fischer, Peralta, '04]

General population protocols

[Angluin, Aspnes, Eisenstat, Ruppert, '07]

Immediate observation population protocols

Application: Correctness of IOPP

[Angluin, Aspnes, Diamada, Fischer, Peralta, '04]

General population protocols

[Angluin, Aspnes, Eisenstat, Ruppert, '07]

Immediate observation population protocols

[Esparza, Ganty, Leroux, Majumdar, '15]

Correctness is reducible to
reachability for Petri nets;
decidable and **TOWER-hard**

Application: Correctness of IOPP

[Angluin, Aspnes, Diamada, Fischer, Peralta, '04]

General population protocols

↓

[Esparza, Ganty, Leroux, Majumdar, '15]

Correctness is reducible to reachability for Petri nets; **decidable** and **TOWER-hard**

[Angluin, Aspnes, Eisenstat, Ruppert, '07]

Immediate observation population protocols

↓

[Esparza, Ganty, Majumdar, Weil-Kennedy, '18]

Correctness is **PSPACE-hard** and in **EXPSPACE**

Application: Correctness of IOPP

[Angluin, Aspnes, Diamada, Fischer, Peralta, '04]

General population protocols

↓

[Esparza, Ganty, Leroux, Majumdar, '15]

Correctness is reducible to reachability for Petri nets; **decidable** and **TOWER-hard**

[Angluin, Aspnes, Eisenstat, Ruppert, '07]

Immediate observation population protocols

↓

[Esparza, Ganty, Majumdar, Weil-Kennedy, '18]

Correctness is **PSPACE-hard** and in **EXPSPACE**

↓

[This paper]

Correctness is **PSPACE-complete**

Main Theorem

Theorem

For *N* an IO net with *n* places, for Γ a counting constraint describing a set *S*,

1. there exist counting constraints for *pre*(S)* and *post*(S)*
2. the size of these counting constraints is $\leq size(\Gamma) + n^3$

Main Theorem

Theorem

For *N* an IO net with *n* places, for Γ a counting constraint describing a set *S*,

1. there exist counting constraints for *pre*(S)* and *post*(S)*
2. the size of these counting constraints is $\leq size(\Gamma) + n^3$

essentially the largest finite bound
of the counting constraint

Main Theorem

In arbitrary conservative nets, 1. is not always true.

Theorem

For *N* an IO net with *n* places, for Γ a counting constraint describing a set *S*,

1. there exist counting constraints for *pre*(S)* and *post*(S)*
2. the size of these counting constraints is $\leq size(\Gamma) + n^3$

Main Theorem

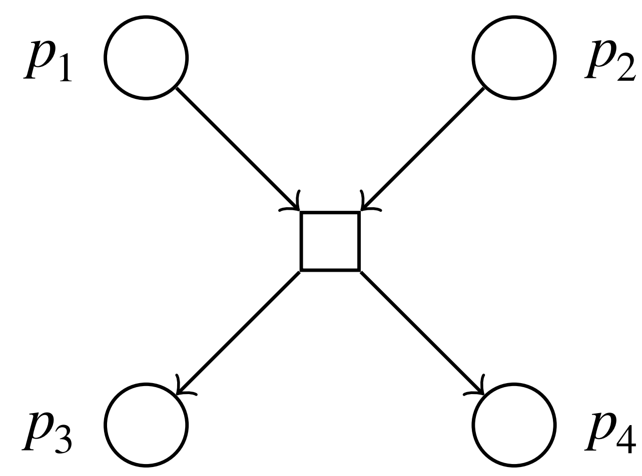
Theorem

For *N* an IO net with *n* places, for Γ a counting constraint describing a set *S*,

1. there exist counting constraints for *pre*(S)* and *post*(S)*
2. the size of these counting constraints is $\leq size(\Gamma) + n^3$

In arbitrary conservative nets, 1. is not always true.

For example, $S = \begin{pmatrix} 1 \leq p_1 \leq \infty \wedge \\ 1 \leq p_2 \leq \infty \wedge \\ 0 \leq p_3 \leq 0 \wedge \\ 0 \leq p_4 \leq 0 \end{pmatrix}$



Main Theorem

In arbitrary conservative nets, 1. is not always true.

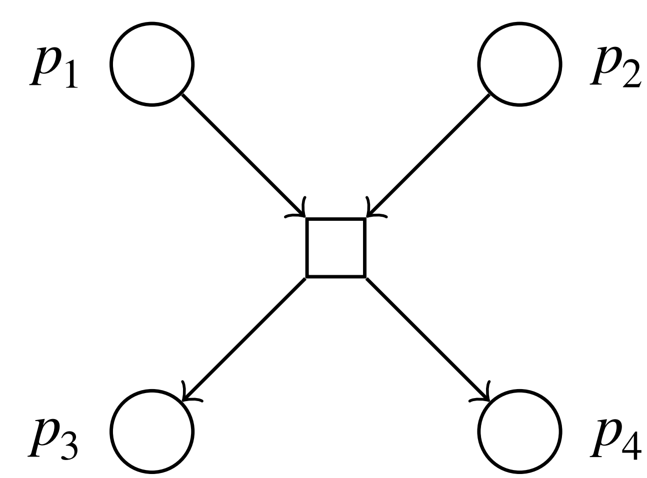
Theorem

For N an IO net with n places, for Γ a counting constraint describing a set S,

1. there exist counting constraints for *pre*(S)* and *post*(S)*
2. the size of these counting constraints is $\leq size(\Gamma) + n^3$

For example, $S = \begin{pmatrix} 1 \leq p_1 \leq \infty \wedge \\ 1 \leq p_2 \leq \infty \wedge \\ 0 \leq p_3 \leq 0 \wedge \\ 0 \leq p_4 \leq 0 \end{pmatrix}$

$$post^*(S) = \begin{pmatrix} 0 \leq p_1 \leq \infty \wedge \\ 0 \leq p_2 \leq \infty \wedge \\ p_3 = p_4 \end{pmatrix}$$



Main Theorem

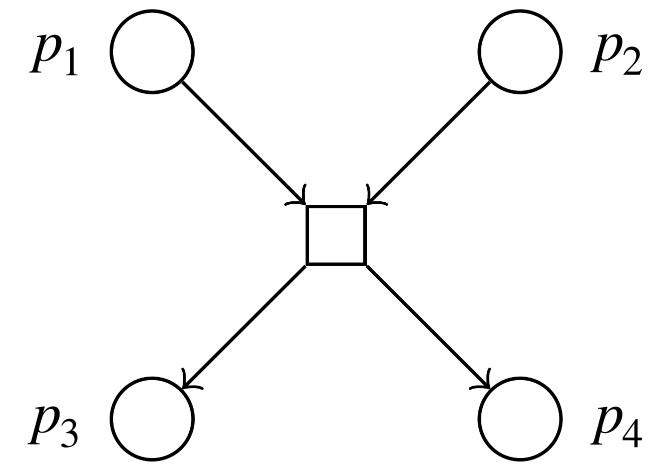
In arbitrary conservative nets, 1. is not always true.

Theorem

For N an IO net with n places, for Γ a counting constraint describing a set S,

1. there exist counting constraints for *pre*(S)* and *post*(S)*
2. the size of these counting constraints is $\leq size(\Gamma) + n^3$

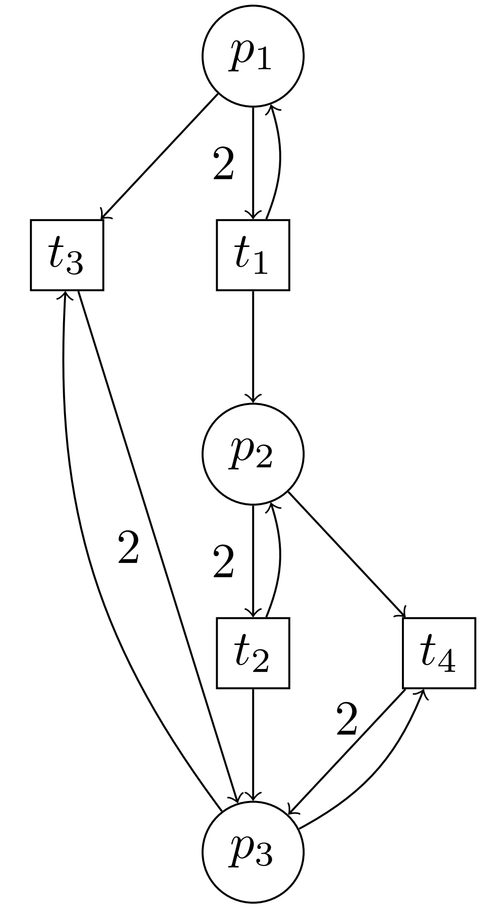
For example, $S = \begin{pmatrix} 1 \leq p_1 \leq \infty \wedge \\ 1 \leq p_2 \leq \infty \wedge \\ 0 \leq p_3 \leq 0 \wedge \\ 0 \leq p_4 \leq 0 \end{pmatrix}$



$$post^*(S) = \begin{pmatrix} 0 \leq p_1 \leq \infty \wedge \\ 0 \leq p_2 \leq \infty \wedge \\ p_3 = p_4 \end{pmatrix}$$

Not a counting constraint!

Applying the Main Theorem

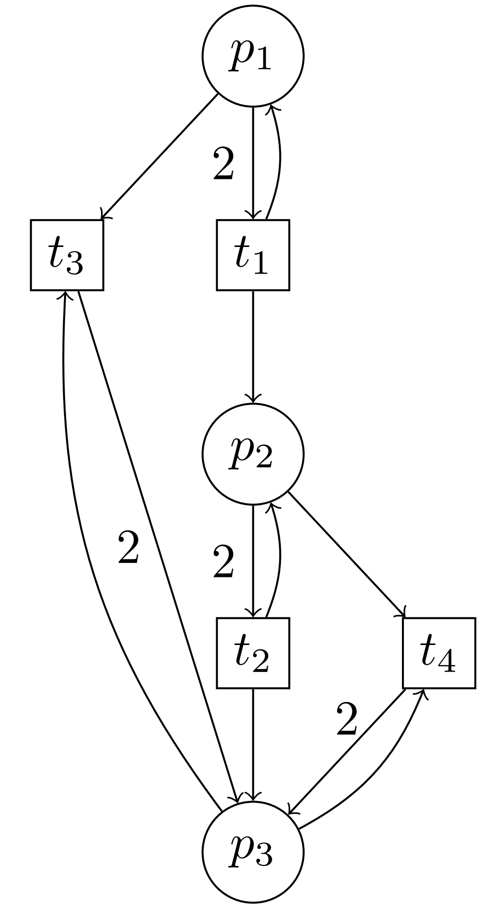


By the Main Theorem, we know we can apply the following algorithm to counting constraint S to obtain $post^*(S)$.

$$\begin{pmatrix} 3 \le p_1 \le \infty \wedge \\ 0 \le p_2 \le 0 \wedge \\ 0 \le p_3 \le 0 \end{pmatrix}$$

S

Applying the Main Theorem

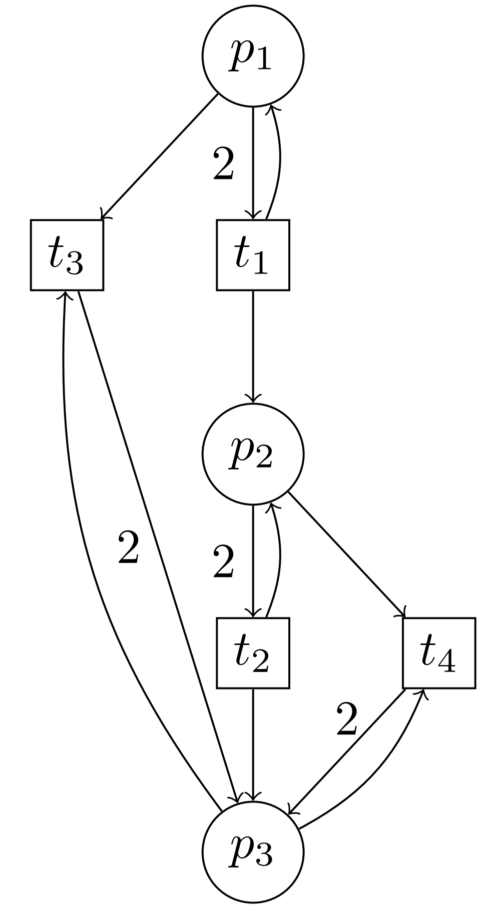


By the Main Theorem, we know we can apply the following algorithm to counting constraint S to obtain $post^*(S)$.

$$\begin{pmatrix} 3 \leq p_1 \leq \infty \wedge \\ 0 \leq p_2 \leq 0 \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix}$$

$S = S_0$

Applying the Main Theorem

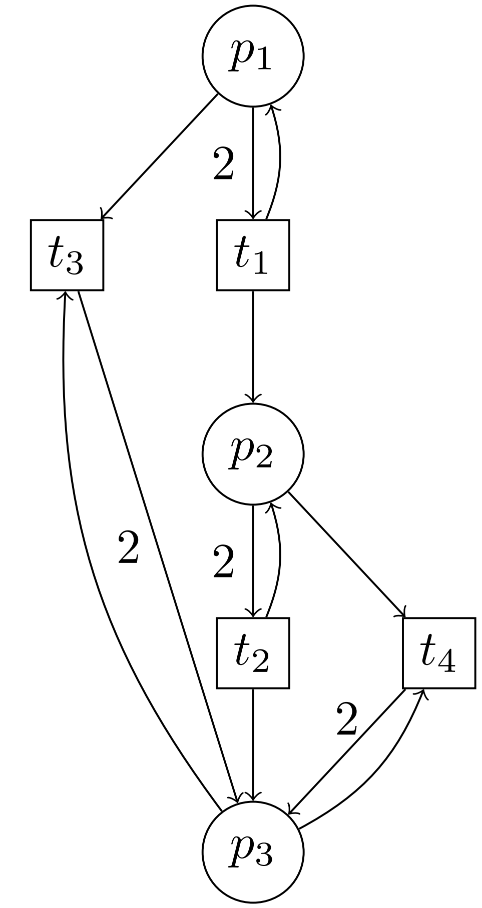


By the Main Theorem, we know we can apply the following algorithm to counting constraint S to obtain $post^*(S)$.

$$\begin{pmatrix} 3 \leq p_1 \leq \infty \wedge \\ 0 \leq p_2 \leq 0 \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix} \xrightarrow{t_1^*} S_0 \vee \begin{pmatrix} 2 \leq p_1 \leq \infty \wedge \\ 1 \leq p_2 \leq \infty \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix} \vee \begin{pmatrix} 1 \leq p_1 \leq \infty \wedge \\ 2 \leq p_2 \leq \infty \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix}$$

$S = S_0$

Applying the Main Theorem

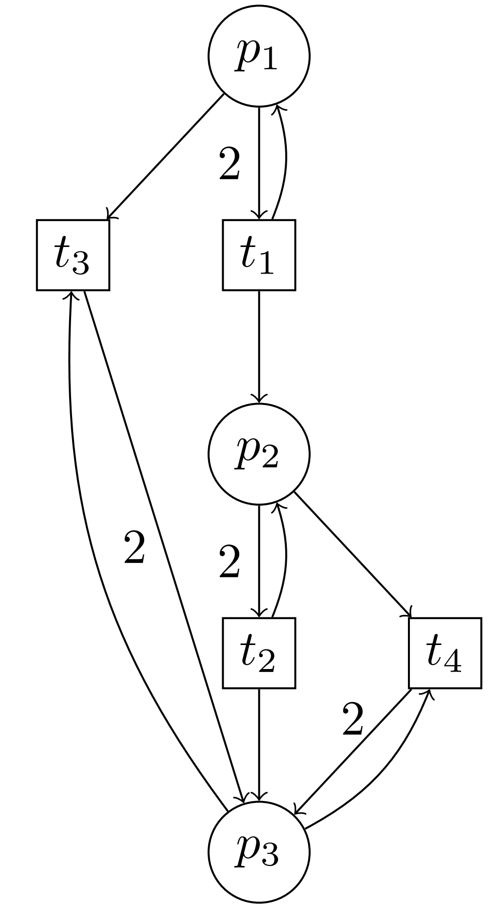


By the Main Theorem, we know we can apply the following algorithm to counting constraint S to obtain $post^*(S)$.

$$\begin{pmatrix} 3 \leq p_1 \leq \infty \wedge \\ 0 \leq p_2 \leq 0 \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix} \xrightarrow{t_1^*} S_0 \vee \begin{pmatrix} 2 \leq p_1 \leq \infty \wedge \\ 1 \leq p_2 \leq \infty \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix} \vee \begin{pmatrix} 1 \leq p_1 \leq \infty \wedge \\ 2 \leq p_2 \leq \infty \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix}$$

$S = S_0$ $\qquad$ S_1

Applying the Main Theorem

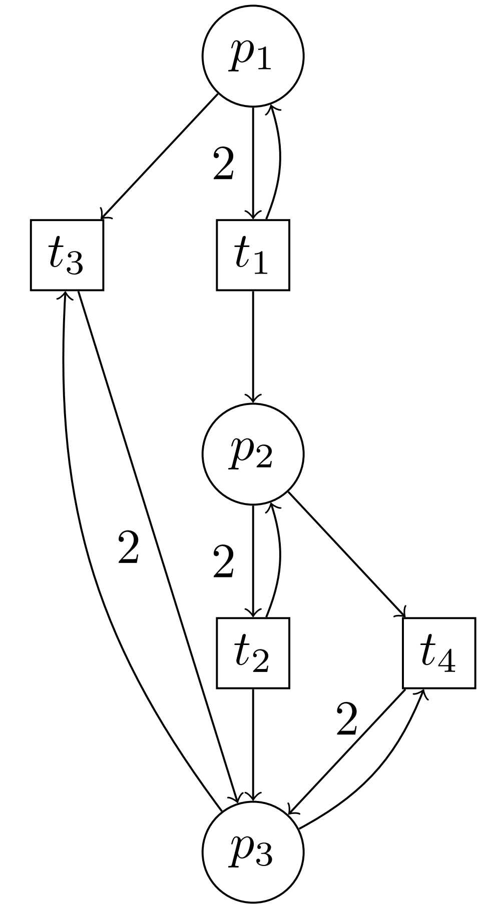


By the Main Theorem, we know we can apply the following algorithm to counting constraint S to obtain $post^*(S)$.

$$\begin{pmatrix} 3 \le p_1 \le \infty \wedge \\ 0 \le p_2 \le 0 \wedge \\ 0 \le p_3 \le 0 \end{pmatrix} \xrightarrow{t_1^*} S_0 \vee \begin{pmatrix} 2 \le p_1 \le \infty \wedge \\ 1 \le p_2 \le \infty \wedge \\ 0 \le p_3 \le 0 \end{pmatrix} \vee \begin{pmatrix} 1 \le p_1 \le \infty \wedge \\ 2 \le p_2 \le \infty \wedge \\ 0 \le p_3 \le 0 \end{pmatrix} \xrightarrow{t_1^*} S_1$$

$S = S_0$ $\qquad$ S_1

Applying the Main Theorem



By the Main Theorem, we know we can apply the following algorithm to counting constraint S to obtain $post^*(S)$.

$$S = S_0: \quad \begin{pmatrix} 3 \le p_1 \le \infty \wedge \\ 0 \le p_2 \le 0 \wedge \\ 0 \le p_3 \le 0 \end{pmatrix}$$

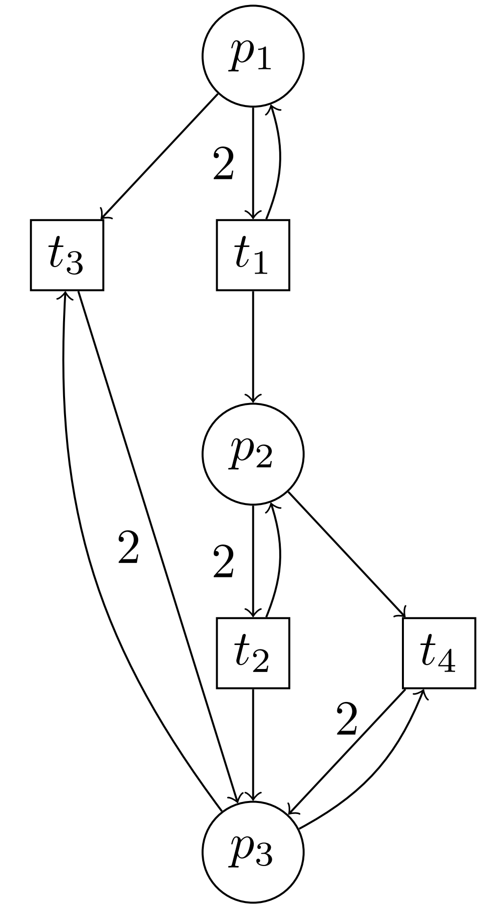
$$\xrightarrow{t_1^*}$$

$$S_1: \quad S_0 \vee \begin{pmatrix} 2 \le p_1 \le \infty \wedge \\ 1 \le p_2 \le \infty \wedge \\ 0 \le p_3 \le 0 \end{pmatrix} \vee \begin{pmatrix} 1 \le p_1 \le \infty \wedge \\ 2 \le p_2 \le \infty \wedge \\ 0 \le p_3 \le 0 \end{pmatrix}$$

$$\xrightarrow{t_1^*} S_1$$

$$\xrightarrow{t_2^*} S_1 \vee \begin{pmatrix} 1 \le p_1 \le \infty \wedge \\ 1 \le p_2 \le \infty \wedge \\ 1 \le p_3 \le \infty \end{pmatrix}$$

Applying the Main Theorem



By the Main Theorem, we know we can apply the following algorithm to counting constraint S to obtain $post^*(S)$.

$$S = S_0: \begin{pmatrix} 3 \leq p_1 \leq \infty \wedge \\ 0 \leq p_2 \leq 0 \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix}$$

$\xrightarrow{t_1^*}$

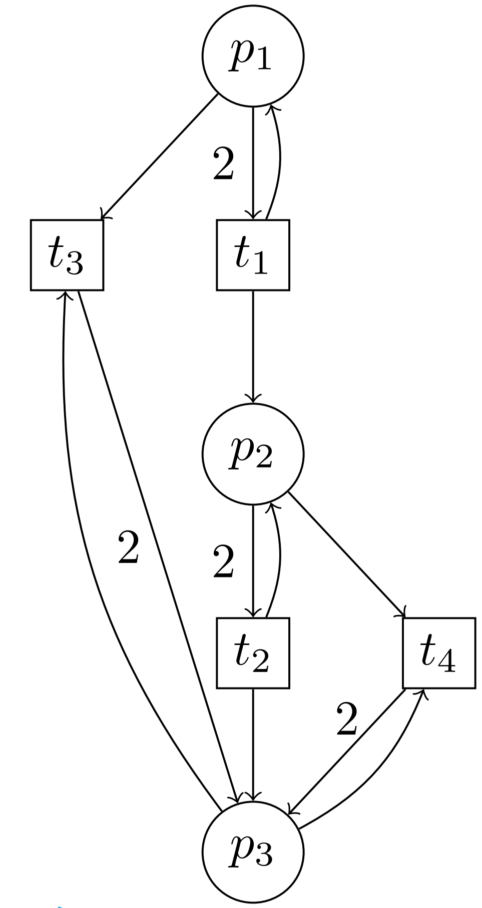
$$S_1: S_0 \vee \begin{pmatrix} 2 \leq p_1 \leq \infty \wedge \\ 1 \leq p_2 \leq \infty \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix} \vee \begin{pmatrix} 1 \leq p_1 \leq \infty \wedge \\ 2 \leq p_2 \leq \infty \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix}$$

$\xrightarrow{t_1^*}$ S_1

$\xrightarrow{t_2^*}$

$$S_2: S_1 \vee \begin{pmatrix} 1 \leq p_1 \leq \infty \wedge \\ 1 \leq p_2 \leq \infty \wedge \\ 1 \leq p_3 \leq \infty \end{pmatrix}$$

Applying the Main Theorem



By the Main Theorem, we know we can apply the following algorithm to counting constraint S to obtain $post^*(S)$.

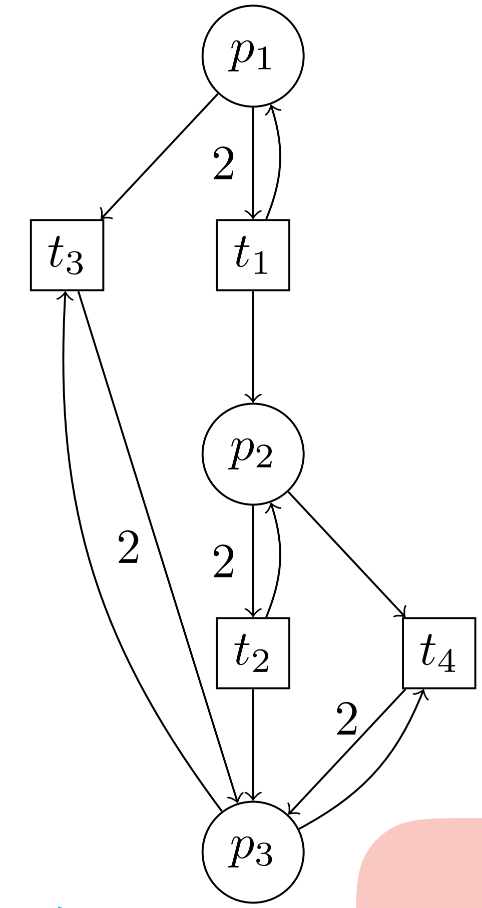
$$S = S_0 = \begin{pmatrix} 3 \leq p_1 \leq \infty \wedge \\ 0 \leq p_2 \leq 0 \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix} \xrightarrow{t_1^*} S_1 = S_0 \vee \begin{pmatrix} 2 \leq p_1 \leq \infty \wedge \\ 1 \leq p_2 \leq \infty \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix} \vee \begin{pmatrix} 1 \leq p_1 \leq \infty \wedge \\ 2 \leq p_2 \leq \infty \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix}$$

$$S_1 \xrightarrow{t_1^*} S_1 \quad (\xrightarrow{t_1^*} \cdots, \ \xrightarrow{t_2^*} \cdots)$$

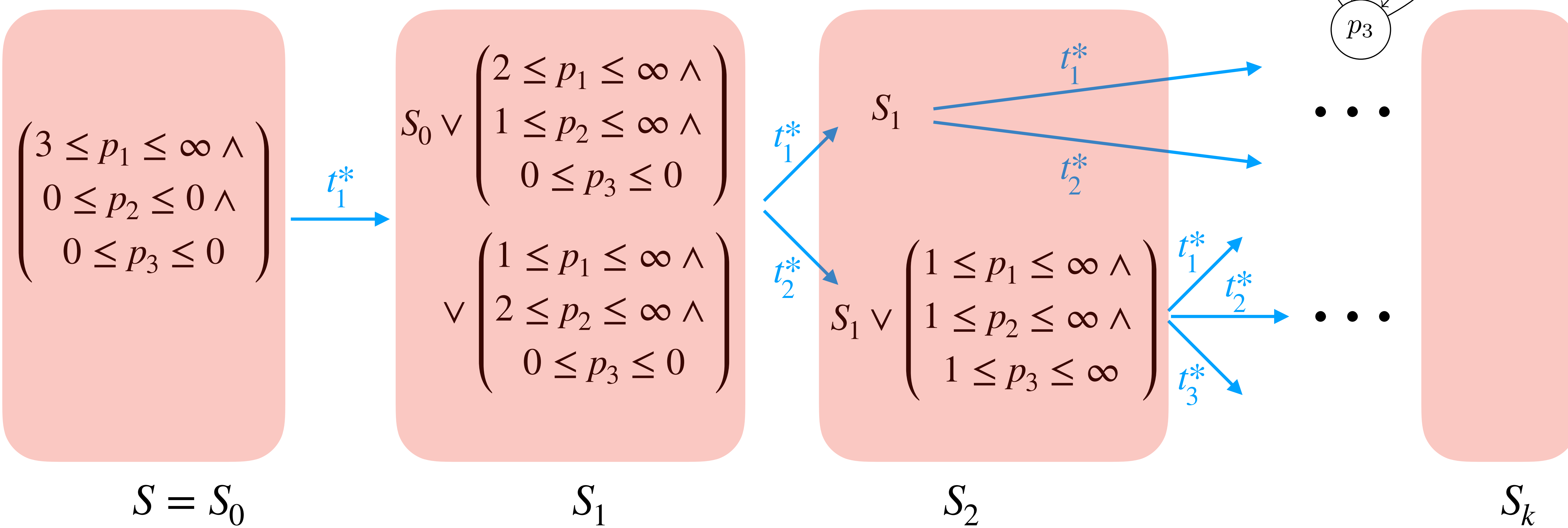
$$S_1 \xrightarrow{t_2^*} S_2 = S_1 \vee \begin{pmatrix} 1 \leq p_1 \leq \infty \wedge \\ 1 \leq p_2 \leq \infty \wedge \\ 1 \leq p_3 \leq \infty \end{pmatrix} \quad (\xrightarrow{t_1^*}, \xrightarrow{t_2^*}, \xrightarrow{t_3^*} \cdots)$$

$S = S_0$ $\qquad S_1$ $\qquad S_2$

Applying the Main Theorem



By the Main Theorem, we know we can apply the following algorithm to counting constraint S to obtain $post^*(S)$.

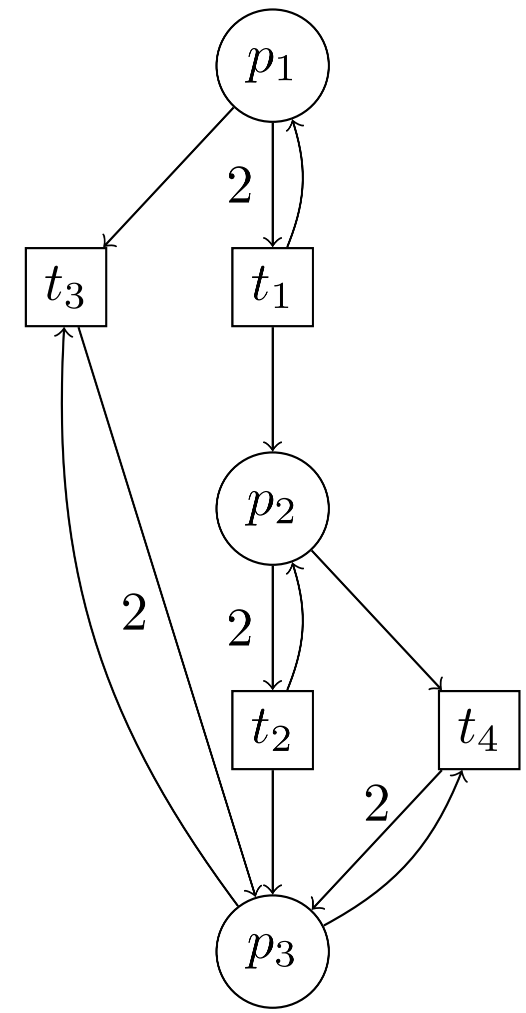


$$post^*(S) = S_0 \vee S_1 \vee S_2 \vee \ldots \vee S_k$$

Applying the Main Theorem

This protocol is correct if and only if for every initial marking $M_0^{(n)}$:

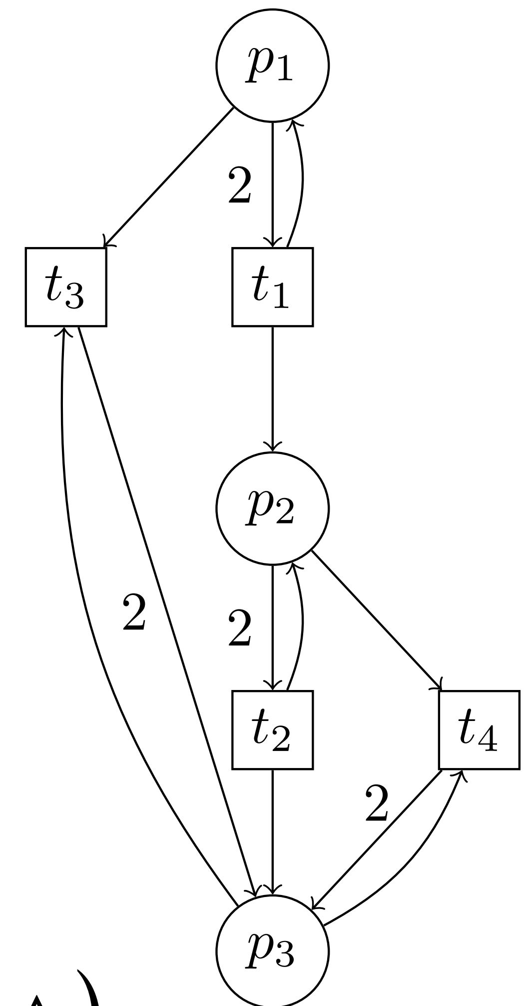
- $n \geq 3 \Rightarrow$ all markings reachable from $M_0^{(n)}$ can reach the marking with all tokens in p_3.
- $n < 3 \Rightarrow$ there is no reachable marking with a token in p_3.



Applying the Main Theorem

This protocol is correct if and only if for every initial marking $M_0^{(n)}$:

- $n \geq 3 \Rightarrow$ all markings reachable from $M_0^{(n)}$ can reach the marking with all tokens in p_3.
- $n < 3 \Rightarrow$ there is no reachable marking with a token in p_3.

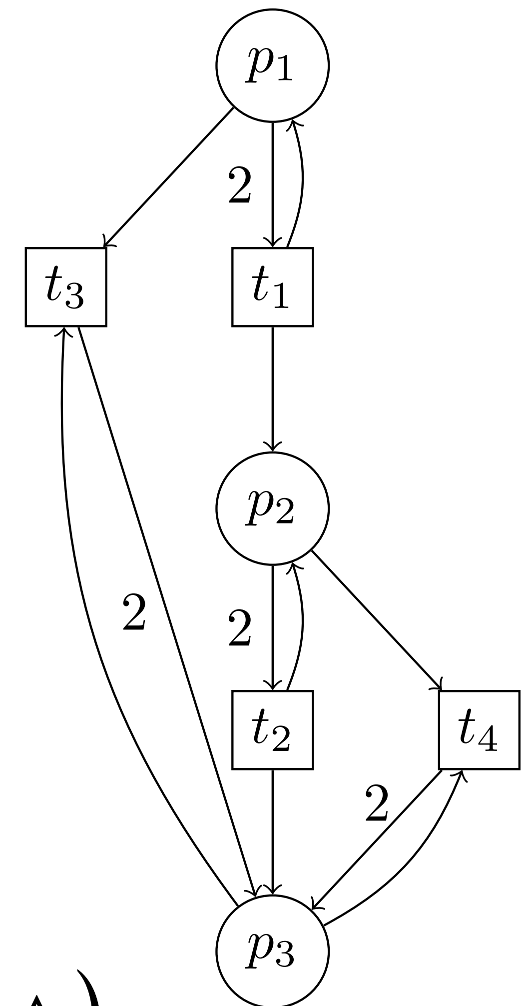


$$post^*\begin{pmatrix} 3 \leq p_1 \leq \infty \wedge \\ 0 \leq p_2 \leq 0 \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix} \subseteq pre^*\begin{pmatrix} 0 \leq p_1 \leq 0 \wedge \\ 0 \leq p_2 \leq 0 \wedge \\ 3 \leq p_3 \leq \infty \end{pmatrix}$$

Applying the Main Theorem

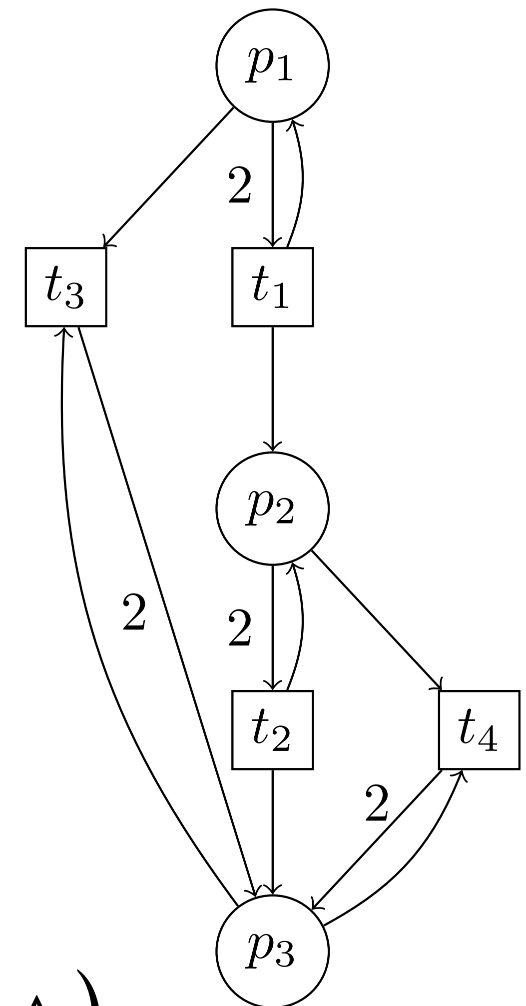
This protocol is correct if and only if for every initial marking $M_0^{(n)}$:

- $n \geq 3 \Rightarrow$ all markings reachable from $M_0^{(n)}$ can reach the marking with all tokens in p_3.
- $n < 3 \Rightarrow$ there is no reachable marking with a token in p_3.



$$post^*\begin{pmatrix} 3 \leq p_1 \leq \infty \wedge \\ 0 \leq p_2 \leq 0 \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix} \subseteq pre^*\begin{pmatrix} 0 \leq p_1 \leq 0 \wedge \\ 0 \leq p_2 \leq 0 \wedge \\ 3 \leq p_3 \leq \infty \end{pmatrix}$$

Applying the Main Theorem



This protocol is correct if and only if for every initial marking $M_0^{(n)}$:

- $n \geq 3 \Rightarrow$ all markings reachable from $M_0^{(n)}$ can reach the marking with all tokens in p_3.
- $n < 3 \Rightarrow$ there is no reachable marking with a token in p_3.

$$post^* \begin{pmatrix} 3 \leq p_1 \leq \infty \wedge \\ 0 \leq p_2 \leq 0 \wedge \\ 0 \leq p_3 \leq 0 \end{pmatrix} \subseteq pre^* \begin{pmatrix} 0 \leq p_1 \leq 0 \wedge \\ 0 \leq p_2 \leq 0 \wedge \\ 3 \leq p_3 \leq \infty \end{pmatrix}$$

check inclusion

Conclusion

- We study class of Petri nets with applications to population protocols and chemical reaction networks.

Conclusion

- We study class of Petri nets with applications to population protocols and chemical reaction networks.

- The applications require to solve parameterized problems → very hard

Conclusion

- We study class of Petri nets with applications to population protocols and chemical reaction networks.

- The applications require to solve parameterized problems → very hard

- Main result: for immediate observation nets, parameterized questions are as simple as “singleton” questions!

Conclusion

- We study class of Petri nets with applications to population protocols and chemical reaction networks.

- The applications require to solve parameterized problems → very hard

- Main result: for immediate observation nets, parameterized questions are as simple as “singleton” questions!

Thank you !

Building Block

Pruning Theorem

Let *N* be an IO net with *n* places.

If a marking *M’* is coverable by some marking *M*, then *M’* is coverable by some marking *C* such that

1. *C* is covered by *M*.
2. $|C| \leq |M| + n^3$

Building Block

Pruning Theorem

Let *N* be an IO net with *n* places.

If a marking *M'* is coverable by some marking *M*, then *M'* is coverable by some marking *C* such that

1. *C* is covered by *M*.
2. $|C| \leq |M| + n^3$

Pruning: we remove tokens from the run that covers M' without modifying its covering property.