

Petri nets and weighted automata

Filip Mazowiecki

UNIwersYTET WARSZAWSKI

Petri nets and weighted automata

Filip Mazowiecki

UNIwersYTET WARSZAWSKI

What I could say that I work on

Formal verification



What I could say that I work on

Formal verification

What I want say that I work on

Math puzzles



What I could say that I work on

Formal verification

What I want say that I work on

Math puzzles

What I will say?

A bit of both



What I could say that I work on

Formal verification

What I want say that I work on

Math puzzles

What I will say?

A bit of both

What are Petri nets and weighted automata?

- Mathematical models to verify programs, business models, etc



What I could say that I work on

Formal verification

What I want say that I work on

Math puzzles

What I will say?

A bit of both

What are Petri nets and weighted automata?

- Mathematical models to verify programs, business models, etc
- Automata with registers: Petri nets over \mathbb{N} (with > 0 tests),
weighted automata over \mathbb{Q} (blind)



Petri nets geometric intuition

Petri net (d, T) : d – dimension, $T \subseteq \mathbb{Z}^d$ – transitions



Petri nets geometric intuition

Petri net (d, T) : d – dimension, $T \subseteq \mathbb{Z}^d$ – transitions

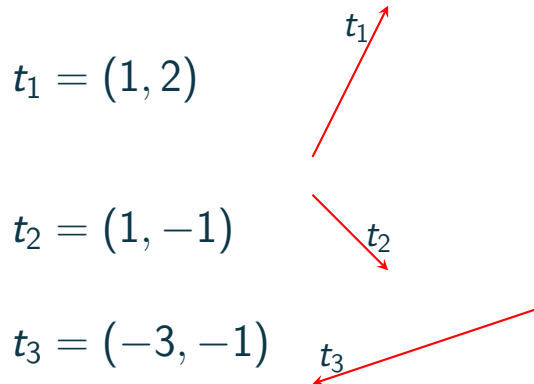
Example: $d = 2$, $T = \{t_1, t_2, t_3\}$



Petri nets geometric intuition

Petri net (d, T) : d – dimension, $T \subseteq \mathbb{Z}^d$ – transitions

Example: $d = 2$, $T = \{t_1, t_2, t_3\}$

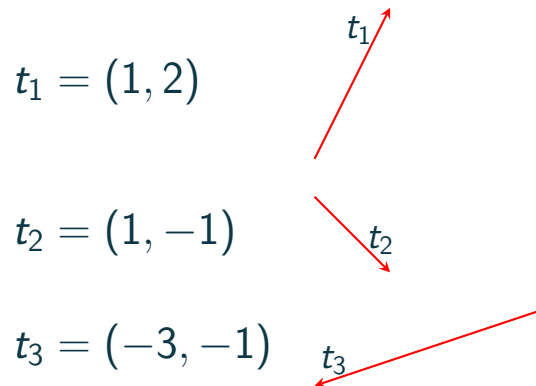


Petri nets geometric intuition



Petri net (d, T) : d – dimension, $T \subseteq \mathbb{Z}^d$ – transitions

Example: $d = 2$, $T = \{t_1, t_2, t_3\}$



Reachability problem: given (d, T) and $\mathbf{a}, \mathbf{b} \in \mathbb{N}^d$

Can I always go from \mathbf{a} to \mathbf{b} staying within \mathbb{N}^d ?

Petri nets geometric intuition

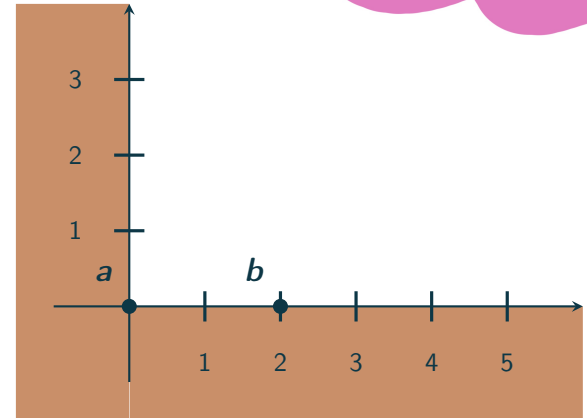
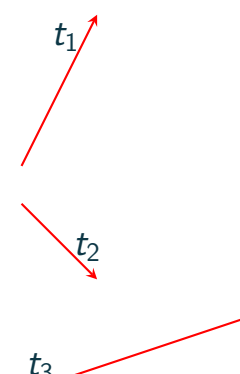
Petri net (d, T) : d – dimension, $T \subseteq \mathbb{Z}^d$ – transitions

Example: $d = 2$, $T = \{t_1, t_2, t_3\}$

$t_1 = (1, 2)$

$t_2 = (1, -1)$

$t_3 = (-3, -1)$



Reachability problem: given (d, T) and $a, b \in \mathbb{N}^d$

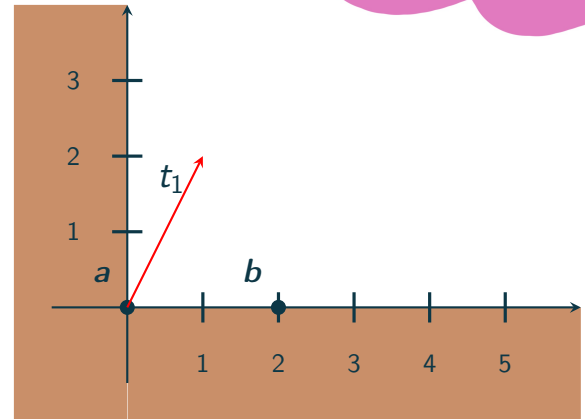
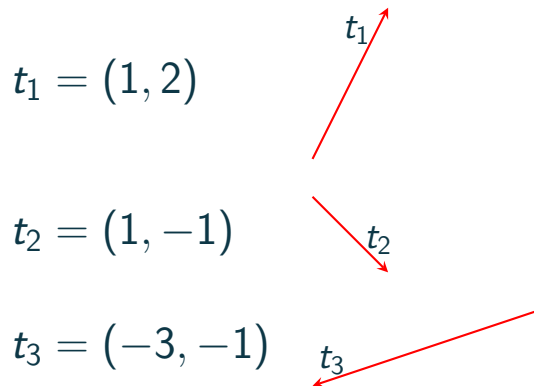
Can I always go from a to b staying within \mathbb{N}^d ?

Petri nets geometric intuition



Petri net (d, T) : d – dimension, $T \subseteq \mathbb{Z}^d$ – transitions

Example: $d = 2$, $T = \{t_1, t_2, t_3\}$



Reachability problem: given (d, T) and $a, b \in \mathbb{N}^d$

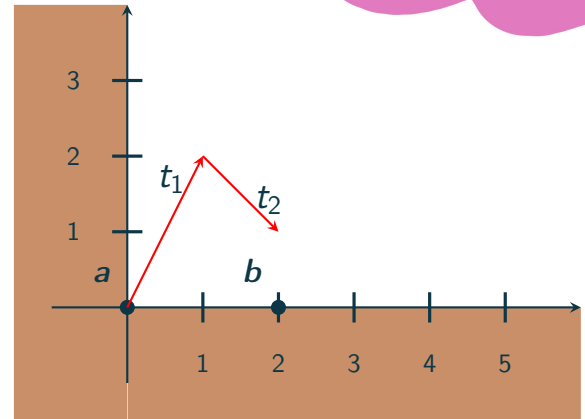
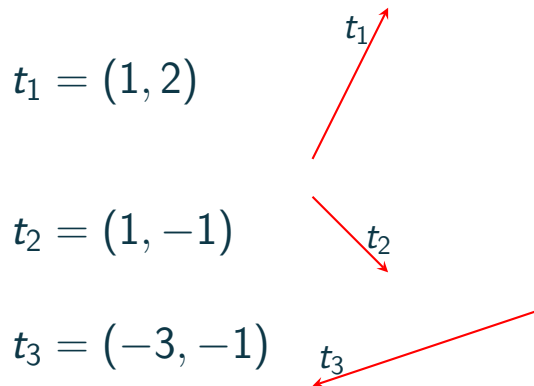
Can I always go from a to b staying within \mathbb{N}^d ?

Petri nets geometric intuition



Petri net (d, T) : d – dimension, $T \subseteq \mathbb{Z}^d$ – transitions

Example: $d = 2$, $T = \{t_1, t_2, t_3\}$



Reachability problem: given (d, T) and $a, b \in \mathbb{N}^d$

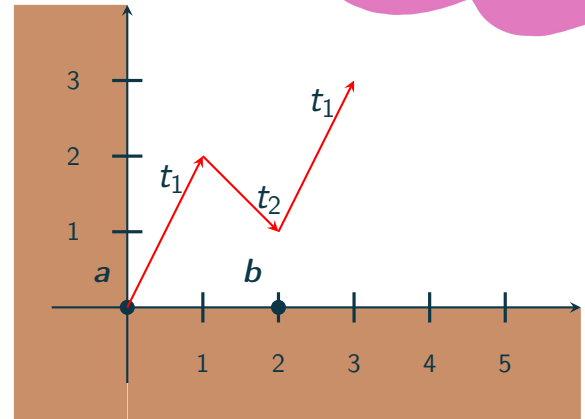
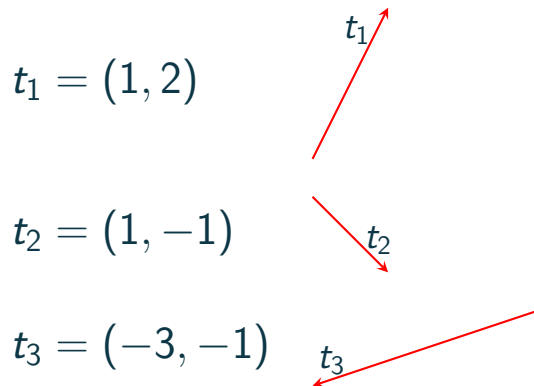
Can I always go from a to b staying within \mathbb{N}^d ?

Petri nets geometric intuition



Petri net (d, T) : d – dimension, $T \subseteq \mathbb{Z}^d$ – transitions

Example: $d = 2$, $T = \{t_1, t_2, t_3\}$



Reachability problem: given (d, T) and $a, b \in \mathbb{N}^d$

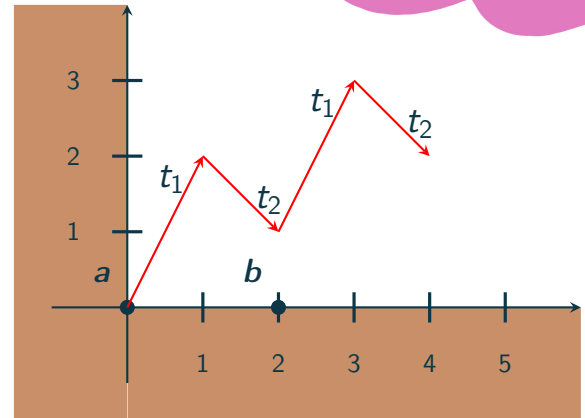
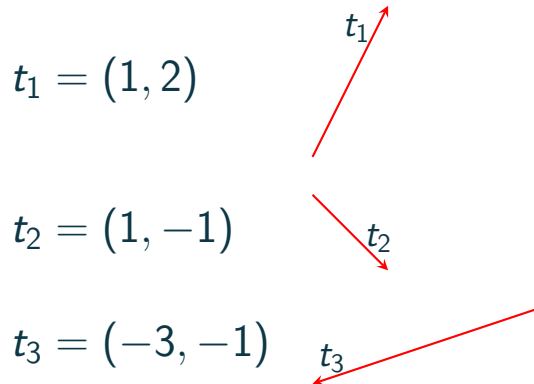
Can I always go from a to b staying within \mathbb{N}^d ?

Petri nets geometric intuition



Petri net (d, T) : d – dimension, $T \subseteq \mathbb{Z}^d$ – transitions

Example: $d = 2$, $T = \{t_1, t_2, t_3\}$



Reachability problem: given (d, T) and $a, b \in \mathbb{N}^d$

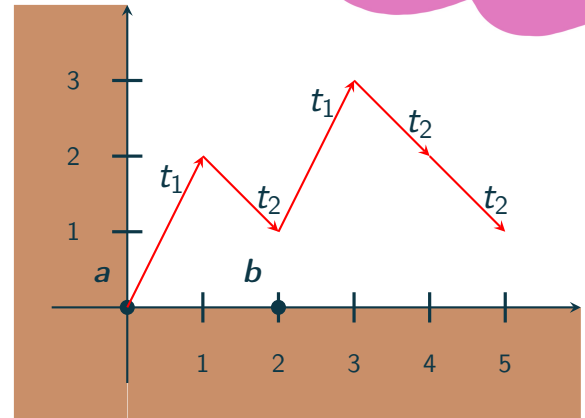
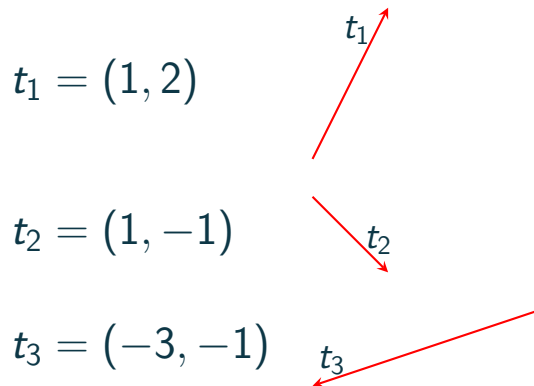
Can I always go from a to b staying within \mathbb{N}^d ?

Petri nets geometric intuition



Petri net (d, T) : d – dimension, $T \subseteq \mathbb{Z}^d$ – transitions

Example: $d = 2$, $T = \{t_1, t_2, t_3\}$



Reachability problem: given (d, T) and $a, b \in \mathbb{N}^d$

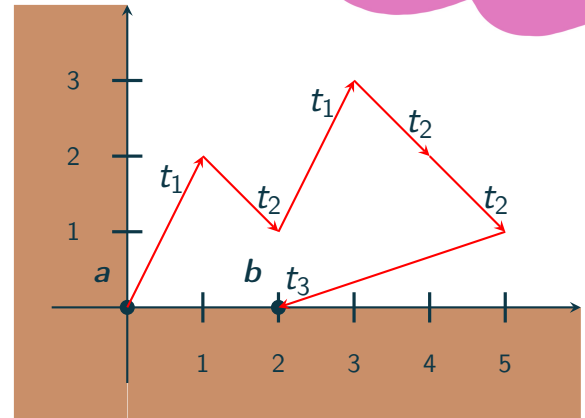
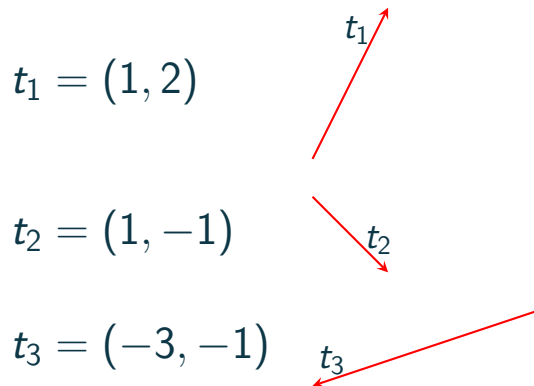
Can I always go from a to b staying within \mathbb{N}^d ?

Petri nets geometric intuition



Petri net (d, T) : d – dimension, $T \subseteq \mathbb{Z}^d$ – transitions

Example: $d = 2$, $T = \{t_1, t_2, t_3\}$



Reachability problem: given (d, T) and $a, b \in \mathbb{N}^d$

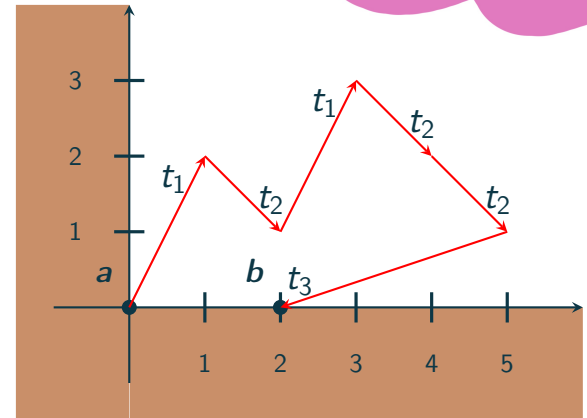
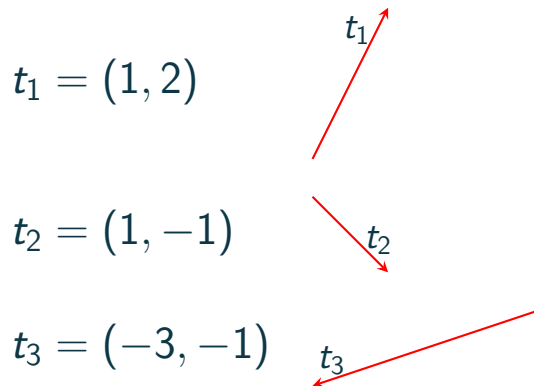
Can I always go from a to b staying within \mathbb{N}^d ?

Petri nets geometric intuition



Petri net (d, T) : d – dimension, $T \subseteq \mathbb{Z}^d$ – transitions

Example: $d = 2$, $T = \{t_1, t_2, t_3\}$



Reachability problem: given (d, T) and $a, b \in \mathbb{N}^d$

Can I always go from a to b staying within \mathbb{N}^d ?

Soundness problems: given (d, T) and starting in $(k, 0, 0, \dots, 0)$

whatever I do can I always reach $(0, 0, \dots, 0, k)$ staying within \mathbb{N}^d ?

How can that be useful?



```
{x = True }  
1: goto 2  
2: if x then goto 3 else goto 1  
3: x = !x  
4: # critical section  
5: x = !x, exit
```

How can that be useful?



```
{x = True }  
1: goto 2  
2: if x then goto 3 else goto 1  
3: x = !x  
4: # critical section  
5: x = !x, exit
```

In the first line arbitrary many processes can appear

Problem: can two processes appear at once in the **critical section**?

How can that be useful?



{x = True }

1: **goto** 2

2: **if** x **then goto** 3 **else goto** 1

3: x = !x

4: **# critical section**

5: x = !x, **exit**



{bathroom free}

go to the door

if free enter, otherwise go back

change sign

bathroom

change sign and leave

In the first line arbitrary many processes can appear

Problem: can two processes appear at once in the **critical section**?

How can that be useful?



{x = True }

1: **goto** 2

2: **if** x **then goto** 3 **else goto** 1

3: x = !x

4: # critical section

5: x = !x, **exit**



{bathroom free}

go to the door

if free enter, otherwise go back

change sign

bathroom

change sign and leave

In the first line arbitrary many processes can appear

Problem: can two processes appear at once in the **critical section**?

We will model this with a Petri net (d, T)

- dimension d : every line and possible values of x,

How can that be useful?



{x = True }

1: **goto** 2

2: **if** x **then goto** 3 **else goto** 1

3: x = !x

4: **# critical section**

5: x = !x, **exit**



{bathroom free}

go to the door

if free enter, otherwise go back

change sign

bathroom

change sign and leave

In the first line arbitrary many processes can appear

Problem: can two processes appear at once in the **critical section**?

We will model this with a Petri net (d, T)

- dimension d : every line and possible values of x ,
- transitions: how processes/people move and change the values of x

Modelling (reachability problem)


corridor

check


bathroom

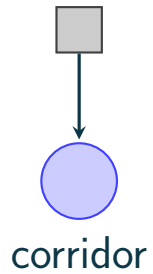


occupied

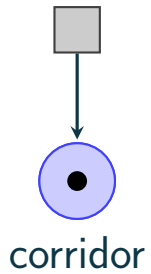

free



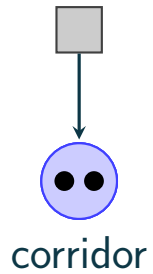
Modelling (reachability problem)



Modelling (reachability problem)



Modelling (reachability problem)



check



bathroom



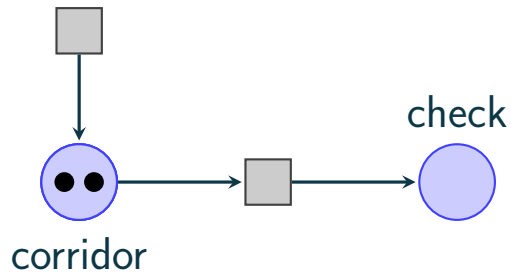
occupied



free



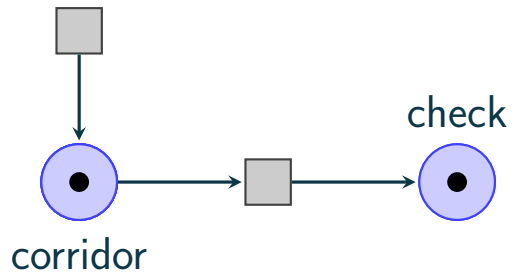
Modelling (reachability problem)



bathroom



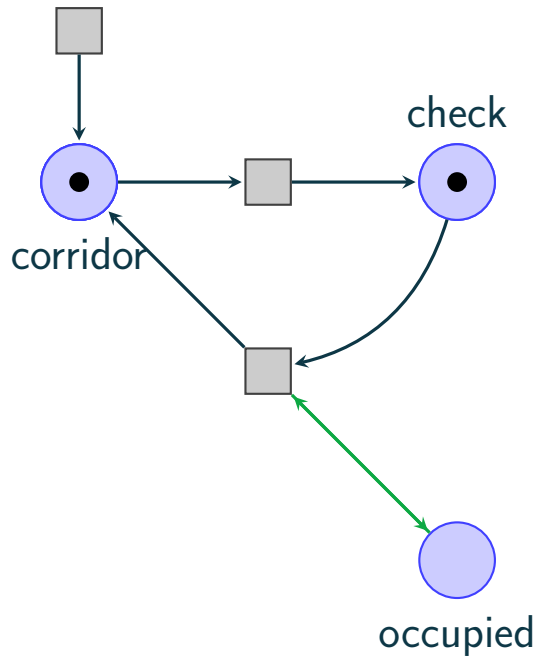
Modelling (reachability problem)



bathroom



Modelling (reachability problem)



bathroom



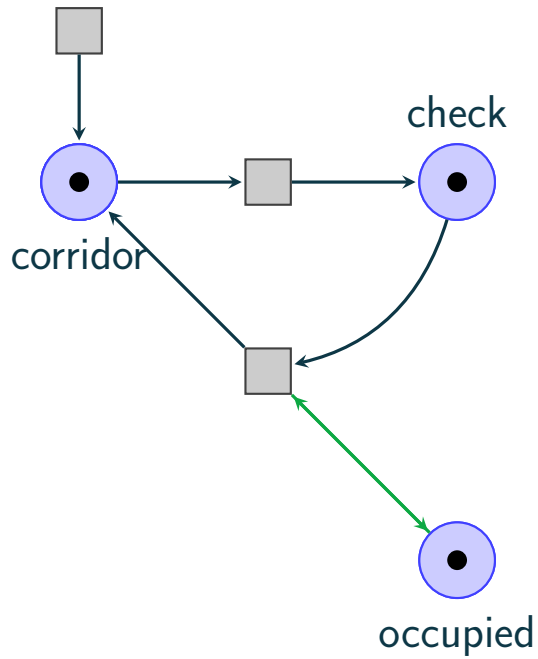
occupied



free



Modelling (reachability problem)



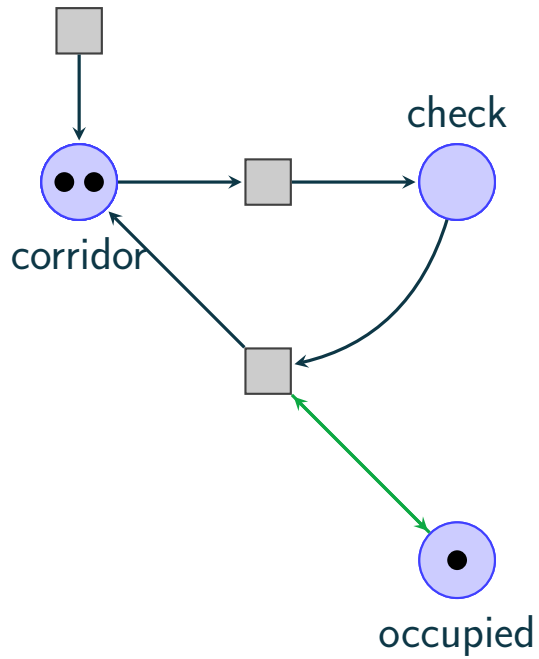
bathroom



free



Modelling (reachability problem)



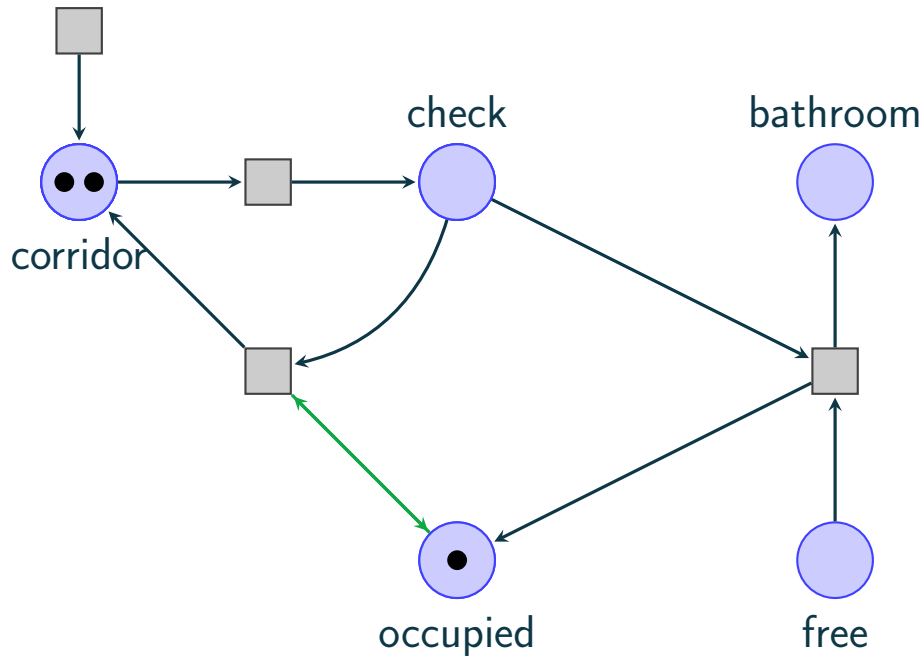
bathroom



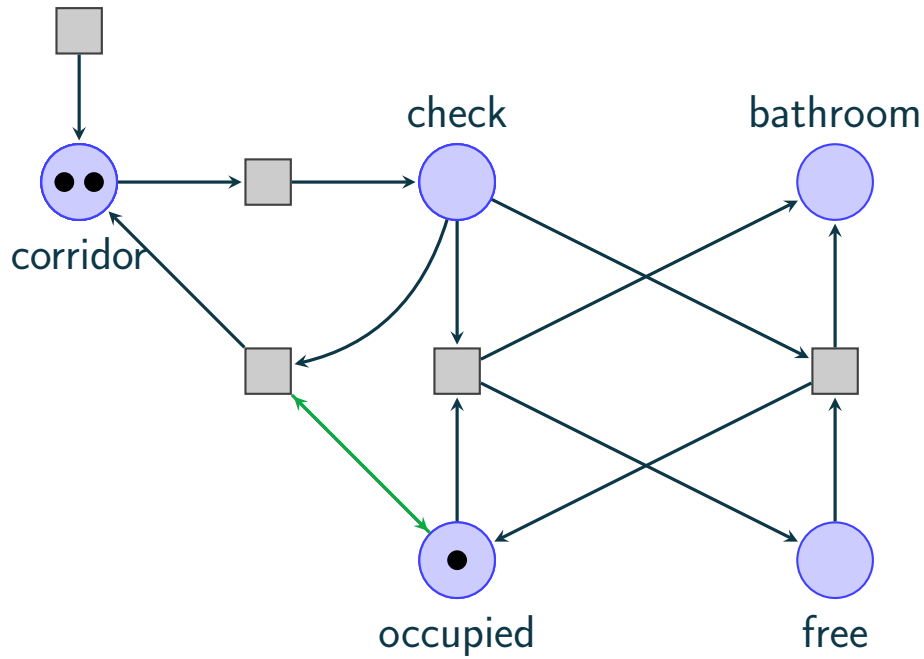
free



Modelling (reachability problem)



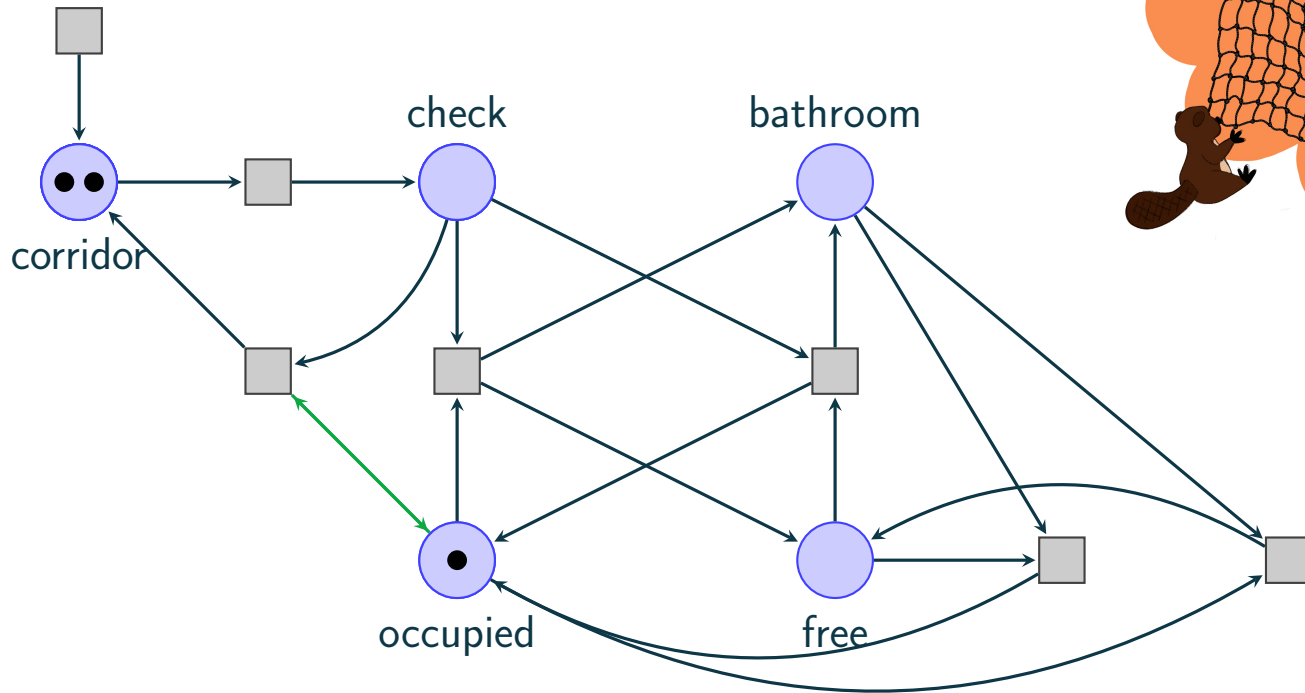
Modelling (reachability problem)



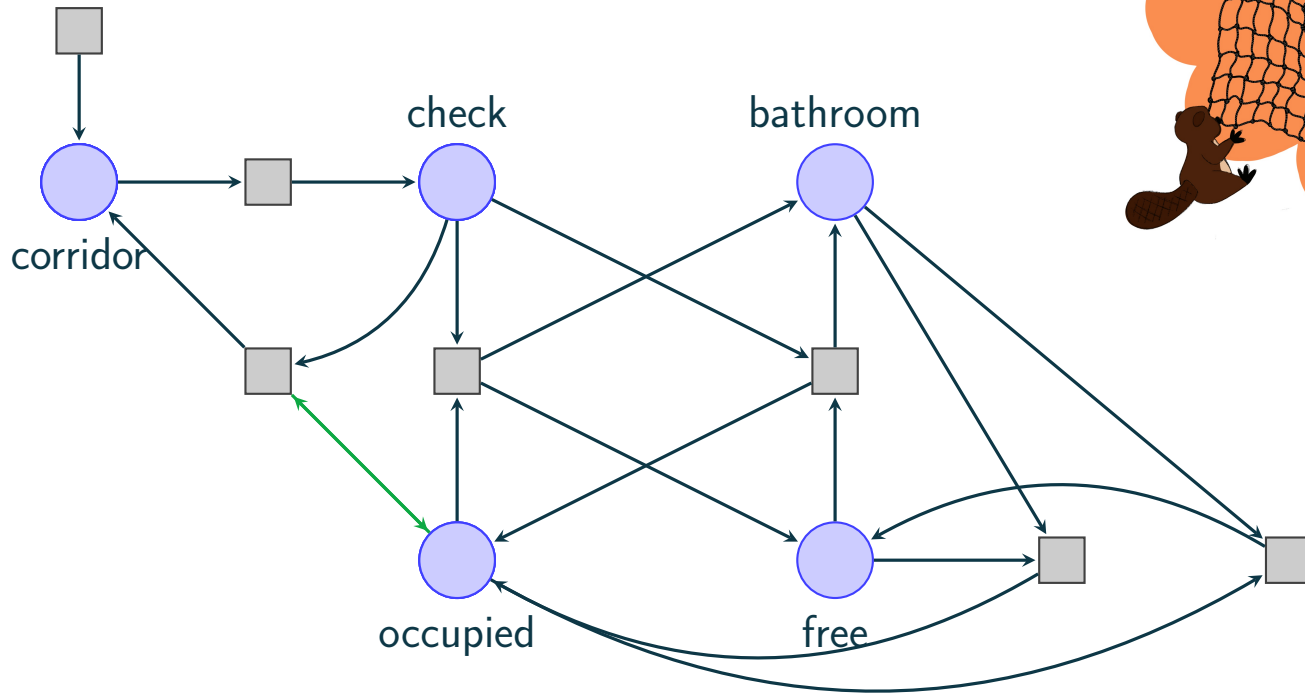
An illustration of three chipmunks fishing in a pond. One chipmunk is at the top, holding a blue fishing rod. Another is on the right, holding a net that is partially submerged in the water. A third chipmunk is at the bottom left, also holding the net. A yellow fish is visible in the water. The background is a simple landscape with a blue sky, green grass, and a small tree.



Modelling (reachability problem)



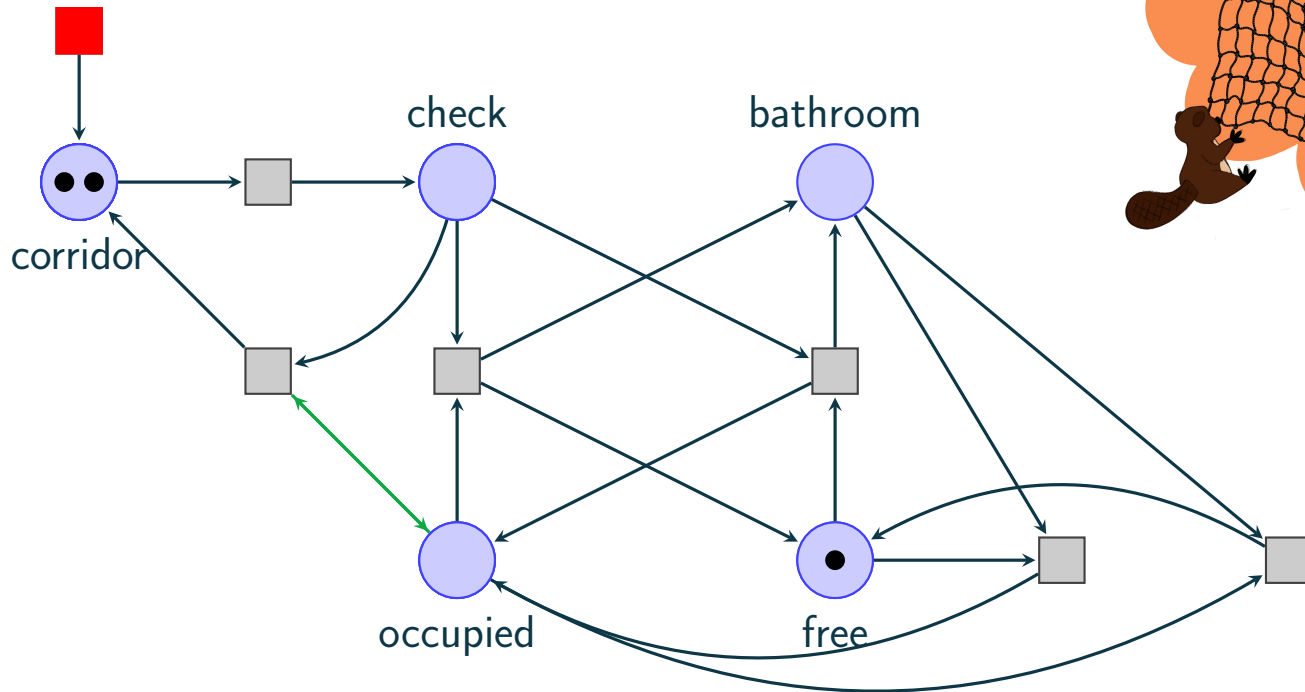
Modelling (reachability problem)



An illustration of three chipmunks fishing in a pond. One chipmunk is at the top left, another is at the bottom left, and a third is on the right side, all holding a large fishing net. A yellow fish is visible in the blue water. The background is a simple landscape with a blue sky and green grass.

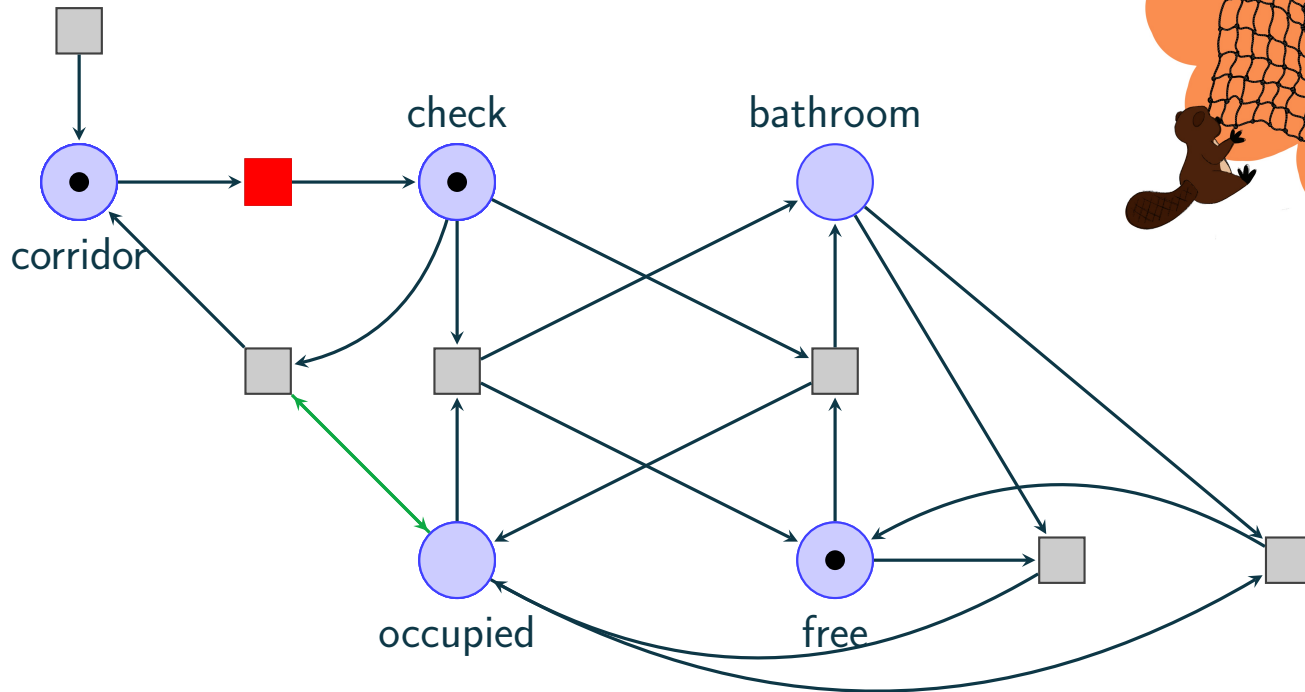


Modelling (reachability problem)



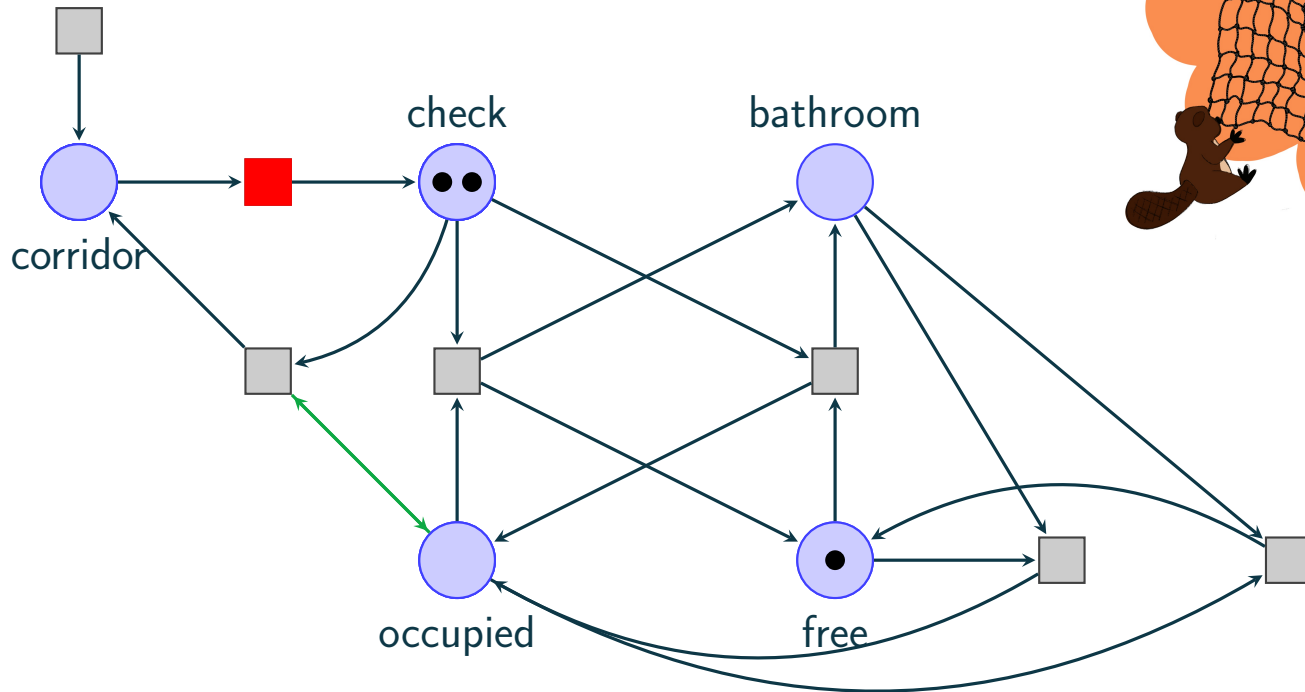
bathroom is free

Modelling (reachability problem)



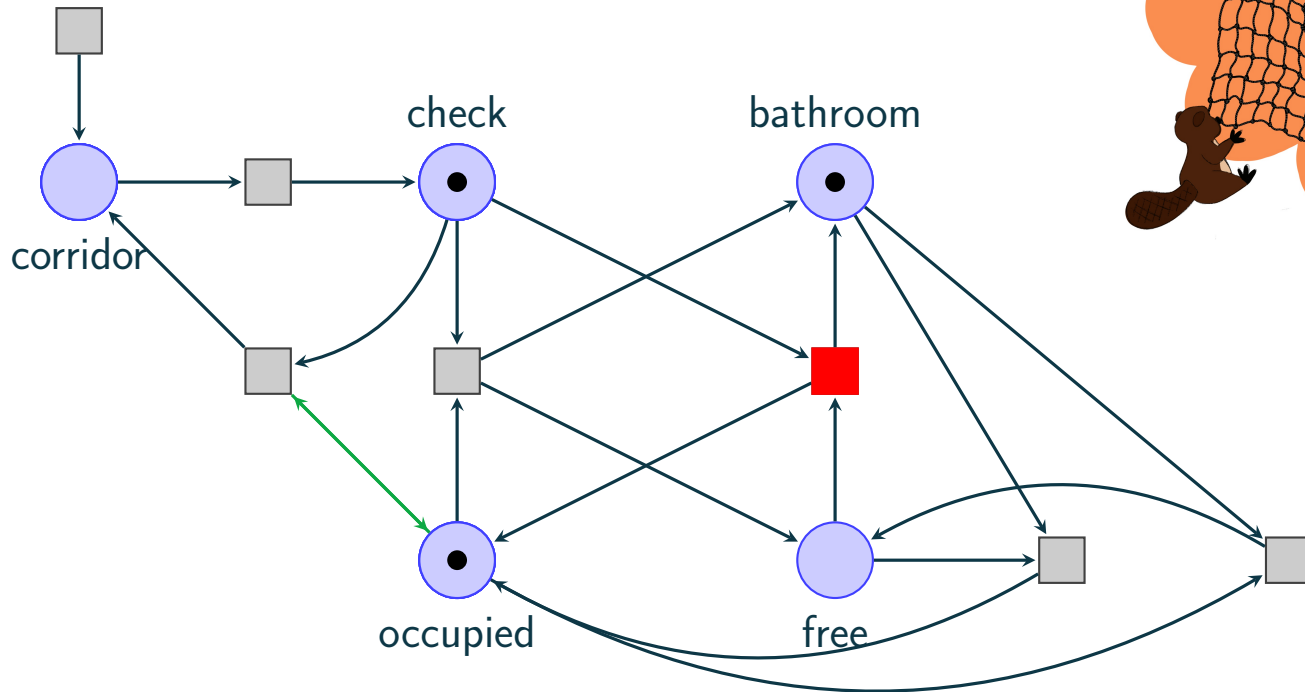
bathroom is free

Modelling (reachability problem)



bathroom is free

Modelling (reachability problem)

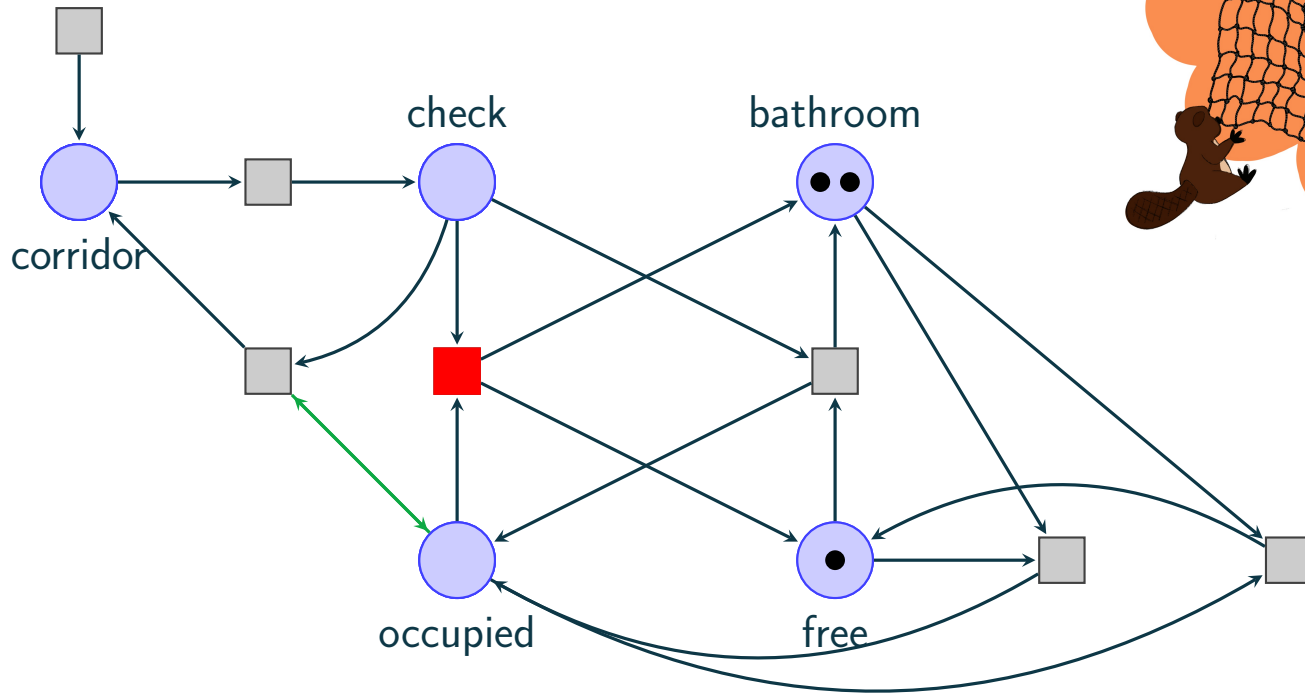


bathroom is free

An illustration of three chipmunks fishing in a pond. One chipmunk is at the top left, another is at the bottom left, and a third is on the right side, all holding a large fishing net. A yellow fish is visible in the blue water. The background is a simple landscape with a blue sky and green grass.



Modelling (reachability problem)



bathroom is free

an error found

It can be fixed, but this way we only detect errors

Modelling 2 (soundness problem)

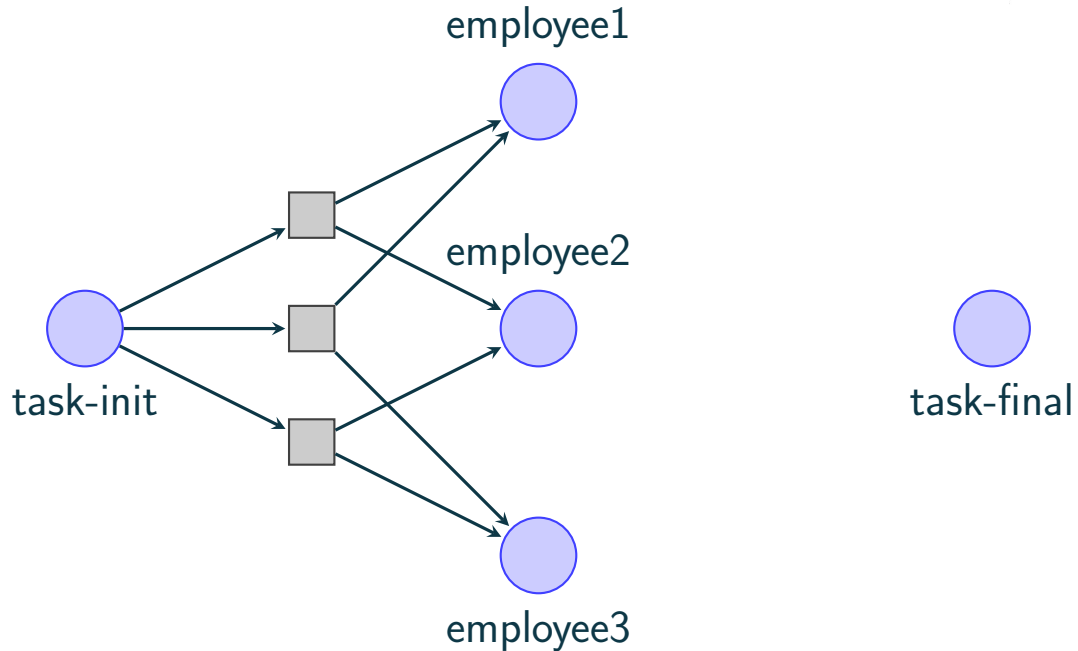
Suppose the administration wants to distribute tasks



Modelling 2 (soundness problem)



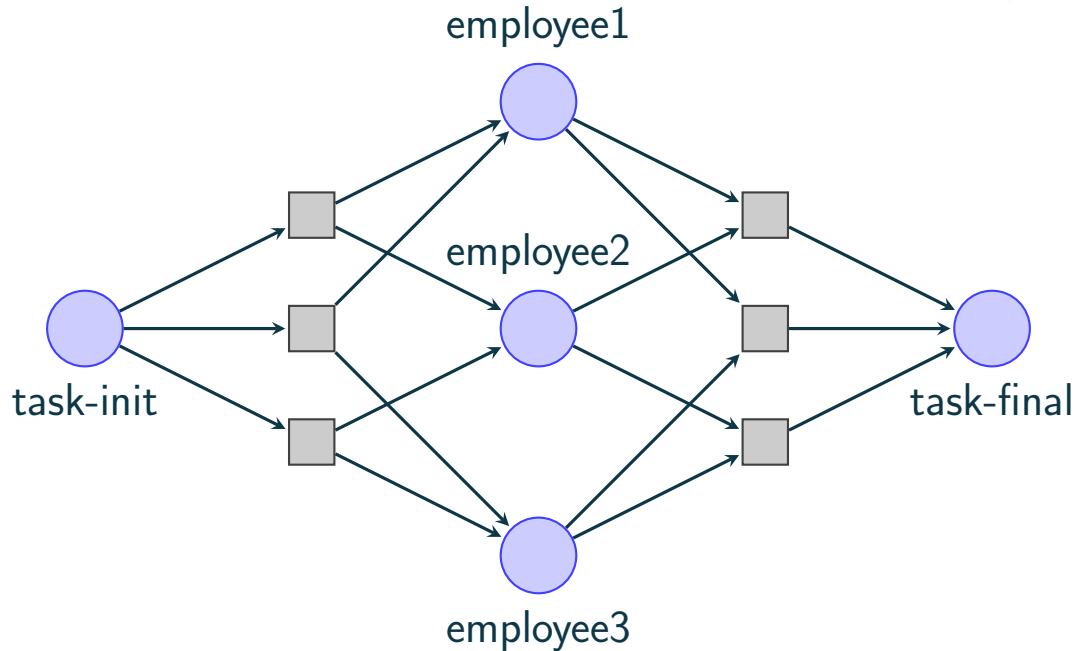
Suppose the administration wants to distribute tasks



Modelling 2 (soundness problem)



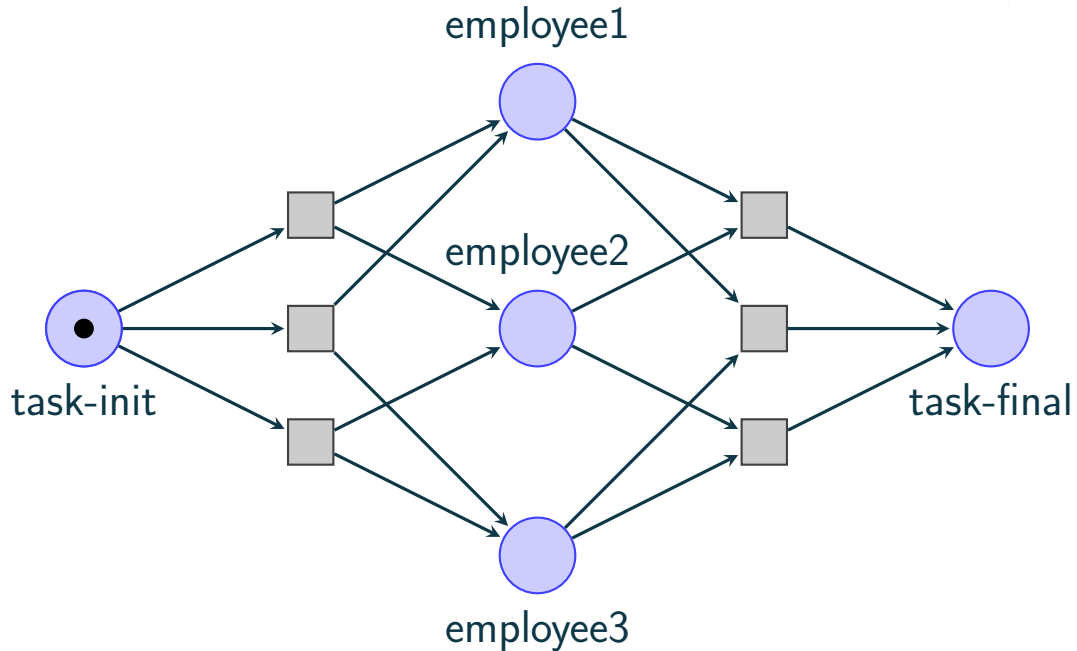
Suppose the administration wants to distribute tasks



Modelling 2 (soundness problem)



Suppose the administration wants to distribute tasks

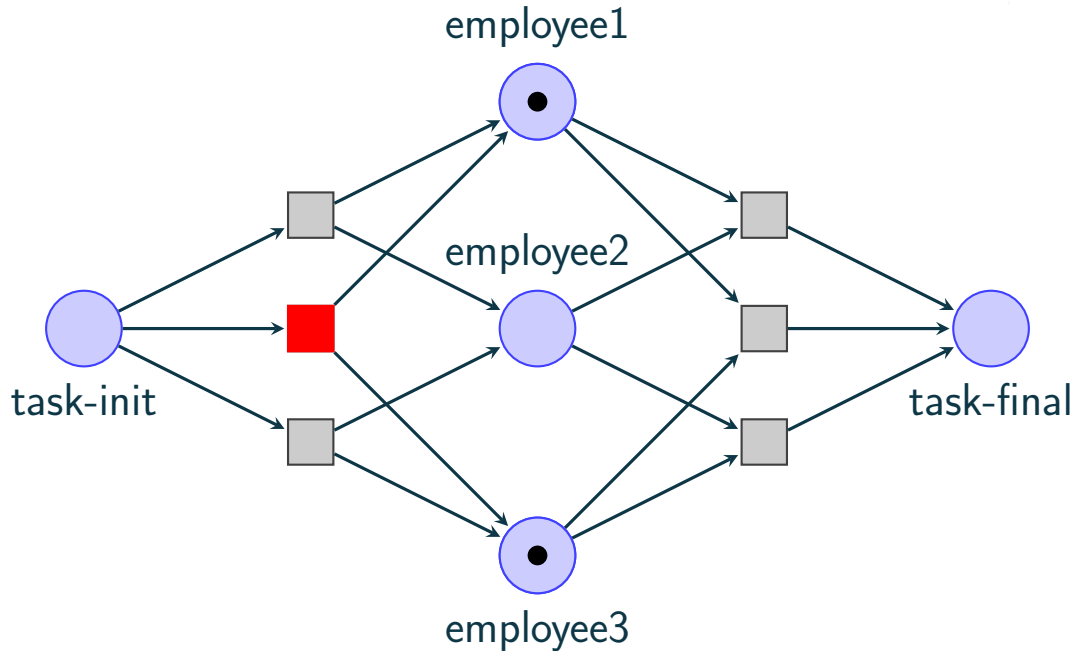


- One can check it's sound for $k = 1$

Modelling 2 (soundness problem)



Suppose the administration wants to distribute tasks

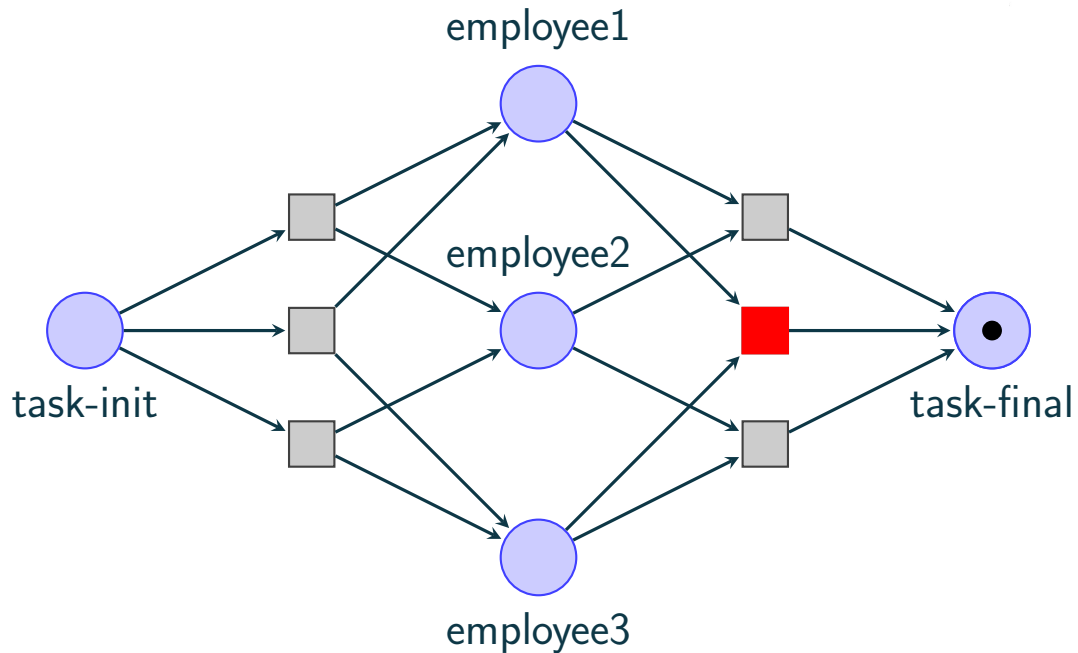


- One can check it's sound for $k = 1$

Modelling 2 (soundness problem)



Suppose the administration wants to distribute tasks

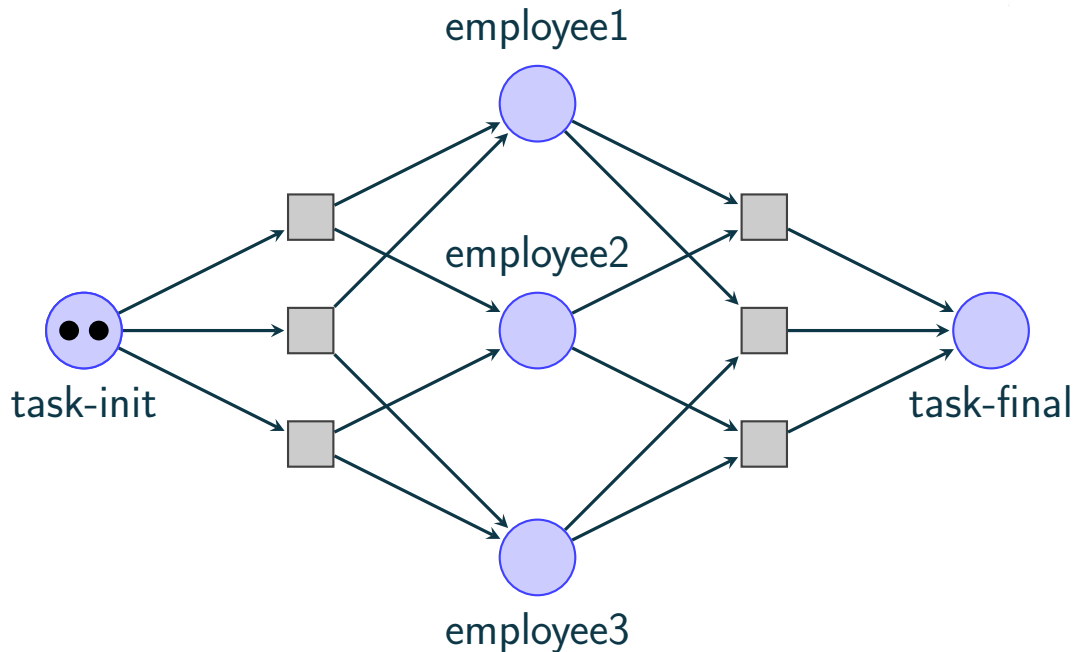


- One can check it's sound for $k = 1$

Modelling 2 (soundness problem)



Suppose the administration wants to distribute tasks

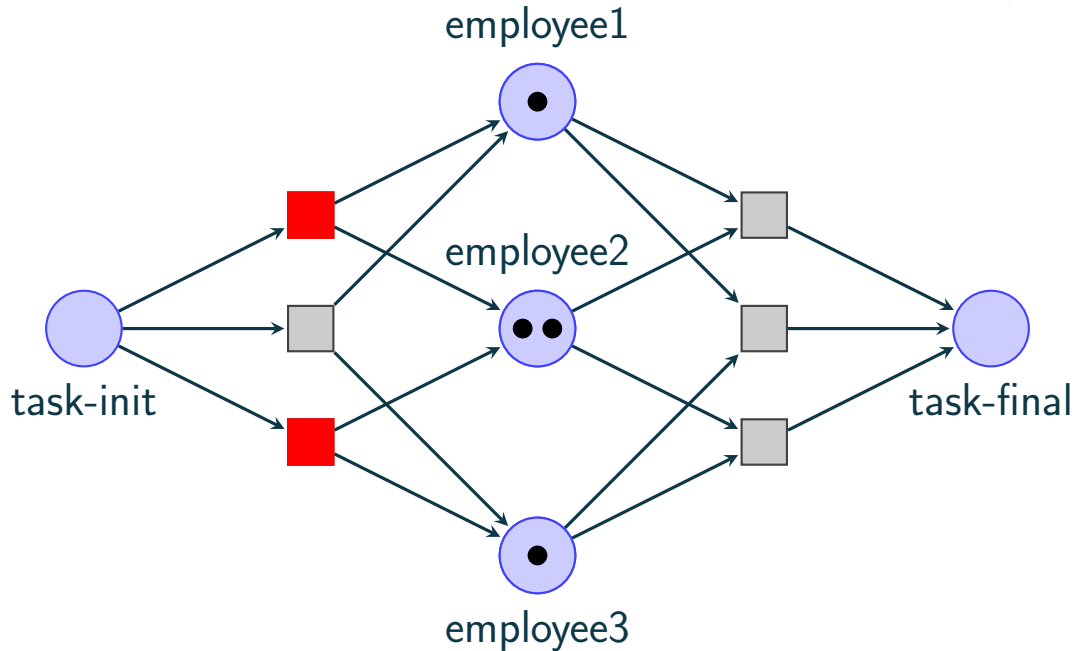


- One can check it's sound for $k = 1$
- But not for $k = 2$

Modelling 2 (soundness problem)



Suppose the administration wants to distribute tasks

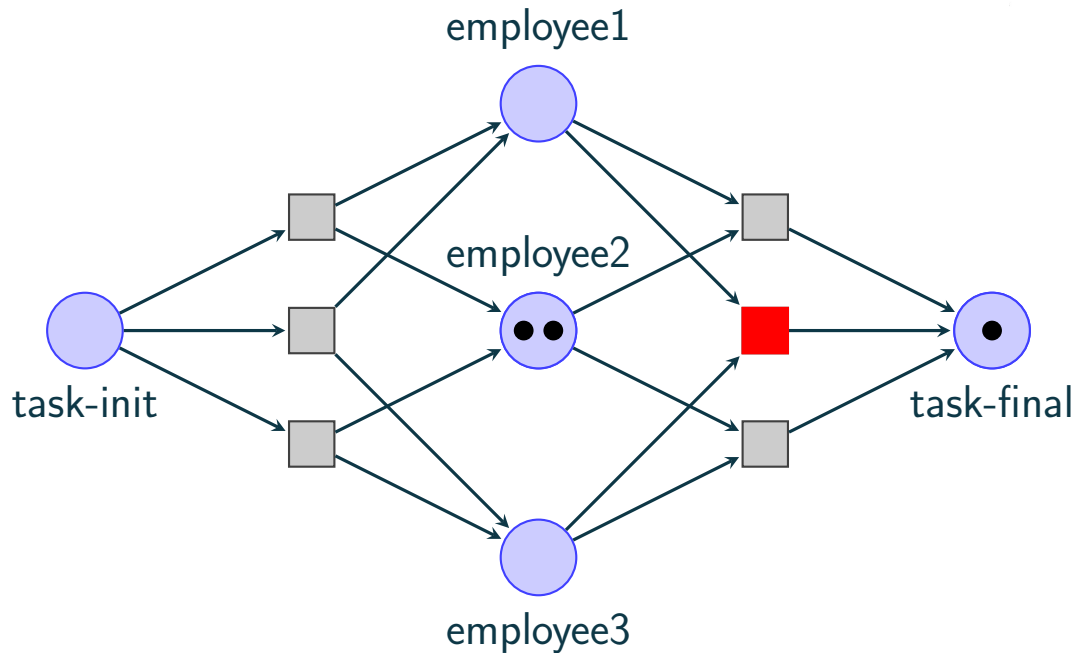


- One can check it's sound for $k = 1$
- But not for $k = 2$

Modelling 2 (soundness problem)



Suppose the administration wants to distribute tasks

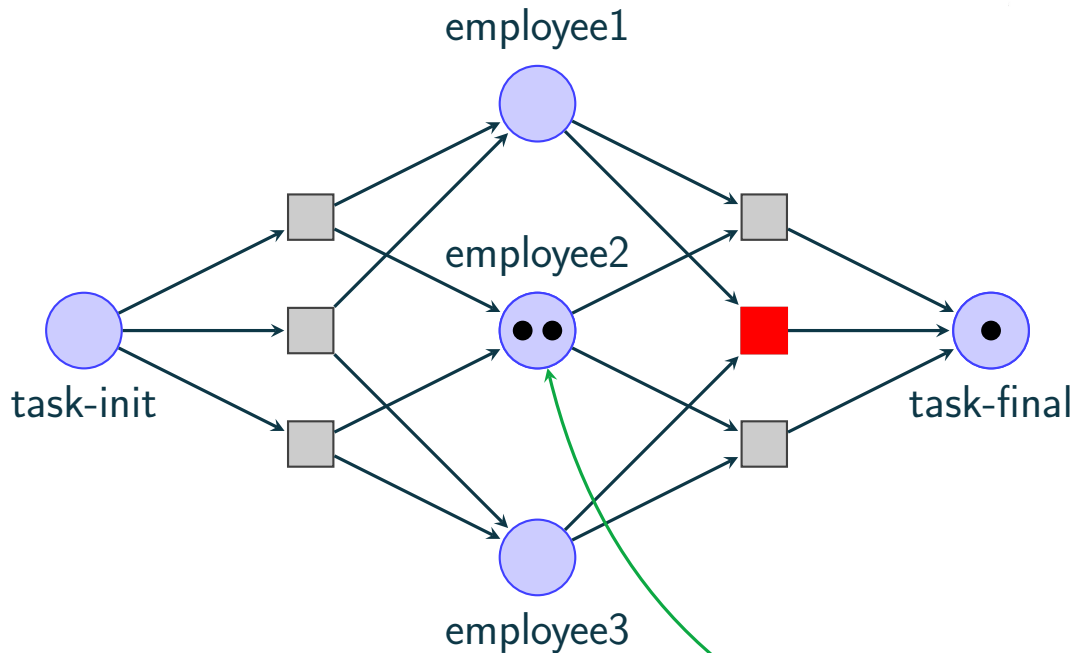


- One can check it's sound for $k = 1$
- But not for $k = 2$

Modelling 2 (soundness problem)



Suppose the administration wants to distribute tasks



- One can check it's sound for $k = 1$
- But not for $k = 2$

Results



Theorem (Czerwiński, Lasota, Lazić, Leroux, M. 2019)

The reachability problem is nonelementary

Results

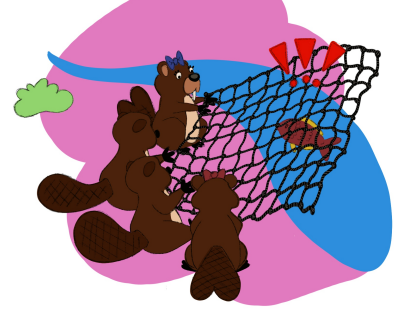


Theorem (Czerwiński, Lasota, Lazić, Leroux, M. 2019)

The reachability problem is nonelementary

In nicer words: the shortest path witnessing reachability can be $2^{2^{\dots}}$ long

Results



Theorem (Czerwiński, Lasota, Lazić, Leroux, M. 2019)

The reachability problem is nonelementary

In nicer words: the shortest path witnessing reachability can be $2^{2^{\dots}}$ long

Theorem (Blondin, M., Offtermatt 2022)

The soundness problems are EXPSPACE-complete or PSPACE-complete

Results



Theorem (Czerwiński, Lasota, Lazić, Leroux, M. 2019)

The reachability problem is nonelementary

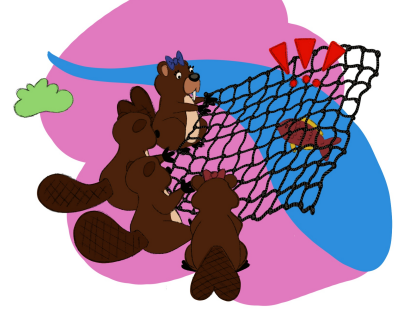
In nicer words: the shortest path witnessing reachability can be $2^{2^{\dots}}$ long

Theorem (Blondin, M., Offtermatt 2022)

The soundness problems are EXPSPACE-complete or PSPACE-complete

In nicer words: to check soundness just consider paths of length 2^{2^n} or 2^n

Results



Theorem (Czerwiński, Lasota, Lazić, Leroux, M. 2019)

The reachability problem is nonelementary

In nicer words: the shortest path witnessing reachability can be $2^{2^{\dots}}$ long

Theorem (Blondin, M., Offtermatt 2022)

The soundness problems are EXPSPACE-complete or PSPACE-complete

In nicer words: to check soundness just consider paths of length 2^{2^n} or 2^n

The second results even has a follow-up implementation

Weighted automata

For simplicity just linear recursive sequences



Weighted automata

For simplicity just linear recursive sequences

Fibonacci sequence, F_n typical definition:

$$F_{n+2} = F_{n+1} + F_n, F_0 = 0, F_1 = 1$$



Weighted automata

For simplicity just linear recursive sequences

Fibonacci sequence, F_n typical definition:

$$F_{n+2} = F_{n+1} + F_n, F_0 = 0, F_1 = 1$$

System of sequences:

$$\begin{cases} F_{n+1} = G_n, & F_0 = 0 \\ G_{n+1} = F_n + G_n, & G_0 = 1 \end{cases}$$



Weighted automata

For simplicity just linear recursive sequences

Fibonacci sequence, F_n typical definition:

$$F_{n+2} = F_{n+1} + F_n, F_0 = 0, F_1 = 1$$

System of sequences:

$$\begin{cases} F_{n+1} = G_n, & F_0 = 0 \\ G_{n+1} = F_n + G_n, & G_0 = 1 \end{cases}$$

Theorem (1)

Linear recursive sequences can be equivalently defined both ways



Weighted automata



For simplicity just linear recursive sequences

Fibonacci sequence, F_n typical definition:

$$F_{n+2} = F_{n+1} + F_n, F_0 = 0, F_1 = 1$$

System of sequences:

$$\begin{cases} F_{n+1} = G_n, & F_0 = 0 \\ G_{n+1} = F_n + G_n, & G_0 = 1 \end{cases}$$

Theorem (1)

Linear recursive sequences can be equivalently defined both ways

Theorem (2)

Linear recursive sequences are of the form $s_n = \sum_i p_i(n) \cdot \rho_i^n$

Weighted automata



For simplicity just linear recursive sequences

Fibonacci sequence, F_n typical definition:

$$F_{n+2} = F_{n+1} + F_n, F_0 = 0, F_1 = 1$$

System of sequences:

$$\begin{cases} F_{n+1} = G_n, & F_0 = 0 \\ G_{n+1} = F_n + G_n, & G_0 = 1 \end{cases}$$

Theorem (1)

Linear recursive sequences can be equivalently defined both ways

Theorem (2)

Linear recursive sequences are of the form $s_n = \sum_i p_i(n) \cdot \rho_i^n$

$$\text{e.g. } F_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^n - \frac{1}{\sqrt{5}} \left(1 - \frac{1+\sqrt{5}}{2} \right)^n$$

Polynomial recursive sequences

What if the recursive updates are given by polynomials?



Polynomial recursive sequences

What if the recursive updates are given by polynomials?

For example $a_n = n!$

$$\begin{cases} a_{n+1} = a_n \cdot b_n & a_0 = 1 \\ b_{n+1} = b_n + 1 & b_0 = 1 \end{cases}$$



Polynomial recursive sequences

What if the recursive updates are given by polynomials?

For example $a_n = n!$

$$\begin{cases} a_{n+1} = a_n \cdot b_n & a_0 = 1 \\ b_{n+1} = b_n + 1 & b_0 = 1 \end{cases}$$

Theorem (Cadilhac, M., Paperman, Pilipczuk, Sénizergues 2020)

1. $n!$ cannot be defined with one sequence
2. n^n cannot be defined at all



Polynomial recursive sequences

What if the recursive updates are given by polynomials?

For example $a_n = n!$

$$\begin{cases} a_{n+1} = a_n \cdot b_n & a_0 = 1 \\ b_{n+1} = b_n + 1 & b_0 = 1 \end{cases}$$

Theorem (Cadilhac, M., Paperman, Pilipczuk, Sénizergues 2020)

1. $n!$ cannot be defined with one sequence
2. n^n cannot be defined at all

What if we change polynomials to rational functions? i.e. $r(n) = \frac{p(n)}{q(n)}$



Polynomial recursive sequences



What if the recursive updates are given by polynomials?

For example $a_n = n!$

$$\begin{cases} a_{n+1} = a_n \cdot b_n & a_0 = 1 \\ b_{n+1} = b_n + 1 & b_0 = 1 \end{cases}$$

Theorem (Cadilhac, M., Paperman, Pilipczuk, Sénizergues 2020)

1. $n!$ cannot be defined with one sequence
2. n^n cannot be defined at all

What if we change polynomials to rational functions? i.e. $r(n) = \frac{p(n)}{q(n)}$

Conjecture (Clemente, Donten-Bury, M., Pilipczuk 2022)

All rational recursive sequences can be defined with one sequence

Polynomial recursive sequences



What if the recursive updates are given by polynomials?

For example $a_n = n!$

$$\begin{cases} a_{n+1} = a_n \cdot b_n & a_0 = 1 \\ b_{n+1} = b_n + 1 & b_0 = 1 \end{cases}$$

Theorem (Cadilhac, M., Paperman, Pilipczuk, Sénizergues 2020)

1. $n!$ cannot be defined with one sequence
2. n^n cannot be defined at all

What if we change polynomials to rational functions? i.e. $r(n) = \frac{p(n)}{q(n)}$

Conjecture (Clemente, Donten-Bury, M., Pilipczuk 2022)

All rational recursive sequences can be defined with one sequence

e.g. for $n!$ we have $a_{n+2} = \frac{(a_{n+1})^2}{a_n} + a_{n+1}$

Other problems

The **Skolem** problem:

given (a_i) is there n such that $a_n = 0$?



Other problems



The **Skolem problem**:

given (a_i) is there n such that $a_n = 0$?

This is a decision problem, but you can essentially ask:

if there is $a_n = 0$ how big can the smallest n be?

Other problems



The **Skolem problem**:

given (a_i) is there n such that $a_n = 0$?

This is a decision problem, but you can essentially ask:

if there is $a_n = 0$ how big can the smallest n be?

Other variants:

- Is $\limsup_{n \rightarrow +\infty} a_n < +\infty$? (boundedness)
- Is $\liminf_{n \rightarrow +\infty} |a_n| > 0$? (zero-isolation)

Other problems



The **Skolem problem**:

given (a_i) is there n such that $a_n = 0$?

This is a decision problem, but you can essentially ask:

if there is $a_n = 0$ how big can the smallest n be?

Other variants:

- Is $\limsup_{n \rightarrow +\infty} a_n < +\infty$? (boundedness)
- Is $\liminf_{n \rightarrow +\infty} |a_n| > 0$? (zero-isolation)

Theorem (Daviaud, Jurdziński, Lazić, M., Pérez, Worrell 2018)

Hard to explain

Other problems



The **Skolem problem**:

given (a_i) is there n such that $a_n = 0$?

This is a decision problem, but you can essentially ask:

if there is $a_n = 0$ how big can the smallest n be?

Other variants:

- Is $\limsup_{n \rightarrow +\infty} a_n < +\infty$? (boundedness)
- Is $\liminf_{n \rightarrow +\infty} |a_n| > 0$? (zero-isolation)

Theorem (Daviaud, Jurdziński, Lazić, M., Pérez, Worrell 2018)

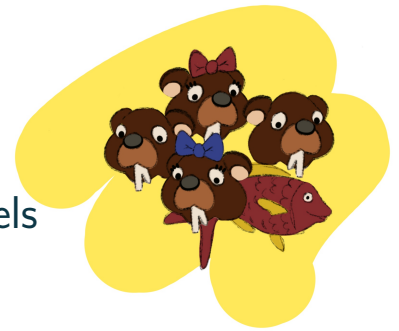
Hard to explain

Theorem (Czerwiński, Lefauchaux, M., Purser, Whiteland 2022)

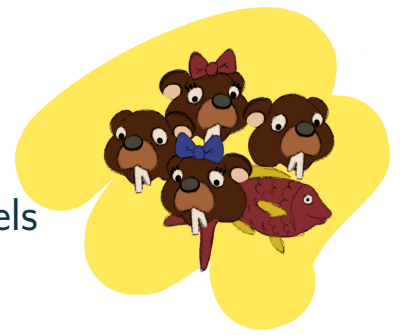
Even harder to explain

Conclusion

- Weighted automata and Petri nets are my favourite models

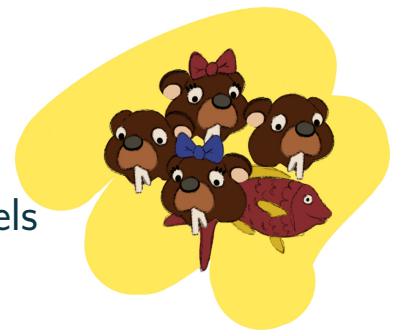


Conclusion



- Weighted automata and Petri nets are my favourite models
- They come from **formal verification**

Conclusion

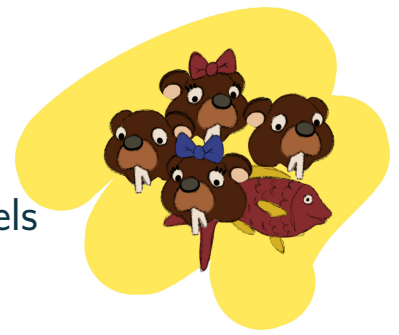


- Weighted automata and Petri nets are my favourite models
- They come from **formal verification**

Special thanks to

- **Filip Murlak** (for babysitting me)

Conclusion

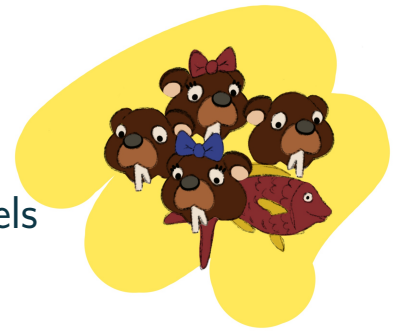


- Weighted automata and Petri nets are my favourite models
- They come from **formal verification**

Special thanks to

- **Filip Murlak** (for babysitting me)
- **Ranko Lazić** (for being my worst friend)

Conclusion



- Weighted automata and Petri nets are my favourite models
- They come from **formal verification**

Special thanks to

- **Filip Murlak** (for babysitting me)
- **Ranko Lazić** (for being my worst friend)
- **Michał Skrzypczak** (for letting me butcher his slide package)