



Network Medicine

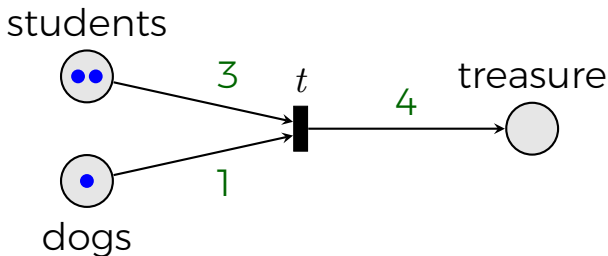
Petri Nets: Properties

Sergiu Ivanov

`sergiu.ivanov@univ-evry.fr`

`https://www.ibisc.univ-evry.fr/~sivanov/en/
pn-biomodelling.html`

Petri Nets: Reminder



$$P = \{\text{students, dogs, treasure}\} \quad T = \{t\}$$

	(students, t)	(dogs, t)	(t, treasure)
W	3	1	4
	students	dogs	treasure
M_0	2	1	0

The Spirit of the Lecture

What? Formal descriptions of what we (would like to) see.

Why? To properly formulate queries to computers.

Feels like? A kind of mind games.

Outline

1. Behavioural Properties

2. Structural Properties

Outline

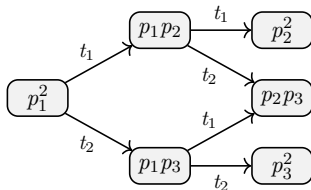
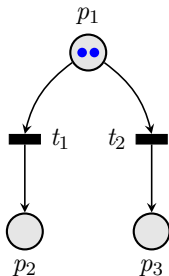
1. Behavioural Properties

2. Structural Properties

Properties of the **state graph**.

- ▶ how does the net evolve?
- ▶ which markings may it attain?

May depend on the **evolution mode**.

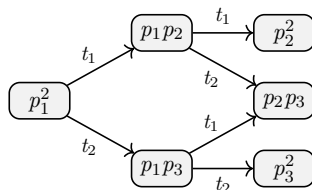
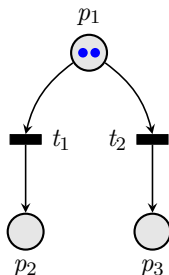


the state graph under asynchronous mode

Reachability

Given a marking (state) M , can the net reach it from its initial marking M_0 ?

Depends on the mode.



asynchronous mode

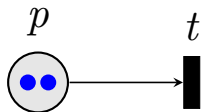
Markings p_1^2 , p_1p_2 , p_1p_3 , p_2^2 , p_2p_3 , p_3^2 are reachable.

Are there any markings this net cannot reach under asyn?

p_1^3 , $p_1^2p_2$, $p_2^2p_3$, etc.

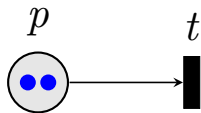
What is the reachability set? $1/3$

Reachability set = all states listed in the state graph



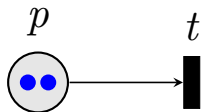
What is the reachability set? $1/3$

Reachability set = all states listed in the state graph



What is the reachability set? $1/3$

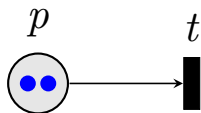
Reachability set = all states listed in the state graph



Answer: $\{p^2, p^1, p, \lambda\}$ \leftarrow λ is the empty marking

What is the reachability set? $1/3$

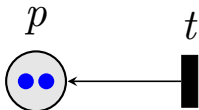
Reachability set = all states listed in the state graph



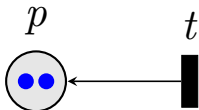
Answer: $\{p^2, p^1, p, \lambda\}$ \leftarrow λ is the empty marking

Does the choice of the evolution mode matter?

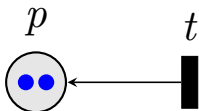
What is the reachability set? $2/3$



What is the reachability set? $2/3$



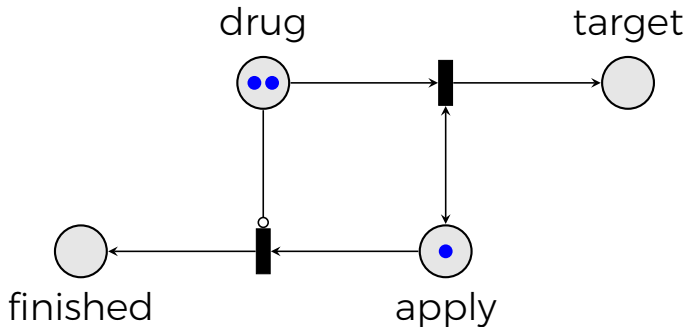
What is the reachability set? $2/3$



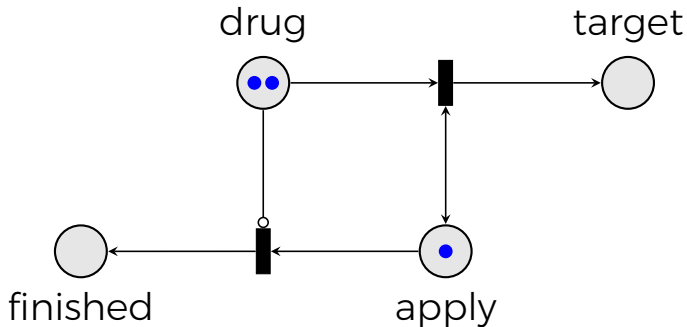
Answer: $\{p^k \mid k \in \mathbb{N}, k \geq 2\}$

Does the choice of the evolution mode matter?

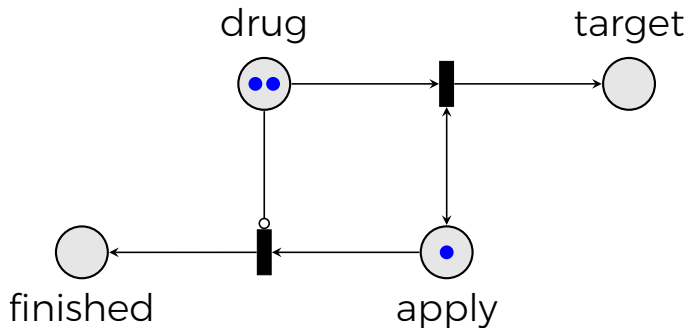
What is the reachability set? 3/3



What is the reachability set? 3/3

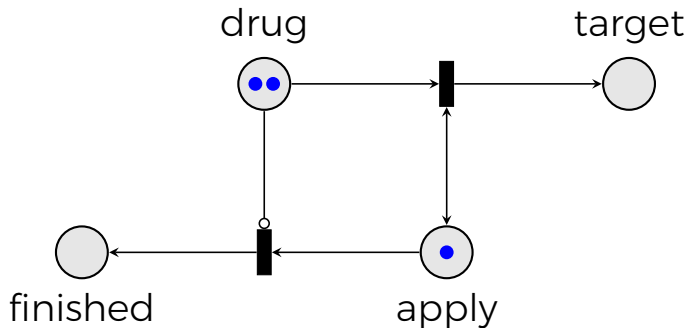


What is the reachability set? ^{3/3}



Answer: $\{d^2a, dat, at^2, ft^2\}$

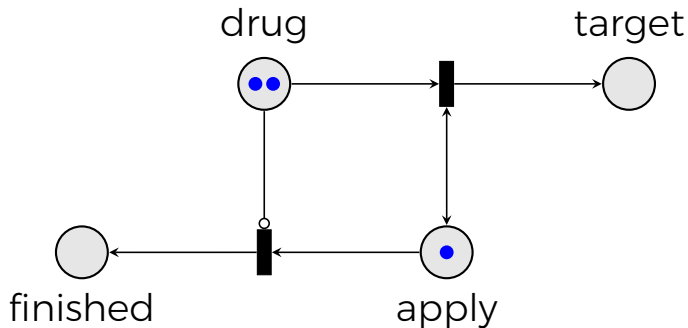
What is the reachability set? ^{3/3}



Answer: $\{d^2a, dat, at^2, ft^2\}$

What is the activity/phenomenon this net models?

What is the reachability set? 3/3



Answer: $\{d^2a, dat, at^2, ft^2\}$

What is the activity/phenomenon this net models?

Does the choice of the evolution mode matter?

Sidenote: Asyn is Often Used in Modelling

The **asynchronous mode** well represents arbitrary interleaving of process interactions.

Ensuring a certain behaviour under the **asynchronous mode** means proper synchronisation.

Reachability is Hard

Reachability is **decidable**.

- ▶ \exists a Turing machine deciding whether a marking is reachable or not.

Reachability is **EXPSPACE-hard**.

- ▶ A Turing machine needs at least **exponential space** on the band in order to decide whether a marking is reachable or not.
- ▶ Essentially, one needs to look over **almost all** of the reachability graph.

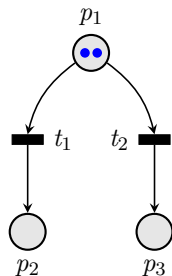
Coverability: “Lighter” Reachability

Given a marking M , can the net reach a marking M' such that M' covers M ?

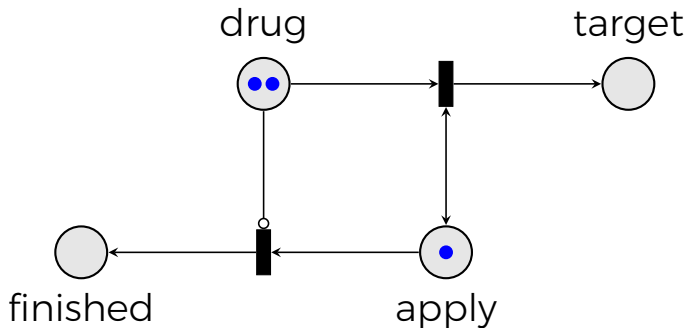
- ▶ M' covers M if all places in M' contain at least as many tokens as in M ($M' \geq M$).

Markings p_2 and p_3

- ▶ are coverable under both **syn** and **asyn**
- ▶ **not** reachable under **syn**



Is a Marking Coverable?

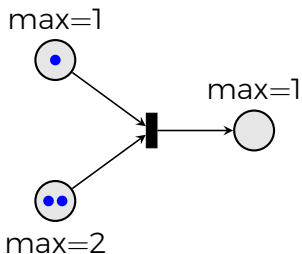


Which of these markings are coverable: dt , af , df ?

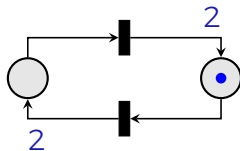
Boundedness

A Petri net is **bounded** if the number of tokens in every place **never exceeds** a fixed constant.

Bounded



Unbounded



The number of tokens in the net increases at every step.

Unboundedness \implies Cycles?

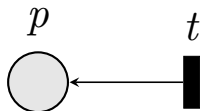
Do all **unbounded** Petri nets have **cycles**?

Unboundedness \implies Cycles?

Do all **unbounded** Petri nets have **cycles**?

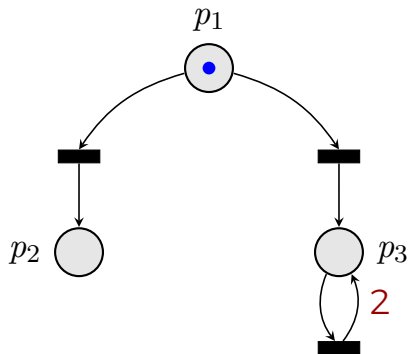
Unboundedness \implies Cycles?

Do all **unbounded** Petri nets have **cycles**?

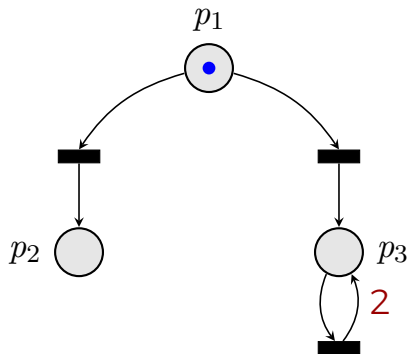


t produces new tokens all the time.

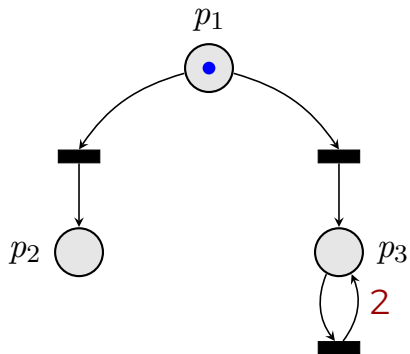
Is This Net Unbounded? $1/2$



Is This Net Unbounded? $1/2$



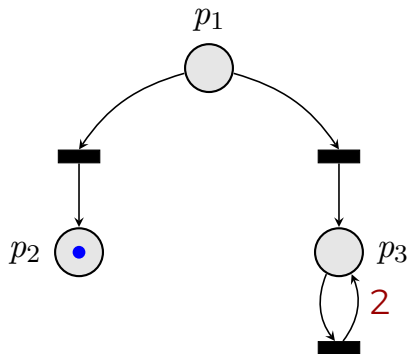
Is This Net Unbounded? $1/2$



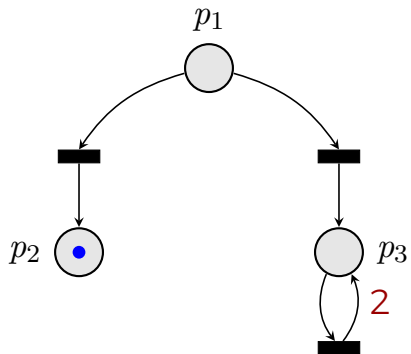
Answer: **Yes**

The token may get into p_3 .

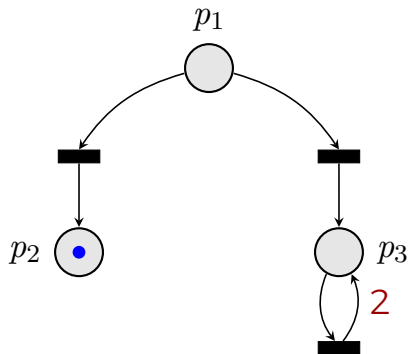
Is This Net Unbounded? $2/2$



Is This Net Unbounded? $2/2$



Is This Net Unbounded? ^{2/2}



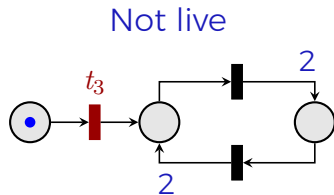
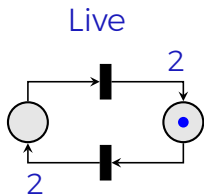
Answer: **No**

No tokens ever get into p_3 .

Liveness

A Petri net is **live** if, starting from any reachable marking, **any transition** in the net can be **eventually fired**.

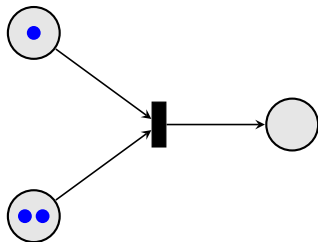
$\forall M \in \text{Reachable}(M_0), \forall t \in \text{Transitions},$
 $\exists M' \in \text{Reachable}(M)$ such that t is enabled at M' .



t_3 may only fire once.

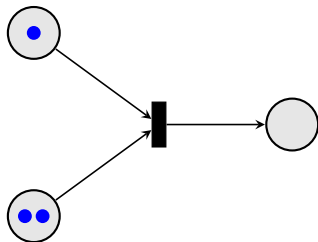
Deadlocks

Is this net *live*?



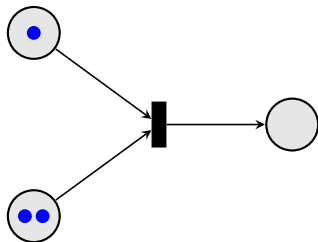
Deadlocks

Is this net *live*?



Deadlocks

Is this net *live*?



Answer: **No**

Deadlock = a state in which **no transitions** are enabled.

What are the **deadlocks** of this net?

Summary of Behavioural Properties

- ▶ **Reachability** and coverability
 - ▶ Can a given marking be reached/covered?
- ▶ **Boundedness**
 - ▶ Is there a fixed upper bound on the number of tokens in all places?
- ▶ **Liveness** and deadlocks
 - ▶ Can any transition fire arbitrarily often?

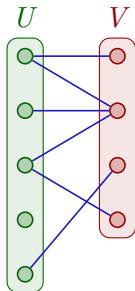
Outline

1. Behavioural Properties

2. Structural Properties

Bipartite Graphs

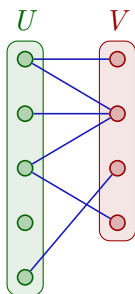
A graph is **bipartite** if its vertices can **partitioned** into two sets U and V such that every **edge** connects a **vertex from U** to a **vertex from V** (or a **vertex from V** to a **vertex from U**).



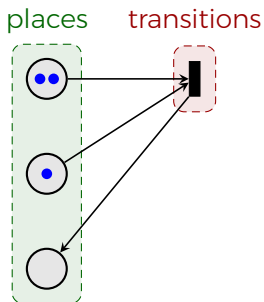
no edges within U or V

Bipartite Graphs

A graph is **bipartite** if its vertices can be **partitioned** into two sets U and V such that every **edge** connects a **vertex from U** to a **vertex from V** (or a **vertex from V** to a **vertex from U**).



no edges within U or V



Petri nets are bipartite graphs.

https://en.wikipedia.org/wiki/Bipartite_graph

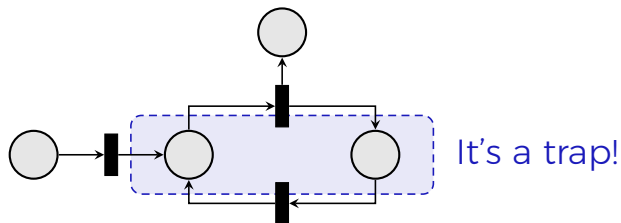
Properties depending only on the **graph structure**.

- ▶ independent of the dynamic states
- ▶ induced by loops, cycles, SCC, etc.

Properties that hold **independently of the initial marking**.

Traps

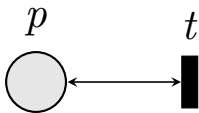
Trap = a subset of places S such that all transitions consuming tokens from S also put tokens into S .



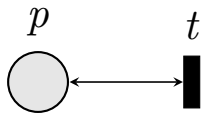
Once a trap contains tokens, it will **always** contain tokens.

Traps do not include transitions.

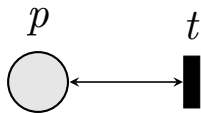
Where Is the Trap? $1/2$



Where Is the Trap? $1/2$



Where Is the Trap? $1/2$



Answer: $\{p\}$

Where Is the Trap? $2/2$



<https://en.wikipedia.org/wiki/Vacuously>

Where Is the Trap? $2/2$



<https://en.wikipedia.org/wiki/Vacuously>

Where Is the Trap? ^{2/2}

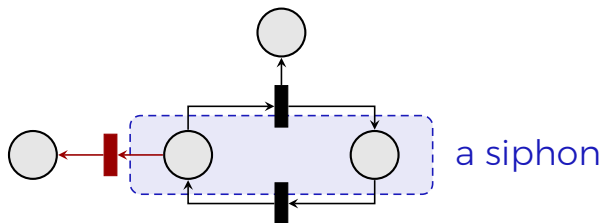


Answer: $\{p_1\}, \{p_2\}, \{p_1, p_2\}$

The property of being a trap is satisfied vacuously.

Siphons

Siphon = a subset of **places** S such that all transitions putting tokens **into** S also consume tokens **from** S .



Siphons are **duals** (the opposite) of **traps**.

Once a siphon contains no tokens, it will **never** contain tokens again.

Where Is the Siphon?



Where Is the Siphon?



Where Is the Siphon?



Answer: $\{p_1\}$, $\{p_2\}$, $\{p_1, p_2\}$

Where Is the Siphon?



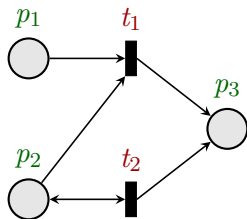
Answer: $\{p_1\}$, $\{p_2\}$, $\{p_1, p_2\}$

Can you think of **other** examples of siphons?

The incidence matrix M of a Petri net contains

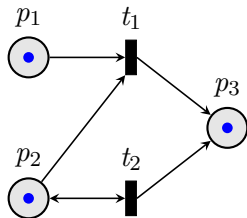
- ▶ one row per place
- ▶ one column per transition

Cell (p, t) contains the value by which the number of tokens in p changes when t fires.



$$M = \begin{array}{c|cc} & t_1 & t_2 \\ \hline p_1 & -1 & 0 \\ p_2 & -1 & 0 \\ p_3 & 1 & 1 \end{array}$$

Petri Nets as Linear Operators: Dynamics

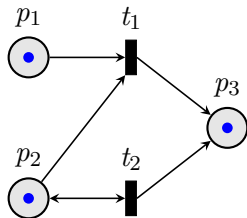


$$M = \begin{array}{c|cc} & t_1 & t_2 \\ \hline p_1 & -1 & 0 \\ p_2 & -1 & 0 \\ p_3 & 1 & 1 \end{array}$$

Current marking: $p_1^1 p_2^1 p_3^1 \mapsto \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

The firing vector: $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ — t_1 fires once, t_2 does not fire

Petri Nets as Linear Operators: Dynamics



$$M = \begin{array}{c|cc} & t_1 & t_2 \\ \hline p_1 & -1 & 0 \\ p_2 & -1 & 0 \\ p_3 & 1 & 1 \end{array}$$

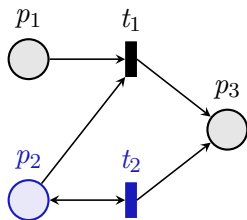
Current marking: $p_1^1 p_2^1 p_3^1 \mapsto \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

The firing vector: $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ — t_1 fires once, t_2 does not fire

Next marking:

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} -1 & 0 \\ -1 & 0 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$$

Petri Nets \neq Linear Operators



$$M = \begin{array}{c|cc} & t_1 & t_2 \\ \hline p_1 & -1 & 0 \\ p_2 & -1 & 0 \\ p_3 & 1 & 1 \end{array}$$

Note: t_2 cannot fire if p_2 is empty.

Cells (p_1, t_2) , (p_2, t_2) both contain 0, but t_2 actually depends on the number of tokens in p_2 !

- ▶ Petri nets cannot be completely reduced to linear operators.

Two Matrix-based Structural Properties

Transition invariant = a firing vector F such that $M \cdot F = 0$.

- ▶ F describes how to fire transitions such that the contents of the places does not change.
- ▶ for **any marking!**

Place invariant = a vector Y such that $M^T \cdot Y = 0$

- ▶ Existence of place invariants with all components non-negative \implies **conservation of tokens** (like in chemistry).

Structural Properties and Behaviour

Structural properties = strong properties

- ▶ derived from the structure of the network
 - ▶ holding for any possible state
-

For some types of Petri nets, **behavioural properties** can be described purely **structurally**:

- ▶ **place invariants** may describe **boundedness**
 - ▶ **traps and siphons** may describe **liveness**
-

Structural properties are easier to handle.

- ▶ no need to look at the state graph

Summary of Structural Properties

- ▶ Traps

- ▶ Once non-empty, always non-empty.

- ▶ Siphons

- ▶ Once empty, always empty.

- ▶ Matrix-based

- ▶ Properties of the incidence matrix.