Algorithmique et complexité des systèmes à compteurs

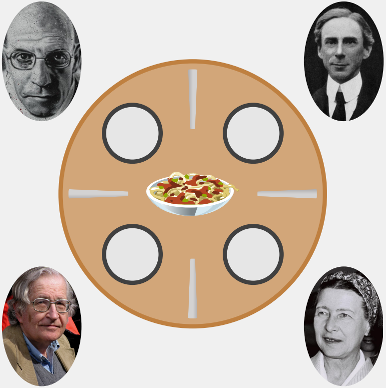
Michael Blondin

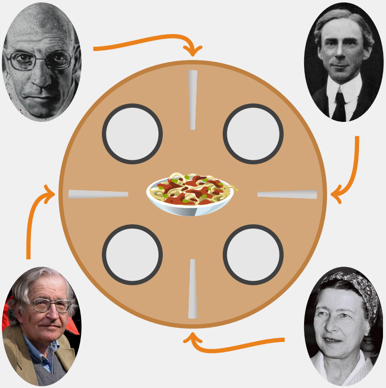
université PARIS-SACLAY

ENS CACHAN

Université de Montréal

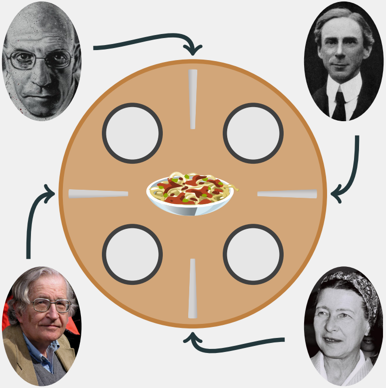
Problème des philosophes





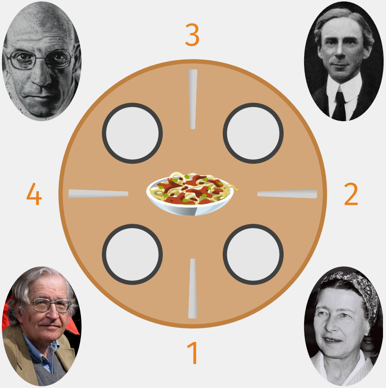
Chaque philosophe prend sa baguette de gauche

Problème des philosophes



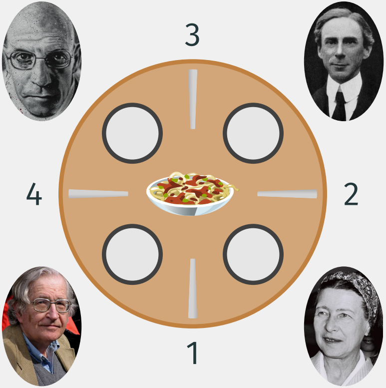
Chaque philosophe prend sa baguette de gauche

Interblocage !



Prioriser la plus « petite » baguette !

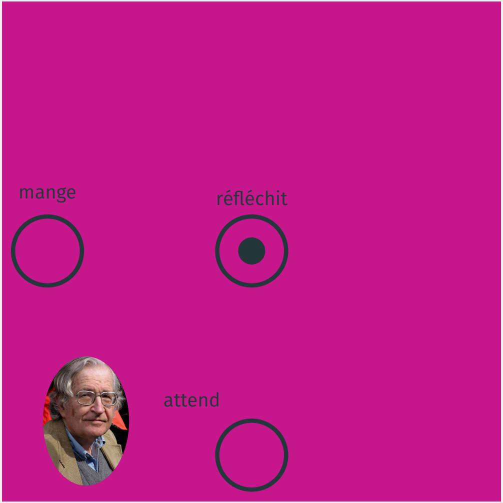
Problème des philosophes



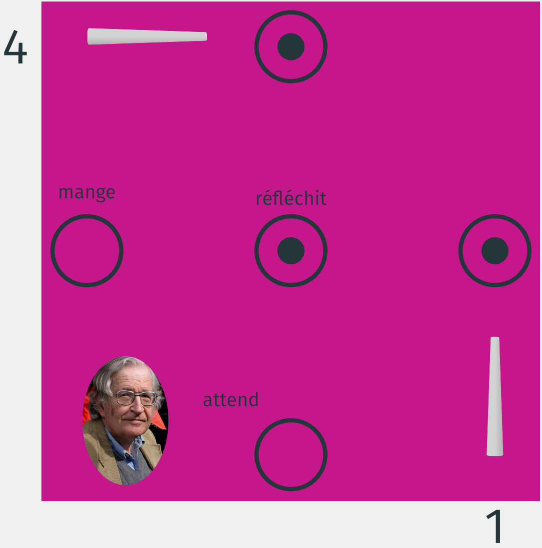
Prioriser la plus « petite » baguette !

Comment vérifier bon fonctionnement ?

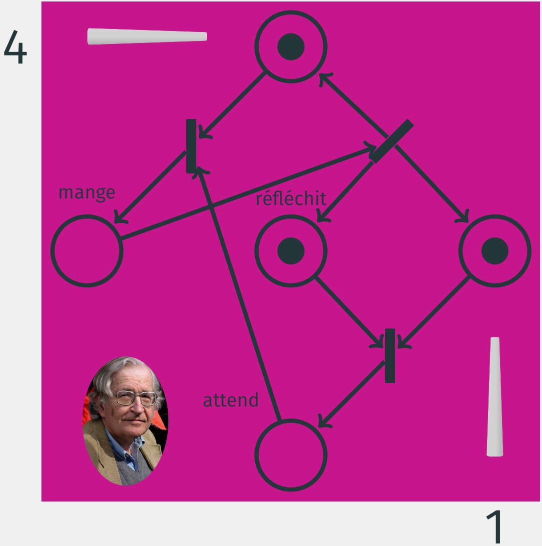
Modélisation de la procédure : réseaux de Petri



Modélisation de la procédure : réseaux de Petri

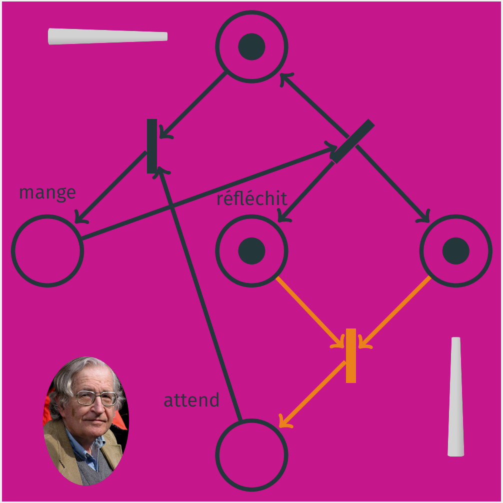


Modélisation de la procédure : réseaux de Petri



Modélisation de la procédure : réseaux de Petri

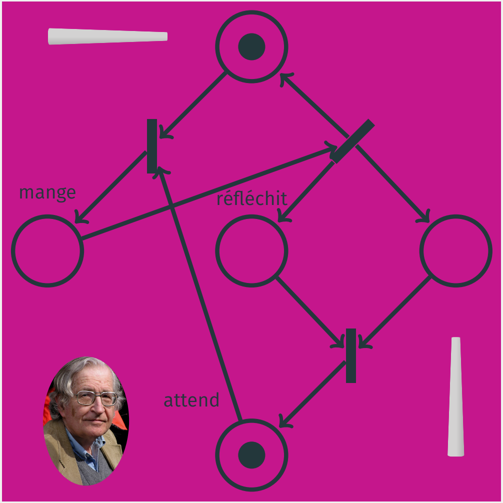
4



1

Modélisation de la procédure : réseaux de Petri

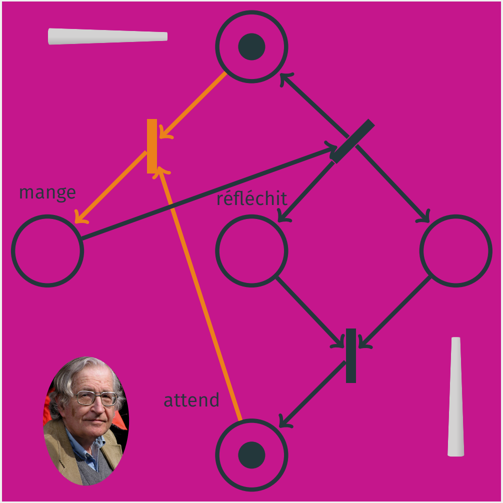
4



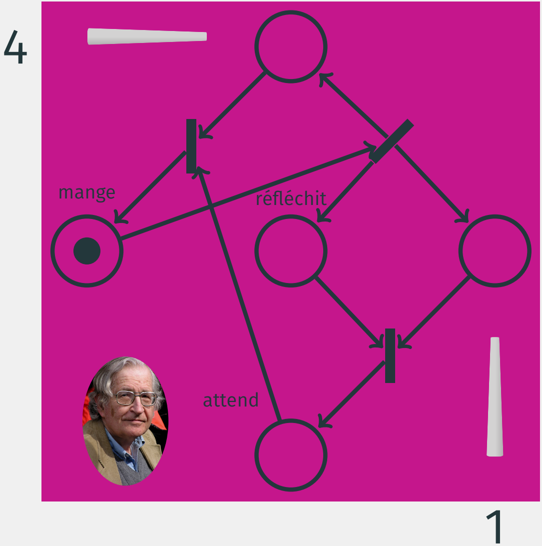
1

Modélisation de la procédure : réseaux de Petri

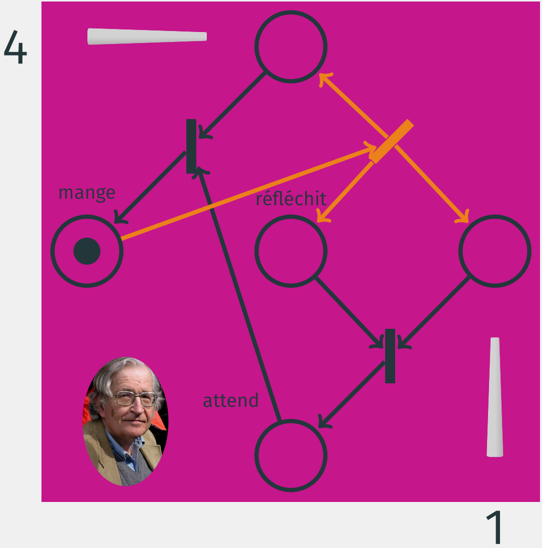
4



1

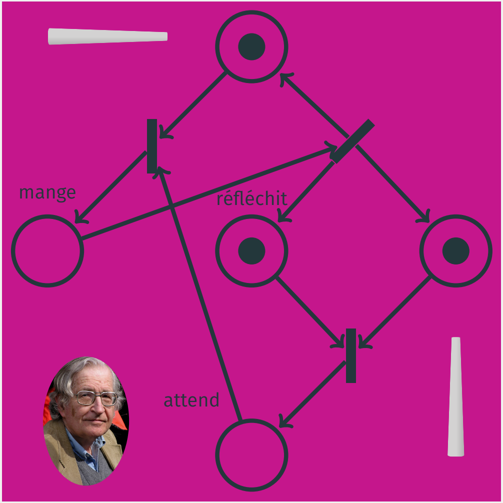

4
mange
réfléchit
attend
1

Modélisation de la procédure : réseaux de Petri



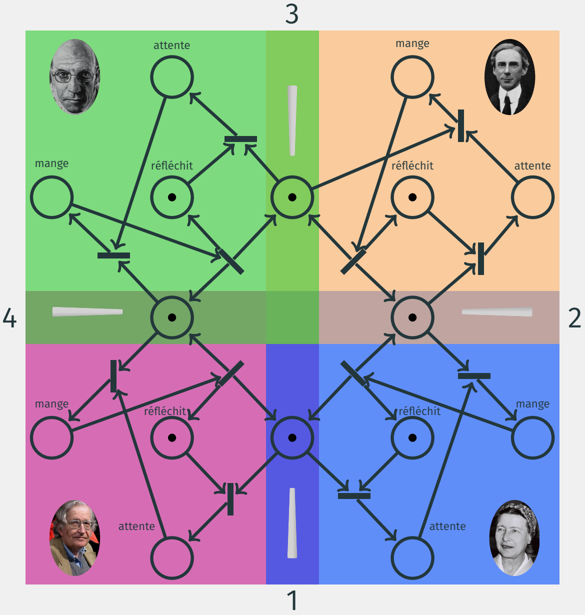
Modélisation de la procédure : réseaux de Petri

4

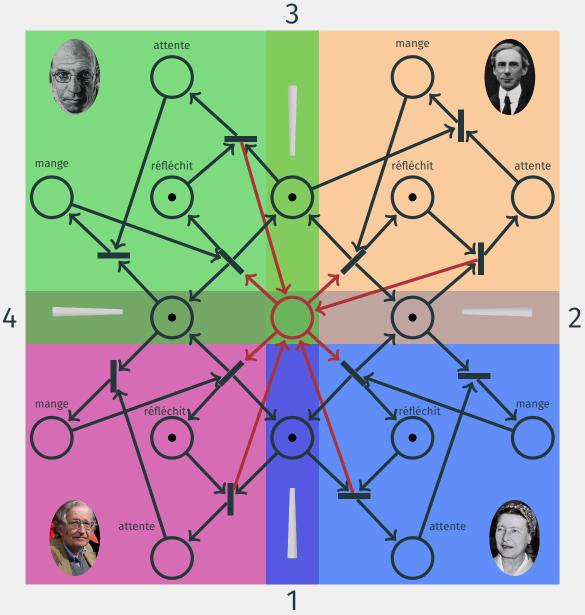


1

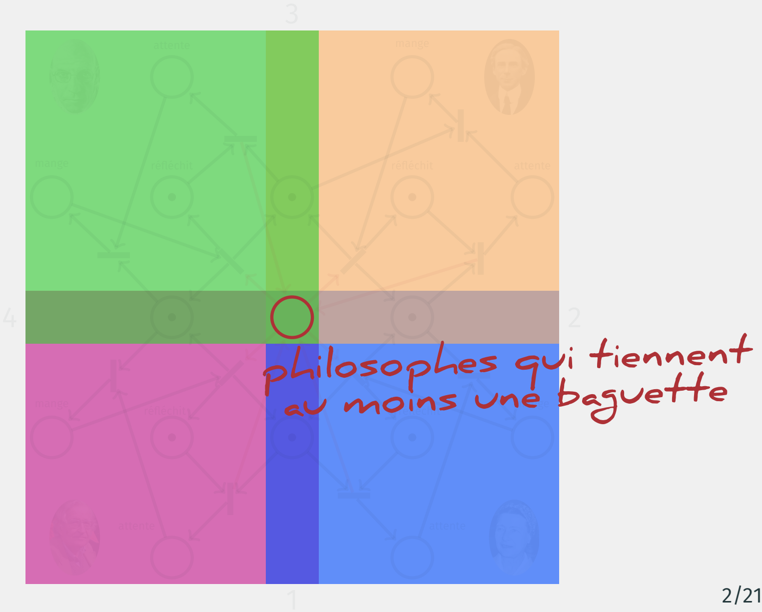
Modélisation de la procédure : réseaux de Petri



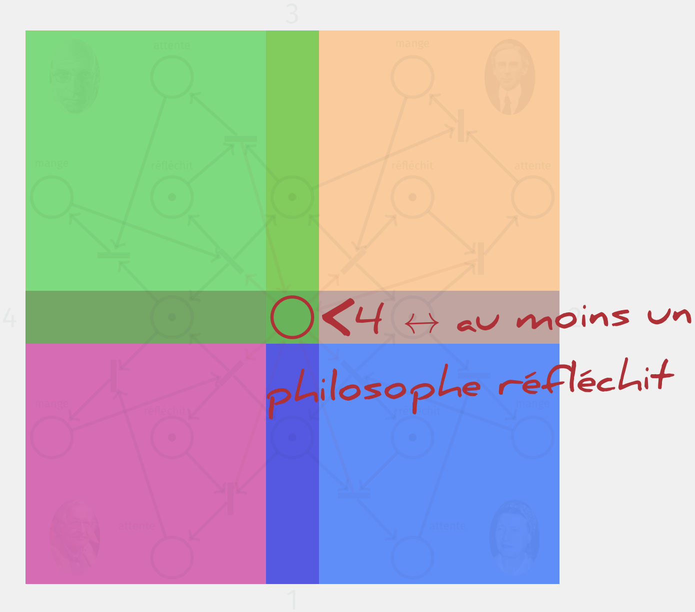
Modélisation de la procédure : réseaux de Petri



Modélisation de la procédure : réseaux de Petri

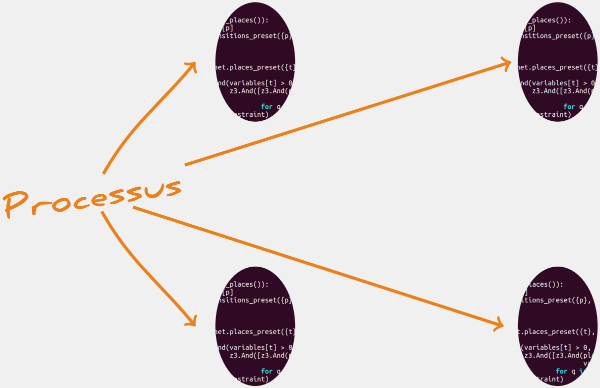


Modélisation de la procédure : réseaux de Petri

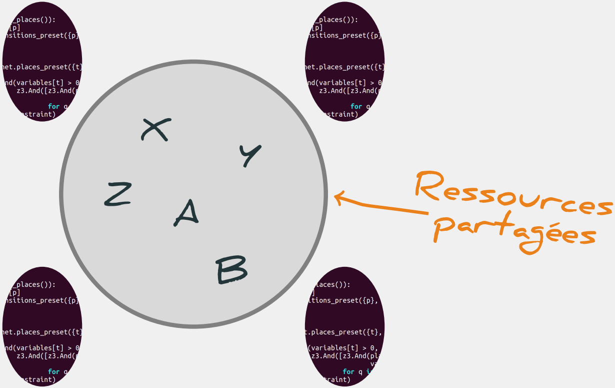


Modélisation de la procédure : réseaux de Petri

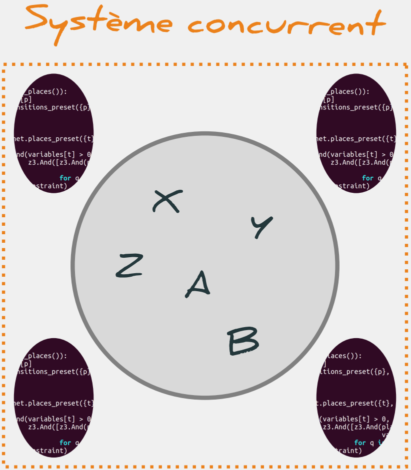
Vérification formelle : « model checking »



Vérification formelle : « model checking »

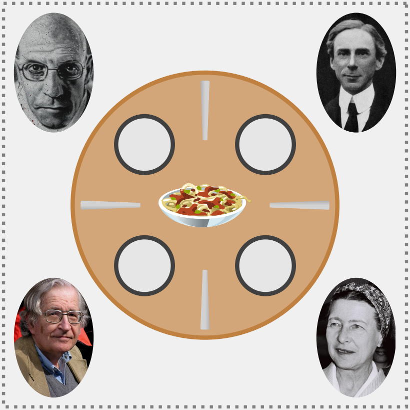


Vérification formelle : « model checking »



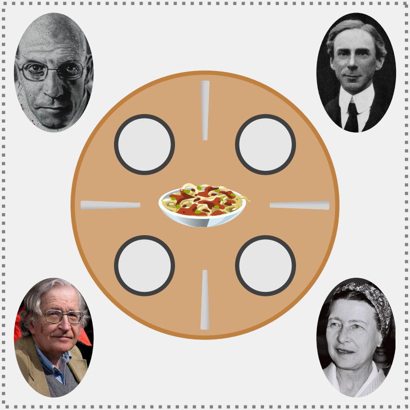
Vérification formelle : « model checking »

Système concurrent



Vérification formelle : « model checking »

Système concurrent

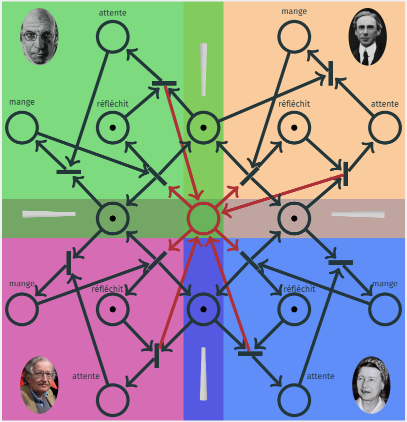


Spécification

En tout temps au moins un philosophe doit réfléchir

Vérification formelle : « model checking »

Modélisation



$$\forall \text{config} \quad \text{init} \rightarrow^* \text{config} \implies \text{config}(\bigcirc) < 4$$

Vérification de programmes

Processus 1

Processus 2

Vérification de programmes

```
tant que vrai:
  x = vrai
  tant que y: pass
  # section critique
  x = faux
```

```
tant que vrai:
▲ y = vrai
  si x alors:
    y = faux
    tant que x: pass
    goto ▲
  # section critique
  y = faux
```

Algorithme d'exclusion mutuelle de Lamport

Vérification de programmes

```
tant que vrai:
  x = vrai
  tant que y: pass
  # section critique
  x = faux
```

ressources partagées : x, y

```
  tant que vrai:
▲   y = vrai
    si x alors:
      y = faux
      tant que x: pass
      goto ▲
    # section critique
    y = faux
```

Vérification de programmes

```
tant que vrai:
  x = vrai
  tant que y: pass
  # section critique
  x = faux
```

```
tant que vrai:
▲ y = vrai
  si x alors:
    y = faux
    tant que x: pass
    goto ▲
  # section critique
  y = faux
```

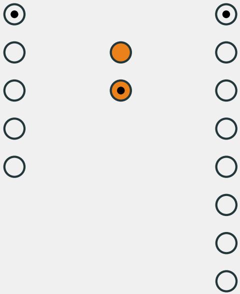
Vérification de programmes

```
tant que vrai:
  x = vrai
  tant que y: pass
  # section critique
  x = faux
```

```
tant que vrai:
▲ y = vrai
  si x alors:
    y = faux
    tant que x: pass
    goto ▲
  # section critique
  y = faux
```

Vérification de programmes

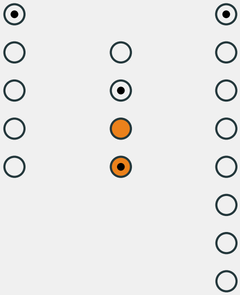
```
tant que vrai:
  x = vrai
  tant que y: pass
  # section critique
  x = faux
```



```
tant que vrai:
▲ y = vrai
  si x alors:
    y = faux
    tant que x: pass
    goto ▲
  # section critique
  y = faux
```

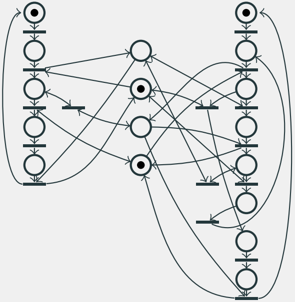
Vérification de programmes

```
tant que vrai:
  x = vrai
  tant que y: pass
  # section critique
  x = faux
```



```
tant que vrai:
▲ y = vrai
  si x alors:
    y = faux
    tant que x: pass
    goto ▲
  # section critique
  y = faux
```

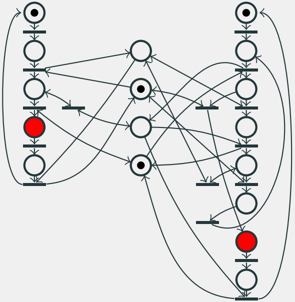
```
tant que vrai:
  x = vrai
  tant que y: pass
  # section critique
  x = faux
```



```
tant que vrai:
▲ y = vrai
  si x alors:
    y = faux
    tant que x: pass
    goto ▲
  # section critique
  y = faux
```

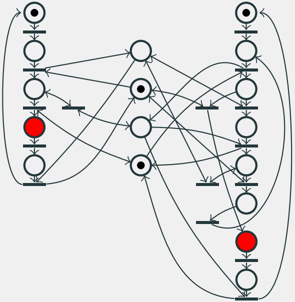
Vérification de programmes

```
tant que vrai:
  x = vrai
  tant que y: pass
  # section critique
  x = faux
```

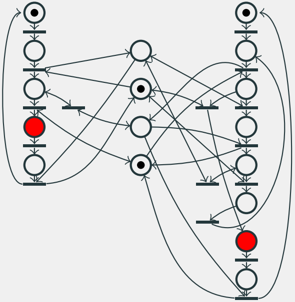


```
tant que vrai:
▲  y = vrai
   si x alors:
     y = faux
     tant que x: pass
     goto ▲
   # section critique
   y = faux
```

Vérification de programmes



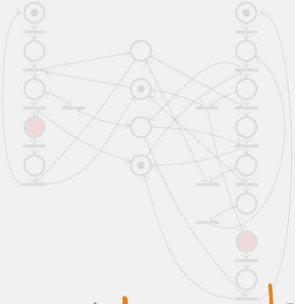
Les deux sections critiques atteintes $\iff$  ≥ 1



Les deux sections critiques atteintes $\iff$  ≥ 1

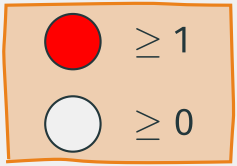
 ≥ 0

Vérification de programmes



Problème de couverture

Les deux sections critiques atteintes $\iff$



Plan

1. QCover : un nouvel outil pour le problème de couverture dans les réseaux de Petri

Plan

1. QCover : un nouvel outil pour le problème de couverture dans les réseaux de Petri

2. Problème d'accessibilité pour systèmes d'addition de vecteurs à deux compteurs est PSPACE-complet

Plan

1. QCover : un nouvel outil pour le problème de couverture dans les réseaux de Petri

2. Problème d'accessibilité pour systèmes d'addition de vecteurs à deux compteurs est PSPACE-complet

3. Décidabilité dans les systèmes de transitions bien structurés à branchement infini

Plan

1. QCover : un nouvel outil pour le problème de couverture dans les réseaux de Petri
2. Problème d'accessibilité pour systèmes d'addition de vecteurs à deux compteurs est PSPACE-complet
3. Décidabilité dans les systèmes de transitions bien structurés à branchement infini

Pistes de recherche

Plan

1 QCover : un nouvel outil pour le problème de couverture dans les réseaux de Petri

2 Problème d'accessibilité pour systèmes d'addition de vecteurs à deux compteurs est PSPACE-complet

3 Décidabilité dans les systèmes de transitions bien structurés à branchement infini

Pistes de recherche

Contributions:

Problème de couverture

Problème

Entrée :	réseau de Petri $\mathcal{N}$, marquages $\boldsymbol{m}_{\text{init}}$, $\boldsymbol{m}_{\text{cible}}$
Question :	Existe-t-il $\boldsymbol{m} \geq \boldsymbol{m}_{\text{cible}}$ accessible à partir de $\boldsymbol{m}_{\text{init}}$?

Problème de couverture

1969 Décidable (Karp & Miller)

Problème de couverture

1976 EXPSPACE-difficile (Lipton)

1969 Décidable (Karp & Miller)

Problème de couverture

1978 EXPSPACE-complet (Rackoff)

1976 EXPSPACE-difficile (Lipton)

1969 Décidable (Karp & Miller)

Problème de couverture

1996 — Décidable : « algorithme arrière » (Abdulla, Cerans, Jonsson & Tsay) (Arnold & Latteux '78)

1978 — EXPSPACE-complet (Rackoff)

1976 — EXPSPACE-difficile (Lipton)

1969 — Décidable (Karp & Miller)

Problème de couverture

- 2007 — Outil: MIST (Ganty, Meuter, Delzanno, Kalyon, Raskin & Van Begin)
- 1996 — Décidable : « algorithme arrière » (Abdulla, Cerans, Jonsson & Tsay) (Arnold & Latteux '78)
- 1978 — EXPSPACE-complet (Rackoff)
- 1976 — EXPSPACE-difficile (Lipton)
- 1969 — Décidable (Karp & Miller)

Problème de couverture

2011 — Algo. arrière : double exponentiel (Bozzelli & Ganty)

2007 — Outil : MIST (Ganty, Meuter, Delzanno, Kalyon, Raskin & Van Begin)

1996 — Décidable : « algorithme arrière » (Abdulla, Cerans, Jonsson & Tsay)
(Arnold & Latteux '78)

1978 — EXPSPACE-complet (Rackoff)

1976 — EXPSPACE-difficile (Lipton)

1969 — Décidable (Karp & Miller)

Problème de couverture

- 2014 — Outil: PETRINIZER (Esparza, Ledesma-Garza, Majumdar, Meyer & Nikšić)
- 2013 — Outil: IIC (Kloos, Majumdar, Nikšić & Piskac)
- 2012 — Outil: BFC (Kaiser, Kroening & Wahl)
- 2011 — Algo. arrière : double exponentiel (Bozzelli & Ganty)
- 2007 — Outil: MIST (Ganty, Meuter, Delzanno, Kalyon, Raskin & Van Begin)
- 1996 — Décidable : « algorithme arrière » (Abdulla, Cerans, Jonsson & Tsay) (Arnold & Latteux '78)
- 1978 — EXPSPACE-complet (Rackoff)
- 1976 — EXPSPACE-difficile (Lipton)
- 1969 — Décidable (Karp & Miller)

Problème de couverture

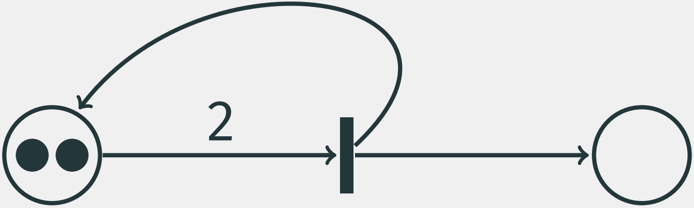
- 2015 — P-complet sous sémantique continue (Fraca & Haddad)
- 2014 — Outil: PETRINIZER (Esparza, Ledesma-Garza, Majumdar, Meyer & Nikšić)
- 2013 — Outil: IIC (Kloos, Majumdar, Nikšić & Piskac)
- 2012 — Outil: BFC (Kaiser, Kroening & Wahl)
- 2011 — Algo. arrière : double exponentiel (Bozzelli & Ganty)
- 2007 — Outil: MIST (Ganty, Meuter, Delzanno, Kalyon, Raskin & Van Begin)
- 1996 — Décidable : « algorithme arrière » (Abdulla, Cerans, Jonsson & Tsay) (Arnold & Latteux '78)
- 1978 — EXPSPACE-complet (Rackoff)
- 1976 — EXPSPACE-difficile (Lipton)
- 1969 — Décidable (Karp & Miller)

Problème de couverture

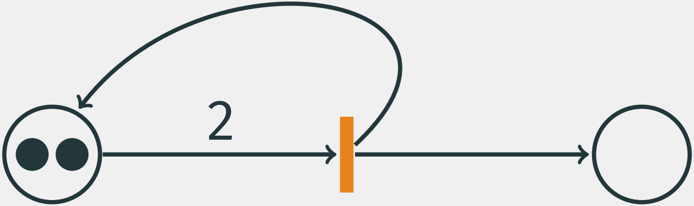
- 2016 — ★ **Outil: QCover (B., Finkel, Haase & Haddad)**
- 2015 — P-complet sous sémantique continue (Fraca & Haddad)
- 2014 — Outil: PETRINIZER (Esparza, Ledesma-Garza, Majumdar, Meyer & Nikšić)
- 2013 — Outil: IIC (Kloos, Majumdar, Nikšić & Piskac)
- 2012 — Outil: BFC (Kaiser, Kroening & Wahl)
- 2011 — Algo. arrière : double exponentiel (Bozzelli & Ganty)
- 2007 — Outil: MIST (Ganty, Meuter, Delzanno, Kalyon, Raskin & Van Begin)
- 1996 — Décidable : « algorithme arrière » (Abdulla, Cerans, Jonsson & Tsay) (Arnold & Latteux '78)
- 1978 — EXPSPACE-complet (Rackoff)
- 1976 — EXPSPACE-difficile (Lipton)
- 1969 — Décidable (Karp & Miller)

Problème de couverture

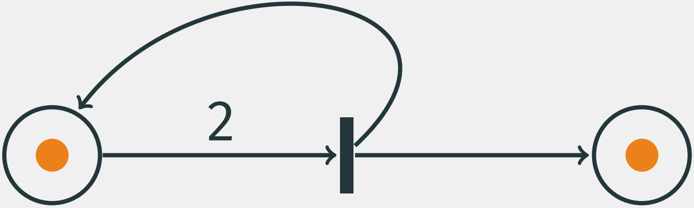
- **2016** — **Outil: QCover (B., Finkel, Haase & Haddad)**
- 2015 — P-complet sous sémantique continue (Fraca & Haddad)
- 2014 — Outil: PETRINIZER (Esparza, Ledesma-Garza, Majumdar, Meyer & Nikšić)
- 2013 — Outil: IIC (Kloos, Majumdar, Nikšić & Piskac)
- 2012 — Outil: BFC (Kaiser, Kroening & Wahl)
- 2011 — Algo. arrière : double exponentiel (Bozzelli & Ganty)
- 2007 — Outil: MIST (Ganty, Meuter, Delzanno, Kalyon, Raskin & Van Begin)
- 1996 — Décidable : « algorithme arrière » (Abdulla, Cerans, Jonsson & Tsay) (Arnold & Latteux '78)
- 1978 — EXPSPACE-complet (Rackoff)
- 1976 — EXPSPACE-difficile (Lipton)
- 1969 — Décidable (Karp & Miller)

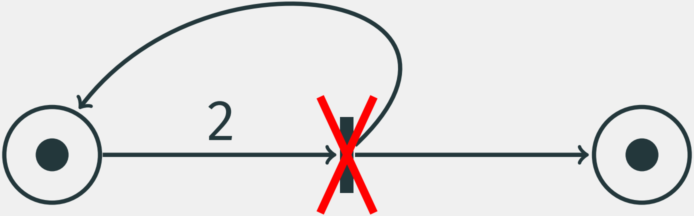

2

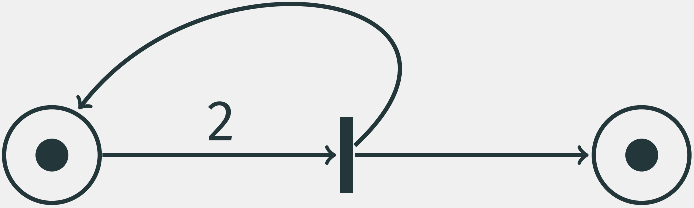
Réseaux de Petri (discrets)

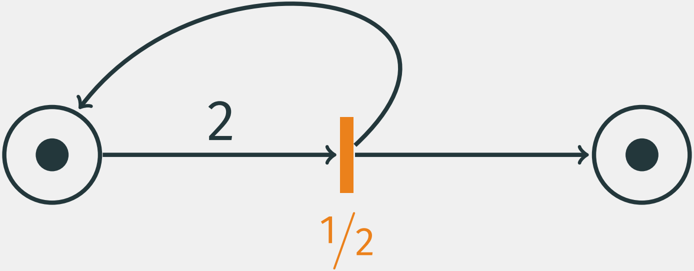


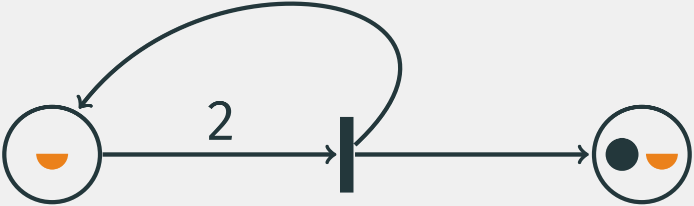
Réseaux de Petri (discrets)

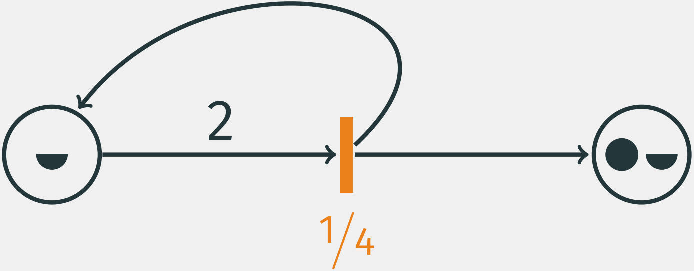


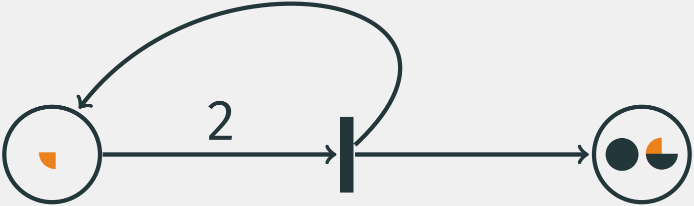

2

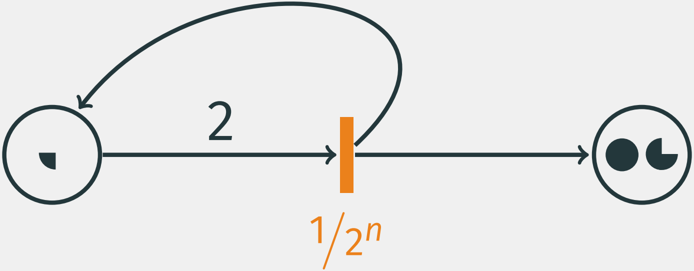

2

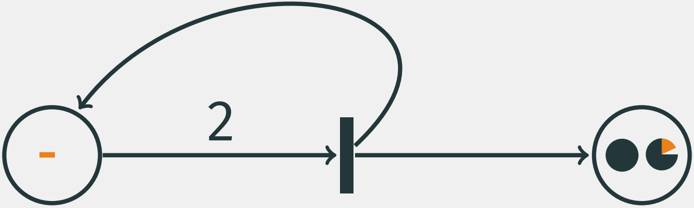

2
1/2


2


2
1/4


2


2
1/2^n


2

$$\boldsymbol{m}_{\text{cible}} \text{ est couvrable à partir de } \boldsymbol{m}_{\text{init}}$$

$$\Downarrow$$

$$\boldsymbol{m}_{\text{cible}} \text{ est } \mathbb{Q}\text{-couvrable à partir de } \boldsymbol{m}_{\text{init}}$$

$\boldsymbol{m}_{\text{cible}}$ pas couvrable à partir de $\boldsymbol{m}_{\text{init}}$

Sûreté

$$\Uparrow$$

$\boldsymbol{m}_{\text{cible}}$ pas $\mathbb{Q}$-couvrable à partir de $\boldsymbol{m}_{\text{init}}$

Algorithme arrière basé sur la $\mathbb{Q}$-couverture

Théorème **B., Finkel, Haase & Haddad '16**

$\mathbb{Q}$-couverture se traduit en formule de taille linéaire

fragment existentiel FO($\mathbb{Q}, +, <$)

Algorithme arrière basé sur la $\mathbb{Q}$-couverture

Théorème — B., Finkel, Haase & Haddad '16

$\mathbb{Q}$-couverture se traduit en formule de taille linéaire

fragment existentiel $\mathrm{FO}(\mathbb{Q}, +, <)$

$\boldsymbol{m}_{\text{cible}}$ couvrable à partir de $\boldsymbol{m}_{\text{init}} \iff \varphi(\boldsymbol{m}_{\text{init}}, \boldsymbol{m}_{\text{cible}})$

Algorithme arrière basé sur la $\mathbb{Q}$-couverture

Théorème — B., Finkel, Haase & Haddad '16

$\mathbb{Q}$-couverture se traduit en formule de taille linéaire

fragment existentiel $\mathrm{FO}(\mathbb{Q}, +, <)$

$$\varphi(\boldsymbol{x}, \boldsymbol{y}) = \exists \boldsymbol{t}, \boldsymbol{z} : \left(\boldsymbol{x}_p + 5\,\boldsymbol{t}_a + 2\,\boldsymbol{t}_b = \boldsymbol{y}_p\right) \wedge \left(\boldsymbol{z}_q > 0 \rightarrow \boldsymbol{z}_p > \boldsymbol{z}_q\right) \ldots$$

Algorithme arrière basé sur la $\mathbb{Q}$-couverture

Théorème — B., Finkel, Haase & Haddad '16

$\mathbb{Q}$-couverture se traduit en formule de taille linéaire

$$\text{fragment existentiel FO}(\mathbb{N}, +, <)$$

Meilleure approximation!

Algorithme arrière basé sur la $\mathbb{Q}$-couverture

Théorème — **B., Finkel, Haase & Haddad '16**

$\mathbb{Q}$-couverture se traduit en formule de taille linéaire

fragment existentiel FO$(\mathbb{N}, +, <)$

Tester validité $\in$ NP
(solveur SMT)

Algorithme arrière basé sur la $\mathbb{Q}$-couverture

Théorème — B., Finkel, Haase & Haddad '16

$\mathbb{Q}$-couverture se traduit en formule de taille linéaire

fragment existentiel $\mathrm{FO}(\mathbb{N}, +, <)$

$\mathbb{Q}$-couverture $\in$ PTIME
(Fraca & Haddad '13)

Couverture réseaux de Petri
sans communication
(Verma, Seidl & Schwentick '05)

Algorithme arrière basé sur la $\mathbb{Q}$-couverture

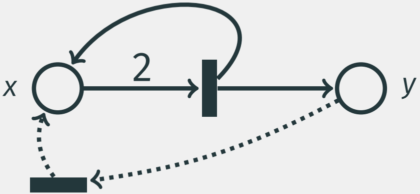
```
si m_cible n'est pas ℚ-couvrable:
  retourner faux
sinon:
```

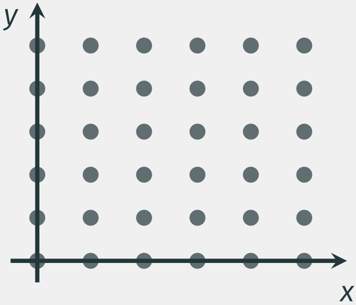
Temps polynomial

si $\boldsymbol{m}_{\text{cible}}$ n'est pas $\mathbb{Q}$-couvrable:

  retourner faux

sinon:

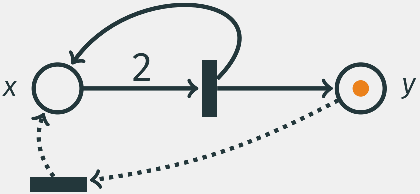


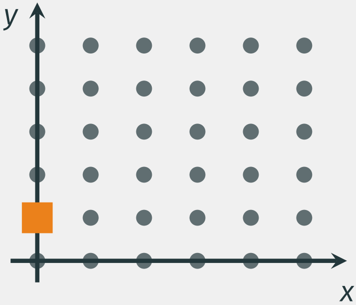


si $\boldsymbol{m}_{\text{cible}}$ n'est pas $\mathbb{Q}$-couvrable:

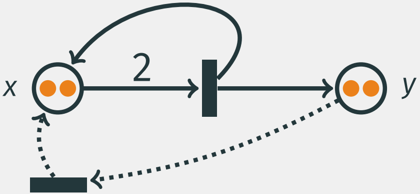
  retourner faux

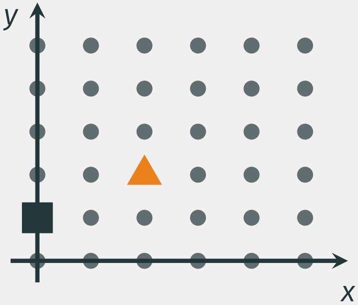
sinon:





```
si m_cible n'est pas ℚ-couvrable:
  retourner faux
sinon:
```

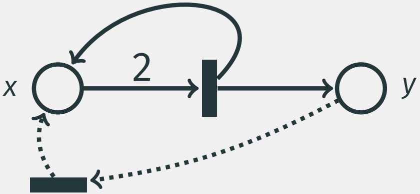


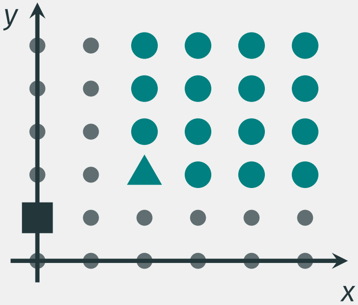


`si` $\boldsymbol{m}_{\text{cible}}$ n'est pas $\mathbb{Q}$-couvrable:

  `retourner` faux

`sinon`:



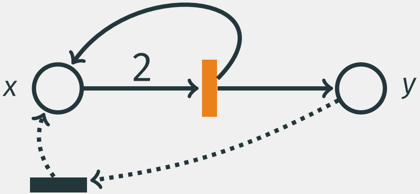


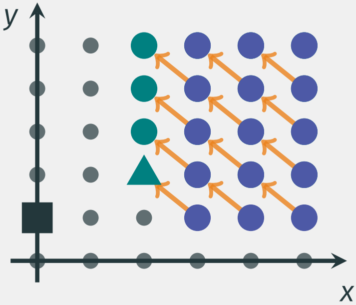
Algorithme arrière basé sur la $\mathbb{Q}$-couverture

si $\boldsymbol{m}_{\text{cible}}$ n'est pas $\mathbb{Q}$-couvrable:

  retourner faux

sinon:

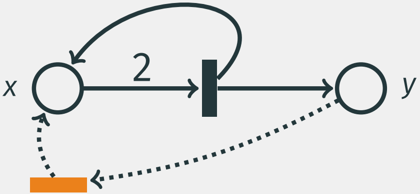


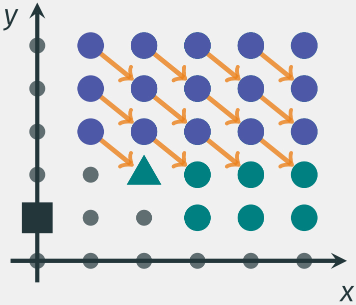


si $\boldsymbol{m}_{\text{cible}}$ n'est pas $\mathbb{Q}$-couvrable:

  retourner faux

sinon:





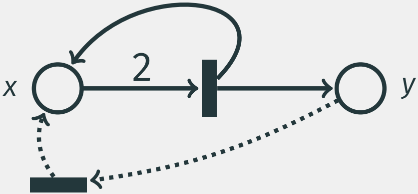
Algorithme arrière basé sur la $\mathbb{Q}$-couverture

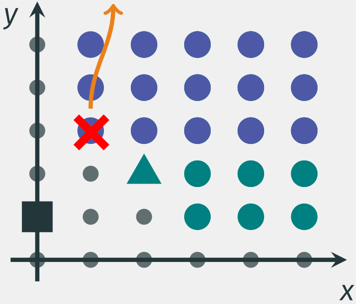
si $\boldsymbol{m}_{\text{cible}}$ n'est pas $\mathbb{Q}$-couvrable:

  retourner faux

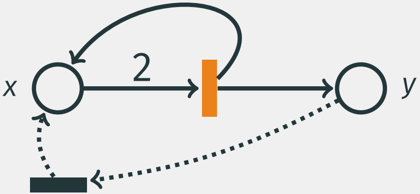
sinon:

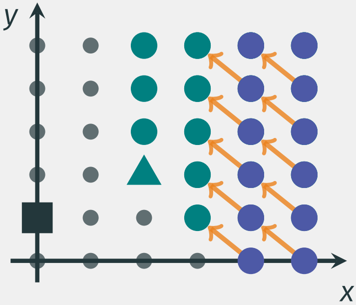
Pas Q-couvrable : $\neg\varphi(■, ●)$





```
si m_cible n'est pas ℚ-couvrable:
  retourner faux
sinon:
```



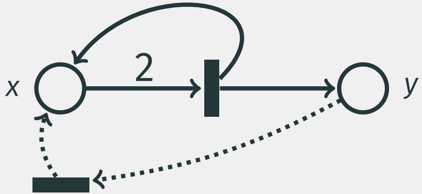


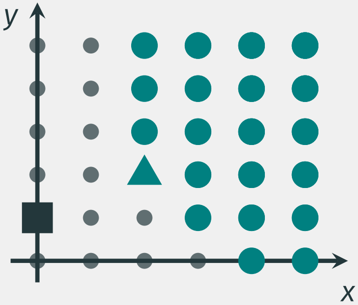
Algorithme arrière basé sur la $\mathbb{Q}$-couverture

si $\boldsymbol{m}_{\text{cible}}$ n'est pas $\mathbb{Q}$-couvrable:

  retourner faux

sinon:





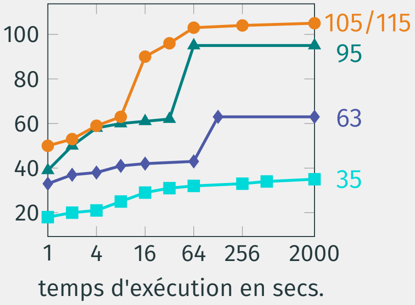
Notre implémentation : QCover

python™ + solveur SMT Z3 (Microsoft Research)

Évalué sur...

- 176 réseaux de Petri (moy. 1054 places, 8458 trans.)
- Programmes C avec fils d'exécution et mémoire partagée
- Programmes concurrents `Erlang`
- Protocoles d'exclusion mutuelle, de communication, etc.
- Provenance de messages : sys. médical et suivi de bogues

Instances prouvées sûres



QCover

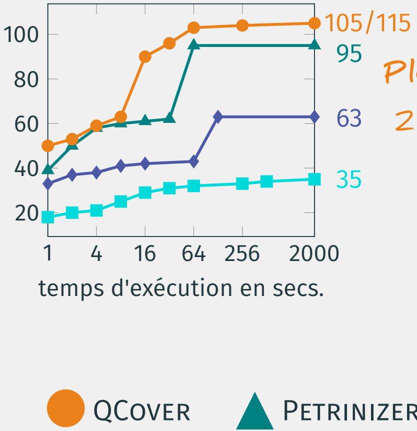
Petrinizer (Esparza *et al.* '14)

BFC (Kaiser *et al.* '12)

MIST (Ganty *et al.* '07)

B., Finkel, Haase & Haddad '16

Instances prouvées sûres



(Esparza *et al.* '14) (Kaiser *et al.* '12) (Ganty *et al.* '07)

Plus grands réseaux prouvés sûrs:

21 143 places
7 150 trans.
42 secs.

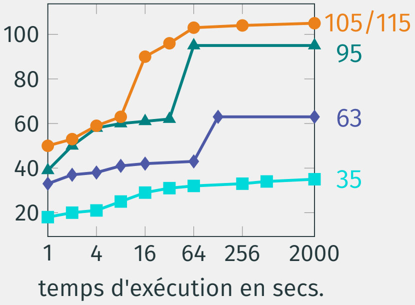
6690 places
11 934 trans.
21 secs.

754 places
27370 trans.
3 secs.

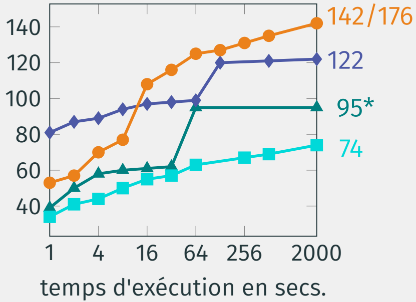
Notre implémentation : QCover

B., Finkel, Haase & Haddad '16

Instances prouvées sûres



Instances prouvées (non) sûres



QCover

Petrinizer (Esparza *et al.* '14)

BFC (Kaiser *et al.* '12)

MIST (Ganty *et al.* '07)

Sommaire

Contributions principales

- Caractérisation logique des réseaux de Petri continus
- Implémentation de l'algorithme de Fraca & Haddad
- Mise au point d'un nouvel algorithme compétitif pour le problème de couverture

Sommaire

Contributions principales

- Caractérisation logique des réseaux de Petri continus
- Implémentation de l'algorithme de Fraca & Haddad
- Mise au point d'un nouvel algorithme compétitif pour le problème de couverture

Publications

B., Finkel, Haase & Haddad

Approaching the Coverability Problem Continuously

[1] Tools and Algorithms for the Construction and Analysis of Systems, 2016

[2] ACM Transactions on Computational Logic, invité

Accessibilité et systèmes d'addition de vecteurs (VASS)

Problème d'accessibilité

Entrée : réseau de Petri $\mathcal{N}$, marquages $\boldsymbol{m}_{\text{init}}$, $\boldsymbol{m}_{\text{cible}}$

Question : $\boldsymbol{m}_{\text{cible}}$ accessible à partir de $\boldsymbol{m}_{\text{init}}$?

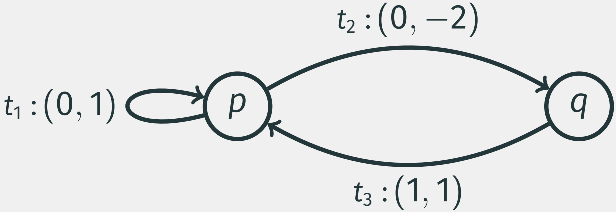
Equivalents

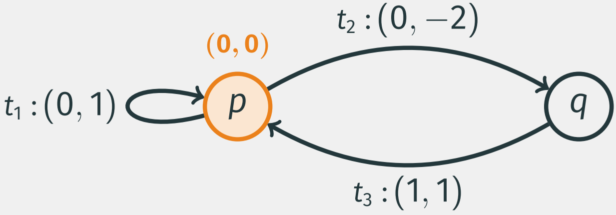
Problème d'accessibilité

Entrée : réseau de Petri $\mathcal{N}$, marquages $\boldsymbol{m}_{\text{init}}$, $\boldsymbol{m}_{\text{cible}}$

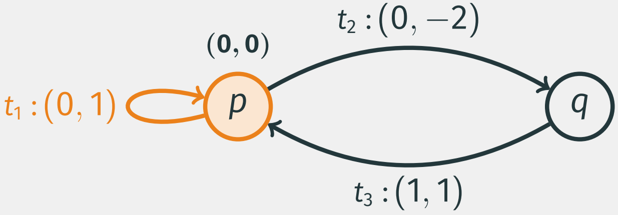
Question : $\boldsymbol{m}_{\text{cible}}$ accessible à partir de $\boldsymbol{m}_{\text{init}}$?

Accessibilité et systèmes d'addition de vecteurs (VASS)



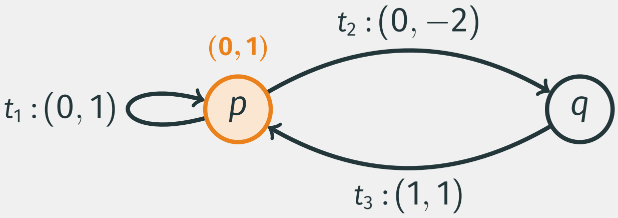


$p(0,0)$



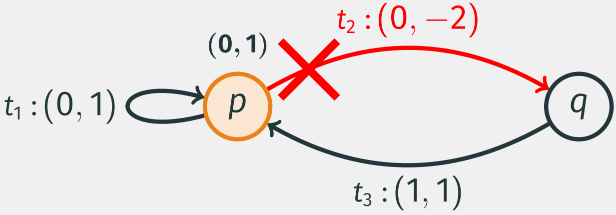
$p(0, 0)$

Accessibilité et systèmes d'addition de vecteurs (VASS)



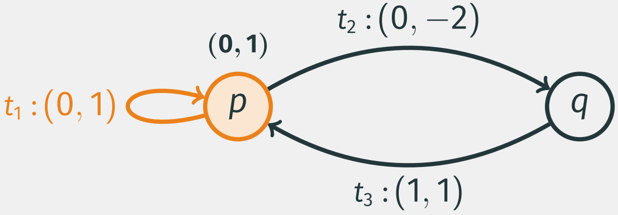
$$p(0,0) \xrightarrow{t_1} p(0,1)$$

Accessibilité et systèmes d'addition de vecteurs (VASS)



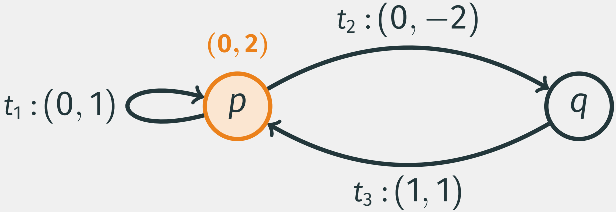
$$p(0,0) \xrightarrow{t_1} p(0,1)$$

Accessibilité et systèmes d'addition de vecteurs (VASS)



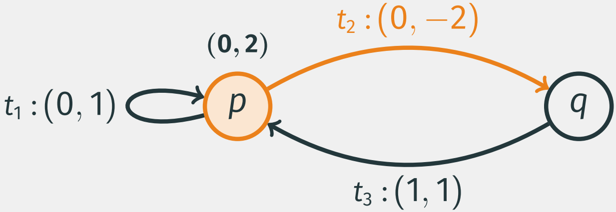
$$p(0, 0) \xrightarrow{t_1} p(0, 1)$$

Accessibilité et systèmes d'addition de vecteurs (VASS)



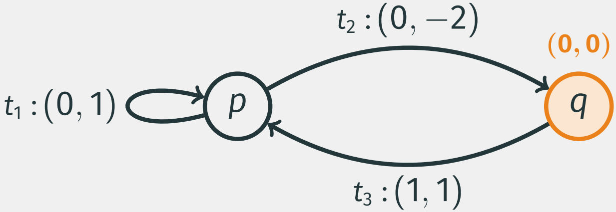
$$p(0,0) \xrightarrow{t_1} p(0,1) \xrightarrow{t_1} p(0,2)$$

Accessibilité et systèmes d'addition de vecteurs (VASS)



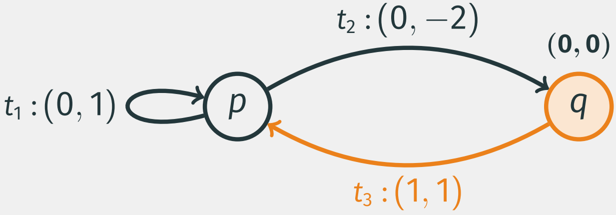
$$p(0,0) \xrightarrow{t_1} p(0,1) \xrightarrow{t_1} p(0,2)$$

Accessibilité et systèmes d'addition de vecteurs (VASS)



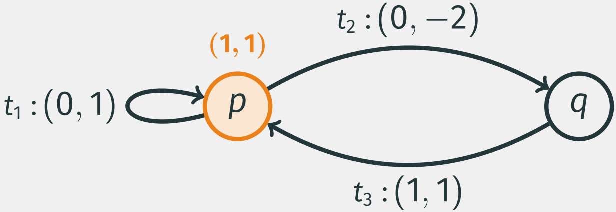
$$p(0,0) \xrightarrow{t_1} p(0,1) \xrightarrow{t_1} p(0,2) \xrightarrow{t_2} q(0,0)$$

Accessibilité et systèmes d'addition de vecteurs (VASS)



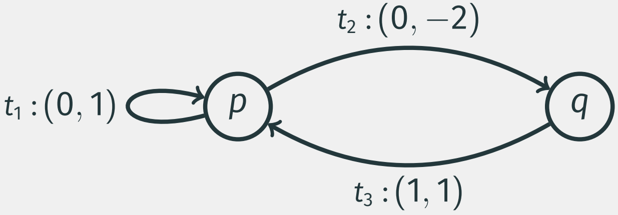
$$p(0,0) \xrightarrow{t_1} p(0,1) \xrightarrow{t_1} p(0,2) \xrightarrow{t_2} q(0,0)$$

Accessibilité et systèmes d'addition de vecteurs (VASS)



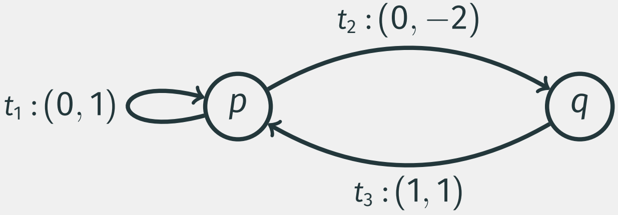
$$p(0,0) \xrightarrow{t_1} p(0,1) \xrightarrow{t_1} p(0,2) \xrightarrow{t_2} q(0,0) \xrightarrow{t_3} p(1,1)$$

Accessibilité et systèmes d'addition de vecteurs (VASS)



$$p(0,0) \xrightarrow{t_1 t_1 t_2 t_3} p(1,1)$$

Accessibilité et systèmes d'addition de vecteurs (VASS)



$$p(0,0) \longrightarrow^{*} p(1,1)$$

Problème d'accessibilité

Entrée : VASS $\mathcal{V}$, configurations $p(\boldsymbol{u}), q(\boldsymbol{v})$

Question : $p(\boldsymbol{u}) \rightarrow^* q(\boldsymbol{v})$?

Historique du problème d'accessibilité

VASS **2-VASS**

Historique du problème d'accessibilité

VASS | **2-VASS**

2015
2013
2012
2011
2009
2004
1992
1986
1982
1981
1979
EXPSPACE-difficile (Lipton) 1976

Historique du problème d'accessibilité

VASS		2-VASS
	2015	
	2013	
	2012	
	2011	
	2009	
	2004	
	1992	
	1986	
	1982	
	1981	
	1979	Décidable (Hopcroft & Pansiot) $\{\boldsymbol{v} : p(\boldsymbol{u}) \to^* q(\boldsymbol{v})\}$ semi-linéaire
EXPSPACE-difficile (Lipton)	1976	

Historique du problème d'accessibilité

VASS			2-VASS
		2015	
		2013	
		2012	
		2011	
		2009	
		2004	
		1992	
		1986	
		1982	
Décidable	(Mayr)	1981	
		1979	Décidable (Hopcroft & Pansiot)
EXPSPACE-difficile	(Lipton)	1976	$\{\boldsymbol{v} : p(\boldsymbol{u}) \to^* q(\boldsymbol{v})\}$ semi-linéaire

Historique du problème d'accessibilité

VASS			2-VASS
		2015	
		2013	
		2012	
		2011	
		2009	
		2004	
		1992	
		1986	
Décidable	(Kosaraju)	1982	
Décidable	(Mayr)	1981	
		1979	Décidable (Hopcroft & Pansiot) $\{\boldsymbol{v} : p(\boldsymbol{u}) \to^* q(\boldsymbol{v})\}$ semi-linéaire
EXPSPACE-difficile	(Lipton)	1976	

Historique du problème d'accessibilité

VASS		2-VASS
	2015	
	2013	
	2012	
	2011	
	2009	
	2004	
	1992	
	1986	2-EXPTIME (Howell, Huynh, Rosier & Yen)
Décidable (Kosaraju)	1982	
Décidable (Mayr)	1981	
	1979	Décidable (Hopcroft & Pansiot) $\{\boldsymbol{v} : p(\boldsymbol{u}) \rightarrow^* q(\boldsymbol{v})\}$ semi-linéaire
EXPSPACE-difficile (Lipton)	1976	

Historique du problème d'accessibilité

VASS		2-VASS
	2015	
	2013	
	2012	
	2011	
	2009	
	2004	
Décidable (Lambert)	1992	
	1986	2-EXPTIME (Howell, Huynh, Rosier & Yen)
Décidable (Kosaraju)	1982	
Décidable (Mayr)	1981	
	1979	Décidable (Hopcroft & Pansiot) $\{\boldsymbol{v} : p(\boldsymbol{u}) \rightarrow^* q(\boldsymbol{v})\}$ semi-linéaire
EXPSPACE-difficile (Lipton)	1976	

Historique du problème d'accessibilité

VASS		2-VASS
	2015	
	2013	
	2012	
	2011	
	2009	
	2004	$\{(\boldsymbol{u}, \boldsymbol{v}) : p(\boldsymbol{u}) \to^* q(\boldsymbol{v})\}$ semi-linéaire (Leroux & Sutre)
Décidable (Lambert)	1992	
	1986	2-EXPTIME (Howell, Huynh, Rosier & Yen)
Décidable (Kosaraju)	1982	
Décidable (Mayr)	1981	
	1979	Décidable (Hopcroft & Pansiot) $\{\boldsymbol{v} : p(\boldsymbol{u}) \to^* q(\boldsymbol{v})\}$ semi-linéaire
EXPSPACE-difficile (Lipton)	1976	

Historique du problème d'accessibilité

VASS		2-VASS
	2015	
	2013	
	2012	
Décidable (Leroux)	2011	
	2009	
	2004	$\{(\boldsymbol{u},\boldsymbol{v}) : p(\boldsymbol{u}) \to^* q(\boldsymbol{v})\}$ semi-linéaire (Leroux & Sutre)
Décidable (Lambert)	1992	
	1986	2-EXPTIME (Howell, Huynh, Rosier & Yen)
Décidable (Kosaraju)	1982	
Décidable (Mayr)	1981	
	1979	Décidable (Hopcroft & Pansiot) $\{\boldsymbol{v} : p(\boldsymbol{u}) \to^* q(\boldsymbol{v})\}$ semi-linéaire
EXPSPACE-difficile (Lipton)	1976	

Historique du problème d'accessibilité

VASS		2-VASS
	2015	
	2013	PSPACE-difficile (Fearnley & Jurdziński)
	2012	
Décidable (Leroux)	2011	
	2009	
	2004	$\{(\boldsymbol{u}, \boldsymbol{v}) : p(\boldsymbol{u}) \to^* q(\boldsymbol{v})\}$ semi-linéaire (Leroux & Sutre)
Décidable (Lambert)	1992	
	1986	2-EXPTIME (Howell, Huynh, Rosier & Yen)
Décidable (Kosaraju)	1982	
Décidable (Mayr)	1981	
	1979	Décidable (Hopcroft & Pansiot)
EXPSPACE-difficile (Lipton)	1976	$\{\boldsymbol{v} : p(\boldsymbol{u}) \to^* q(\boldsymbol{v})\}$ semi-linéaire

Historique du problème d'accessibilité

VASS		2-VASS
$\mathbf{F}_{\omega^3}$ (Leroux & Schmitz)	2015	
	2013	PSPACE-difficile (Fearnley & Jurdziński)
	2012	
Décidable (Leroux)	2011	
	2009	
	2004	$\{(\boldsymbol{u}, \boldsymbol{v}) : p(\boldsymbol{u}) \to^* q(\boldsymbol{v})\}$ semi-linéaire (Leroux & Sutre)
Décidable (Lambert)	1992	
	1986	2-EXPTIME (Howell, Huynh, Rosier & Yen)
Décidable (Kosaraju)	1982	
Décidable (Mayr)	1981	
	1979	Décidable (Hopcroft & Pansiot)
EXPSPACE-difficile (Lipton)	1976	$\{\boldsymbol{v} : p(\boldsymbol{u}) \to^* q(\boldsymbol{v})\}$ semi-linéaire

Historique du problème d'accessibilité

VASS		2-VASS
$\mathbf{F}_{\omega^3}$ (Leroux & Schmitz)	**2015**	**PSPACE-complet**
	2013	PSPACE-difficile (Fearnley & Jurdziński)
	2012	
Décidable (Leroux)	2011	
	2009	
	2004	$\{(\boldsymbol{u}, \boldsymbol{v}) : p(\boldsymbol{u}) \to^* q(\boldsymbol{v})\}$ semi-linéaire (Leroux & Sutre)
Décidable (Lambert)	1992	
	1986	2-EXPTIME (Howell, Huynh, Rosier & Yen)
Décidable (Kosaraju)	1982	
Décidable (Mayr)	1981	
	1979	Décidable (Hopcroft & Pansiot) $\{\boldsymbol{v} : p(\boldsymbol{u}) \to^* q(\boldsymbol{v})\}$ semi-linéaire
EXPSPACE-difficile (Lipton)	1976	

Accessibilité 2-VASS $\in$ PSPACE

Théorème — B., Finkel, Göller, Haase & McKenzie '15

Il existe $c \in \mathbb{N}$ tel que pour tout 2-VASS $\mathcal{V}$

$$p(\boldsymbol{u}) \rightarrow^* q(\boldsymbol{v}) \implies p(\boldsymbol{u}) \xrightarrow{\pi} q(\boldsymbol{v}) \quad \text{où} \quad |\pi| \leq c^{|\mathcal{V}|} + \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$$

Accessibilité 2-VASS $\in$ PSPACE

Théorème B., Finkel, Göller, Haase & McKenzie '15

Il existe $c \in \mathbb{N}$ tel que pour tout 2-VASS $\mathcal{V}$

$$p(\boldsymbol{u}) \to^* q(\boldsymbol{v}) \implies p(\boldsymbol{u}) \xrightarrow{\pi} q(\boldsymbol{v}) \quad \text{où} \quad |\pi| \leq c^{|\mathcal{V}|} + \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$$

Corollaire

Accessibilité 2-VASS $\in$ PSPACE

Accessibilité 2-VASS $\in$ PSPACE

Théorème B., Finkel, Göller, Haase & McKenzie '15

Il existe $c \in \mathbb{N}$ tel que pour tout 2-VASS $\mathcal{V}$

$$p(\boldsymbol{u}) \to^* q(\boldsymbol{v}) \implies p(\boldsymbol{u}) \xrightarrow{\pi} q(\boldsymbol{v}) \quad \text{où} \quad |\pi| \leq c^{|\mathcal{V}|} + \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$$

Corollaire : esquisse de preuve

exponentielle

$$p_1(\boldsymbol{u}_1) \to p_2(\boldsymbol{u}_2) \to \cdots \to p_n(\boldsymbol{u}_n)$$

Théorème — B., Finkel, Göller, Haase & McKenzie '15

Il existe $c \in \mathbb{N}$ tel que pour tout 2-VASS $\mathcal{V}$

$$p(\boldsymbol{u}) \to^* q(\boldsymbol{v}) \implies p(\boldsymbol{u}) \xrightarrow{\pi} q(\boldsymbol{v}) \quad \text{où} \quad |\pi| \leq c^{|\mathcal{V}|} + \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$$

Corollaire : esquisse de preuve

exponentielle

$$p_1(\boldsymbol{u}_1) \to p_2(\boldsymbol{u}_2) \to \cdots \to p_n(\boldsymbol{u}_n)$$

valeurs ≤ exponentielles

Accessibilité 2-VASS $\in$ PSPACE

Théorème — B., Finkel, Göller, Haase & McKenzie '15

Il existe $c \in \mathbb{N}$ tel que pour tout 2-VASS $\mathcal{V}$

$$p(\boldsymbol{u}) \rightarrow^* q(\boldsymbol{v}) \implies p(\boldsymbol{u}) \xrightarrow{\pi} q(\boldsymbol{v}) \quad \text{où} \quad |\pi| \leq c^{|\mathcal{V}|} + \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$$

Corollaire : esquisse de preuve

$$\overbrace{p_1(\boldsymbol{u}_1) \rightarrow p_2(\boldsymbol{u}_2) \rightarrow \cdots \rightarrow p_n(\boldsymbol{u}_n)}^{\text{exponentielle}}$$

encodage ≤ polynomial

Accessibilité 2-VASS $\in$ PSPACE

Théorème — B., Finkel, Göller, Haase & McKenzie '15

Il existe $c \in \mathbb{N}$ tel que pour tout 2-VASS $\mathcal{V}$

$$p(\boldsymbol{u}) \rightarrow^* q(\boldsymbol{v}) \implies p(\boldsymbol{u}) \xrightarrow{\pi} q(\boldsymbol{v}) \quad \text{où} \quad |\pi| \leq c^{|\mathcal{V}|} + \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$$

Corollaire : esquisse de preuve

$$\overbrace{p_1(\boldsymbol{u}_1) \rightarrow p_2(\boldsymbol{u}_2) \rightarrow \cdots \rightarrow p_n(\boldsymbol{u}_n)}^{\text{exponentielle}}$$

deviner à la volée

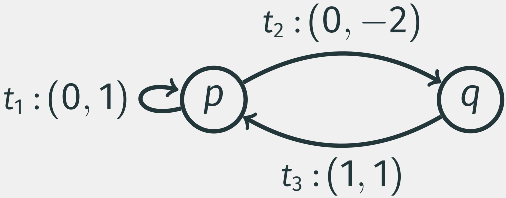
Théorème — B., Finkel, Göller, Haase & McKenzie '15

Il existe $c \in \mathbb{N}$ tel que pour tout 2-VASS $\mathcal{V}$

$$p(\boldsymbol{u}) \to^* q(\boldsymbol{v}) \implies p(\boldsymbol{u}) \xrightarrow{\pi} q(\boldsymbol{v}) \quad \text{où} \quad |\pi| \leq c^{|\mathcal{V}|} + \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$$

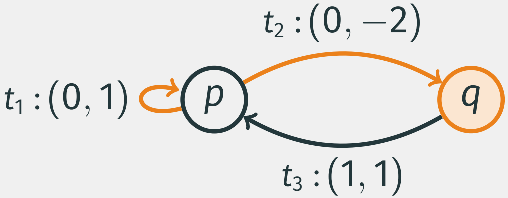
Preuve subtile et technique
(courte esquisse à venir)

Borner les chemins grâce à l'aplatissement



Chemins de p à q :

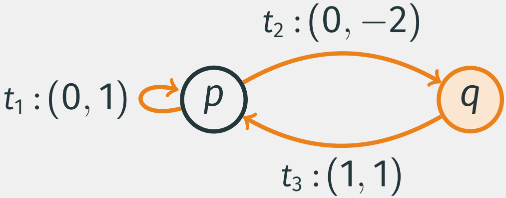
Borner les chemins grâce à l'aplatissement



Chemins de p à q :

$t_1{}^* t_2$

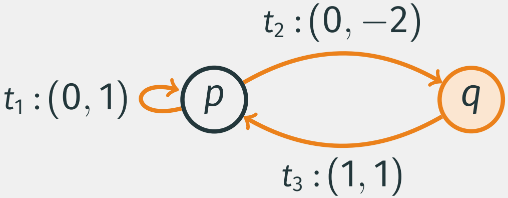
Borner les chemins grâce à l'aplatissement



Chemins de p à q :

$$t_1{}^* t_2 \; (t_3 t_1{}^* t_2)$$

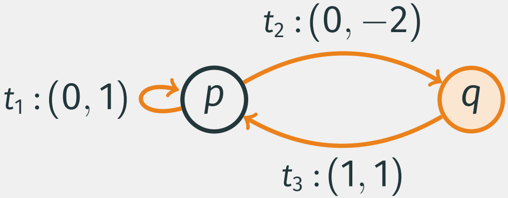
Borner les chemins grâce à l'aplatissement



Chemins de p à q :

$$t_1{}^* t_2 \; (t_3 t_1{}^* t_2) \; (t_3 t_1{}^* t_2)$$

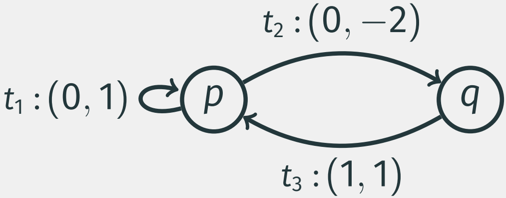
Borner les chemins grâce à l'aplatissement



Chemins de p à q :

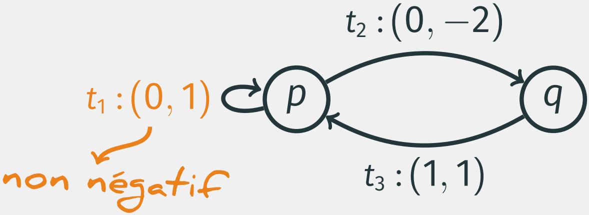
$$t_1{}^* t_2 \; (t_3 t_1{}^* t_2) \; (t_3 t_1{}^* t_2) \cdots \; (t_3 t_1{}^* t_2)$$

Borner les chemins grâce à l'aplatissement



Chemins de p à q :

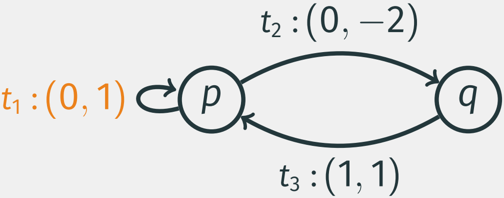
$$t_1{}^* t_2 \; (t_3 t_1{}^* t_2) \; (t_3 t_1{}^* t_2) \cdots \; (t_3 t_1{}^* t_2)$$



Chemins de p à q :

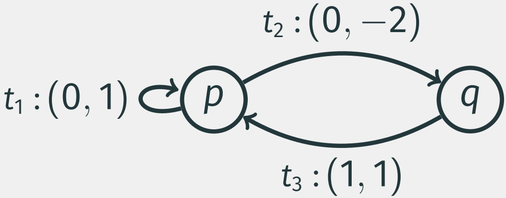
$$t_1{}^* t_2 \; (t_3 t_1{}^* t_2) \; (t_3 t_1{}^* t_2) \cdots \; (t_3 t_1{}^* t_2)$$

Borner les chemins grâce à l'aplatissement



Chemins de p à q :

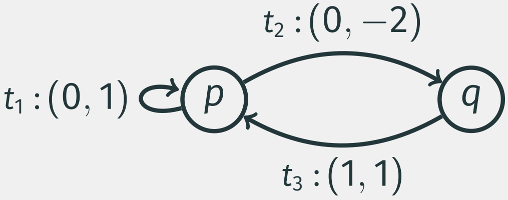
$$\boldsymbol{t_1}^* t_2 \; (t_3 \quad t_2) \; (t_3 \quad t_2) \cdots (t_3 \quad t_2)$$



Chemins de p à q :

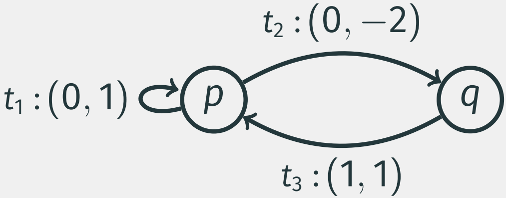
$$t_1{}^* t_2 \; (t_3 \quad t_2) \; (t_3 \quad t_2) \cdots (t_3 \quad t_2)$$

Borner les chemins grâce à l'aplatissement



Chemins de p à q :

$$t_1^* t_2 \, (t_3 \quad t_2)^*$$



Chemins de p à q :

$$t_1{}^* t_2 \; (t_3 \quad t_2)^*$$

Borner les chemins grâce à l'aplatissement

2-VASS sont aplatissables (Leroux & Sutre '04)

$$\exists S = \bigcup_{\text{finie}} \alpha_0 {\beta_1}^* \alpha_1 \cdots {\beta_k}^* \alpha_k$$

Borner les chemins grâce à l'aplatissement

2-VASS sont aplatissables **(Leroux & Sutre '04)**

$$\exists S = \bigcup_{\text{finie}} \underbrace{\alpha_0 {\beta_1}^* \alpha_1 \cdots {\beta_k}^* \alpha_k}_{\textit{schéma linéaire}}$$

Borner les chemins grâce à l'aplatissement

2-VASS sont aplatissables (Leroux & Sutre '04)

$$\exists S = \bigcup_{\text{finie}} \underbrace{\alpha_0 {\beta_1}^* \alpha_1 \cdots {\beta_k}^* \alpha_k}_{\textit{schéma linéaire}} \text{ tel que}$$

$$p(\boldsymbol{u}) \to^* q(\boldsymbol{v}) \implies p(\boldsymbol{u}) \xrightarrow{\pi \in S} q(\boldsymbol{v})$$

Borner les chemins grâce à l'aplatissement

2-VASS sont aplatissables (Leroux & Sutre '04)

$$\exists S = \bigcup_{\text{finie}} \underbrace{\alpha_0 {\beta_1}^* \alpha_1 \cdots {\beta_k}^* \alpha_k}_{\textit{schéma linéaire}} \text{ tel que}$$

$$p(\boldsymbol{u}) \to^* q(\boldsymbol{v}) \implies p(\boldsymbol{u}) \xrightarrow{\pi \in S} q(\boldsymbol{v})$$

2-VASS aplatissables par « petits » schémas linéaires

- $|\alpha_i|, |\beta_i| \quad \leq (\#\text{ états} + \|\text{transitions}\|)^{O(1)}$

Borner les chemins grâce à l'aplatissement

2-VASS sont aplatissables (Leroux & Sutre '04)

$$\exists S = \bigcup_{\text{finie}} \underbrace{\alpha_0 {\beta_1}^* \alpha_1 \cdots {\beta_k}^* \alpha_k}_{\textit{schéma linéaire}} \text{ tel que}$$

$$p(\boldsymbol{u}) \rightarrow^* q(\boldsymbol{v}) \implies p(\boldsymbol{u}) \xrightarrow{\pi \in S} q(\boldsymbol{v})$$

2-VASS aplatissables par « petits » schémas linéaires

- $|\alpha_i|, |\beta_i| \quad \leq (\text{\# états} + \|\text{transitions}\|)^{O(1)}$
- $k \quad \in O((\text{\# états})^2)$

Borner les chemins grâce à l'aplatissement

2-VASS sont aplatissables **(Leroux & Sutre '04)**

$$\exists S = \bigcup_{\text{finie}} \underbrace{\alpha_0 \beta_1{}^* \alpha_1 \cdots \beta_k{}^* \alpha_k}_{\textit{schéma linéaire}} \text{ tel que}$$

$$p(\boldsymbol{u}) \to^* q(\boldsymbol{v}) \implies p(\boldsymbol{u}) \xrightarrow{\pi \in S} q(\boldsymbol{v})$$

2-VASS aplatissables par « petits » schémas linéaires

- $|\alpha_i|, |\beta_i| \leq (\#\text{ états} + \|\text{transitions}\|)^{O(1)}$
- $k \in O((\#\text{ états})^2)$
- exposants-$*$ $\leq (\#\text{ états} + \|\text{transitions}\| + \|\boldsymbol{u}\| + \|\boldsymbol{v}\|)^{O(1)}$

Borner les chemins grâce à l'aplatissement

2-VASS sont aplatissables (Leroux & Sutre '04)

$$\exists S = \bigcup_{\text{finie}} \underbrace{\alpha_0 {\beta_1}^* \alpha_1 \cdots {\beta_k}^* \alpha_k}_{\textit{schéma linéaire}} \text{ tel que}$$

$$p(\boldsymbol{u}) \rightarrow^* q(\boldsymbol{v}) \implies p(\boldsymbol{u}) \xrightarrow{\pi \in S} q(\boldsymbol{v})$$

2-VASS aplatissables par « petits » schémas linéaires

- $|\alpha_i|, |\beta_i| \leq$ exponentielle
- $k \in$ polynomiale
- exposants-$* \leq$ exponentielle

Sommaire

Contributions principales

- 2-VASS aplatissables par petits schémas linéaires
- Complexité précise accessibilité 2-VASS : PSPACE-complet
- Complexité problèmes connexes

Sommaire

Contributions principales

- 2-VASS aplatissables par petits schémas linéaires
- Complexité précise accessibilité 2-VASS : PSPACE-complet
- Complexité problèmes connexes

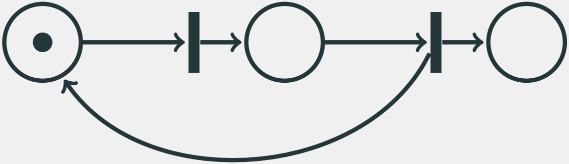
Publications

B., Finkel, Göller, Haase & McKenzie

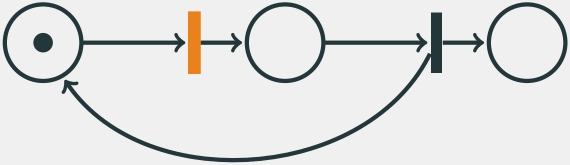
Reachability in Two-Dimensional Vector Addition Systems with States is PSPACE-complete

[3] ACM/IEEE Symposium on Logic in Computer Science, 2015

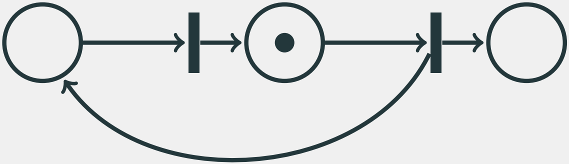
Systèmes à branchement infini



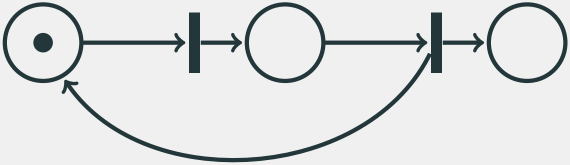
Systèmes à branchement infini



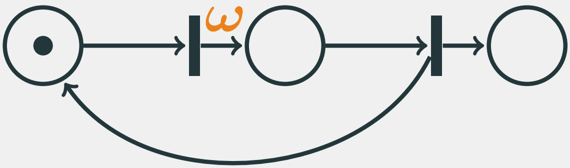
Systèmes à branchement infini



Systèmes à branchement infini

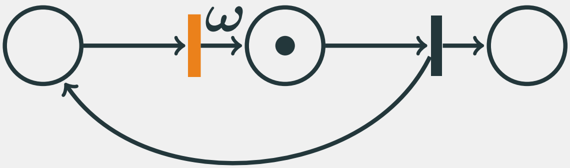


$$\mathrm{Succ}(\odot\bigcirc\bigcirc) = \bigcirc\odot\bigcirc$$

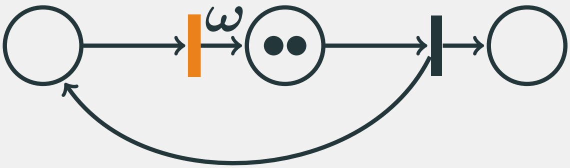


réseaux de Petri-ω (Geeraerts *et al.* '13)

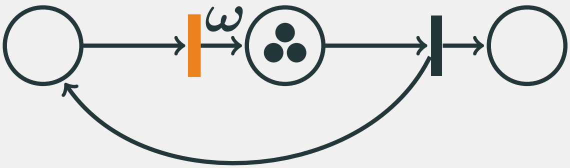
Systèmes à branchement infini



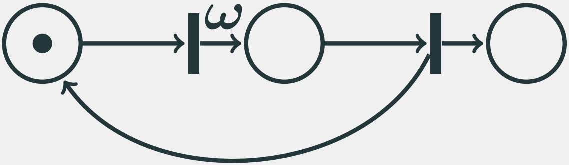
Systèmes à branchement infini



Systèmes à branchement infini



Systèmes à branchement infini



Modèles à branchement infini

- Réseaux de Petri-ω (Geeraerts *et al.* '13)
- Automates à insertion (Cécé, Finkel & Iyer '96)
- Automates à insertion avec files (Bouyer *et al.* '12)
- Systèmes parallèles récursifs (Kouchnarenko & Schnoebelen '96)
- Systèmes paramétrés, etc.

Systèmes à branchement infini

Modèles à branchement infini

- Réseaux de Petri-ω (Geeraerts *et al.* '13)
- Automates à insertion (Cécé, Finkel & Iyer '96)
- Automates à insertion avec files (Bouyer *et al.* '12)
- Systèmes parallèles récursifs (Kouchnarenko & Schnoebelen '96)
- Systèmes paramétrés, etc.

★ Unifier leur étude

Systèmes de transitions bien structurés (WSTS)

VASS

réseaux de Petri

réseaux de Petri-ω (Geeraerts *et al.* '13)

Systèmes de transitions bien structurés

VASS

réseaux de Petri

réseaux de Petri-ω (Geeraerts *et al.* '13)

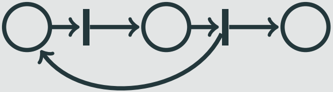
systèmes à canaux non fiables (Abdulla *et al.* '96)

plusieurs autres (Finkel & Schnoebelen '01)

WSTS **Finkel '87**

$\mathcal{S} = (X, \rightarrow, \leq)$ où

- ensemble X
- $\rightarrow \subseteq X \times X$
- monotonicité
- $\leq$ beau préordre

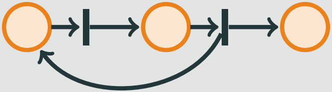


Systèmes de transitions bien structurés (WSTS)

WSTS **Finkel '87**

$\mathcal{S} = (X, \rightarrow, \leq)$ où

- $\mathbb{N}^3$
- $\rightarrow \subseteq X \times X$
- monotonicité
- $\leq$ beau préordre

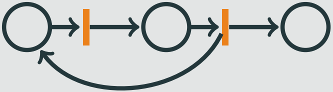


Systèmes de transitions bien structurés (WSTS)

WSTS **Finkel '87**

$\mathcal{S} = (X, \rightarrow, \leq)$ où

- ensemble X
- $\rightarrow \subseteq \mathbb{N}^3 \times \mathbb{N}^3$
- monotonicité
- $\leq$ beau préordre

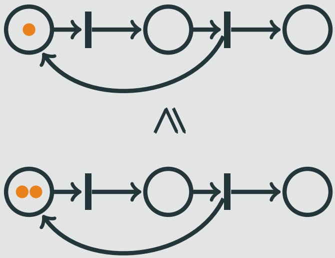


Systèmes de transitions bien structurés (WSTS)

WSTS **Finkel '87**

$\mathcal{S} = (X, \rightarrow, \leq)$ où

- ensemble X
- $\rightarrow \subseteq X \times X$
- monotonicité
- $\leq$ beau préordre

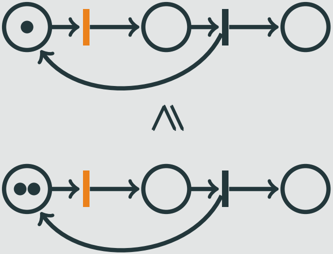


Systèmes de transitions bien structurés (WSTS)

WSTS **Finkel '87**

$\mathcal{S} = (X, \rightarrow, \leq)$ où

- ensemble X
- $\rightarrow \subseteq X \times X$
- monotonicité
- $\leq$ beau préordre

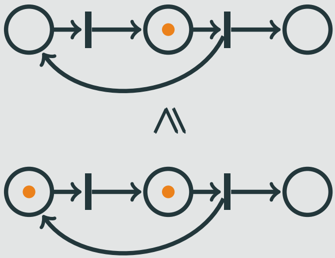


Systèmes de transitions bien structurés (WSTS)

WSTS — **Finkel '87**

$\mathcal{S} = (X, \to, \leq)$ où

- ensemble X
- $\to \subseteq X \times X$
- monotonicité
- $\leq$ beau préordre



Systèmes de transitions bien structurés (WSTS)

WSTS **Finkel '87**

$\mathcal{S} = (X, \rightarrow, \leq)$ où

- ensemble X
- $\rightarrow \subseteq X \times X$
- monotonicité
- $\leq$ beau préordre

$$\begin{array}{ccc} x & \longrightarrow & y \\ \rotatebox{90}{$\leq$} & & \\ x' & & \end{array}$$

WSTS **Finkel '87**

$\mathcal{S} = (X, \to, \leq)$ où

- ensemble X
- $\to \subseteq X \times X$
- monotonicité
- $\leq$ beau préordre

$$\begin{array}{ccc} x & \longrightarrow & y \\ \rotatebox[origin=c]{-90}{$\leq$} & & \rotatebox[origin=c]{-90}{$\leq$} \\ x' & \longrightarrow^{*} & y' \end{array}$$

Systèmes de transitions bien structurés (WSTS)

WSTS **Finkel '87**

$\mathcal{S} = (X, \rightarrow, \leq)$ où

- ensemble X
- $\rightarrow \subseteq X \times X$
- monotonicité
- $\leq$ beau préordre

réflexive, transitive et

$$\forall\, x_0, x_1, x_2, x_3, x_4, \ldots$$

Systèmes de transitions bien structurés (WSTS)

WSTS **Finkel '87**

$\mathcal{S} = (X, \rightarrow, \leq)$ où

- ensemble X
- $\rightarrow \subseteq X \times X$
- monotonicité
- $\leq$ beau préordre

réflexive, transitive et

$$\exists\, x_0, \underbrace{x_1, x_2, x_3, x_4}_{\leq}, \ldots$$

Branchement

Un WSTS est à *branchement fini* si pour tout x

$$\mathrm{Succ}(x) = \{y : x \to y\} \text{ est fini}$$

Branchement

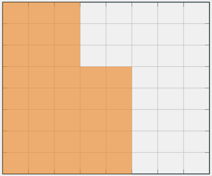
Un WSTS est à *branchement fini* si pour tout x

$$\mathrm{Succ}(x) = \{y : x \to y\} \text{ est fini}$$

Sinon, branchement infini

Du branchement infini à fini : complétion

Travailler sur ensemble clos par le bas



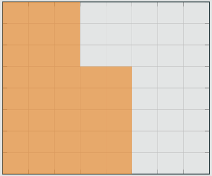
$\Downarrow$

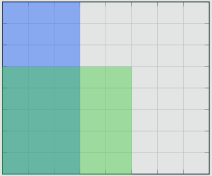
récupérer branchement fini

Du branchement infini à fini : complétion

Théorème **Finkel & Goubault-Larrecq '09, Fraïssé '86, Bonnet '75, ...**

$$\text{Ensemble clos par le bas } D \quad \Longrightarrow \quad D = \bigcup_{\text{finie}} \text{idéaux}$$

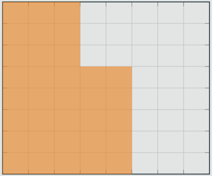


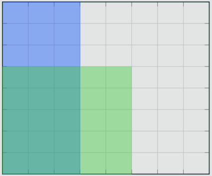


Du branchement infini à fini : complétion

Théorème — **Finkel & Goubault-Larrecq '09, Fraïssé '86, Bonnet '75, ...**

Ensemble clos par le bas $D \implies D = \bigcup_{\text{finie}}$ idéaux



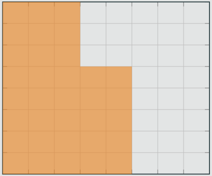


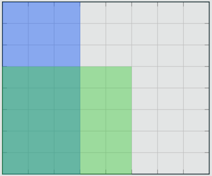
$D = \downarrow D$ où $\downarrow D = \{x : x \leq y \text{ pour } y \in D\}$

Du branchement infini à fini : complétion

Théorème — **Finkel & Goubault-Larrecq '09, Fraïssé '86, Bonnet '75, ...**

Ensemble clos par le bas $D \implies D = \bigcup_{\text{finie}}$ idéaux





$a, b \in I \implies \exists c \in I$ tel que $c \geq a$ et $c \geq b$

Complétion

B., Finkel & McKenzie '14

La *complétion* de $\mathcal{S} = (X, \leq, \rightarrow)$ est $\widehat{\mathcal{S}} = (\text{Idéaux}(X),\ \subseteq, \rightsquigarrow)$ où

$$I \rightsquigarrow J \text{ si } {\downarrow}\mathrm{Succ}(I) = \underbrace{\ldots \cup\ J\ \cup \ldots}_{\text{décomposition (unique)}}$$

Du branchement infini à fini : complétion

Complétion — **B., Finkel & McKenzie '14**

La *complétion* de $\mathcal{S} = (X, \leq, \rightarrow)$ est $\widehat{\mathcal{S}} = (\text{Idéaux}(X),\ \subseteq, \rightsquigarrow)$ où

$$I \rightsquigarrow J \text{ si } \downarrow\text{Succ}(I) = \underbrace{\ldots \cup\ J\ \cup \ldots}_{\text{décomposition (unique)}}$$

branchement fini et monotonicité

Du branchement infini à fini : complétion

Complétion — B., Finkel & McKenzie '14

La *complétion* de $\mathcal{S} = (X, \leq, \rightarrow)$ est $\widehat{\mathcal{S}} = (\text{Idéaux}(X), \subseteq, \rightsquigarrow)$ où

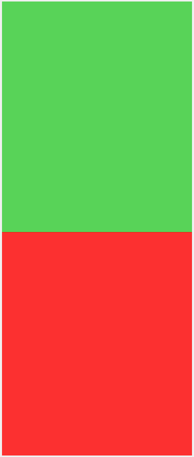
$$I \rightsquigarrow J \text{ si } \downarrow\text{Succ}(I) = \underbrace{\ldots \cup J \cup \ldots}_{\text{décomposition (unique)}}$$

$\subseteq$ pas toujours beau préordre... (Jančar '99)

...mais presque ! (Finkel & Goubault-Larrecq '09)

WSTS à branchement infini : décidabilité

Décidable



Indécidable

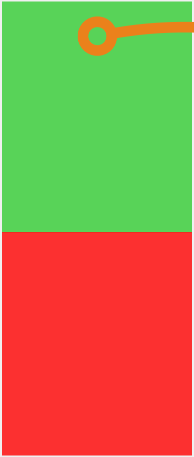
Couverture

Entrée : WSTS $\mathcal{S} = (X, \rightarrow, \leq)$ et $x, y \in X$

Question : Existe-t-il $y' \geq y$ tel que $x \rightarrow^* y'$?

WSTS à branchement infini : décidabilité

Décidable

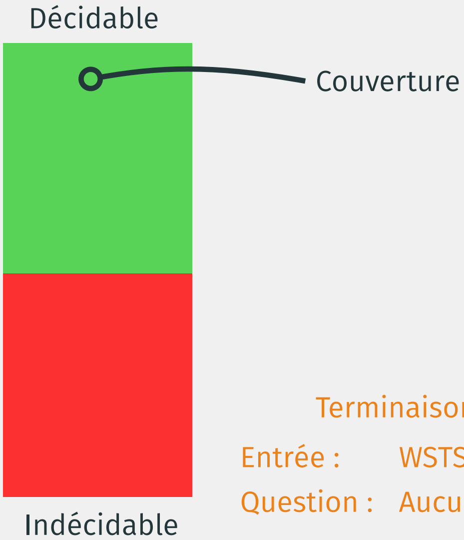


Indécidable

Couverture

- si $\rightsquigarrow$ décidable

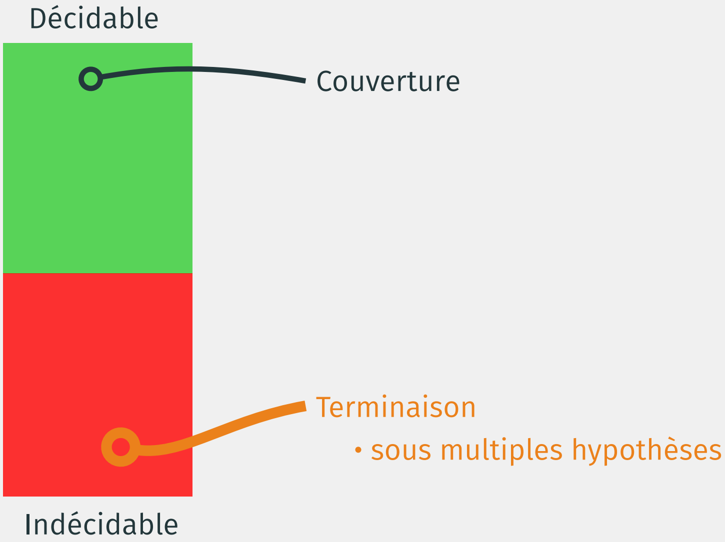
WSTS à branchement infini : décidabilité

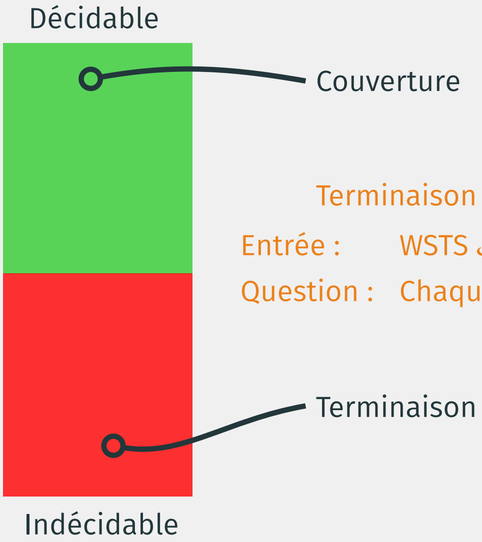


Terminaison

Entrée :	WSTS $\mathcal{S} = (X, \rightarrow, \leq)$ et $x \in X$
Question :	Aucune $\underbrace{x \rightarrow y_1 \rightarrow y_2 \rightarrow \cdots}_{\text{infinie}}$?

WSTS à branchement infini : décidabilité





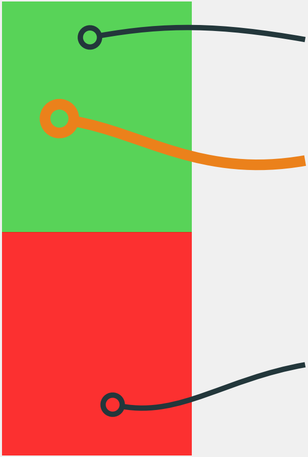
Terminaison forte

Entrée : WSTS $\mathcal{S} = (X, \to, \leq)$ et $x \in X$

Question : Chaque $x \to y_1 \to y_2 \to \ldots$ bornée ?

WSTS à branchement infini : décidabilité

Décidable



Couverture

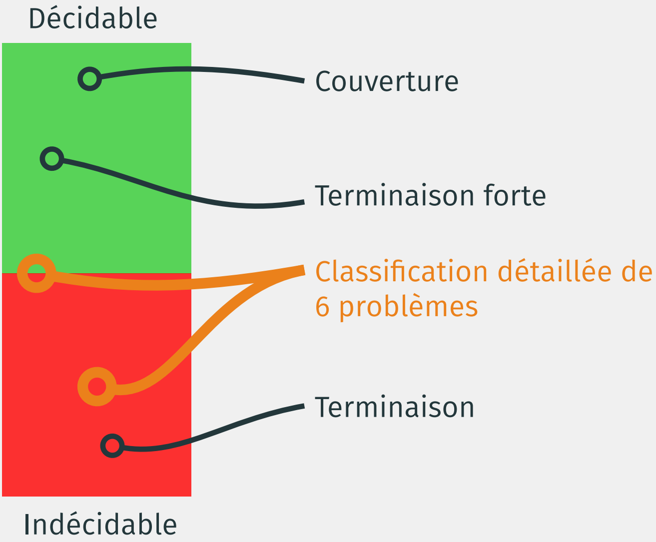
Terminaison forte

- si $\rightsquigarrow$ décidable
- $\subseteq$ beau préordre
- hypothèse monotonicité

Terminaison

Indécidable

WSTS à branchement infini : décidabilité



Sommaire

Contributions principales

- Complétion adaptée au branchement infini
- Étude approfondie des propriétés de la complétion
- Classification détaillée des problèmes de vérification pour WSTS à branchement infini

Sommaire

Contributions principales

- Complétion adaptée au branchement infini
- Étude approfondie des propriétés de la complétion
- Classification détaillée des problèmes de vérification pour WSTS à branchement infini

Publications

B., Finkel & McKenzie

Handling Infinitely Branching WSTS

[4] Int. Colloquium on Automata, Languages, and Programming, 2014

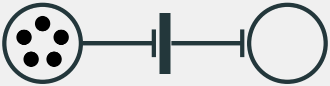
[5] Information & Computation, accepté

Pistes de recherche

- Prendre en charge extensions de réseaux de Petri : arcs de transfert/remise à zéro
- Combiner approche à algorithme de Karp & Miller
- Structures de données pour ensembles clos par le haut
  (p. ex. « sharing trees » Delzanno *et al.* '04)
- Geffroy, Leroux & Sutre RP'16 ?

Pistes de recherche

- Prendre en charge extensions de réseaux de Petri : arcs de transfert/remise à zéro
- Combiner approche à algorithme de Karp & Miller
- Structures de données pour ensembles clos par le haut
  (p. ex. « sharing trees » Delzanno *et al.* '04)
- Geffroy, Leroux & Sutre RP'16 ?



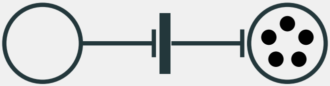
Pistes de recherche

- Prendre en charge extensions de réseaux de Petri : arcs de transfert/remise à zéro
- Combiner approche à algorithme de Karp & Miller
- Structures de données pour ensembles clos par le haut (p. ex. « sharing trees » Delzanno *et al.* '04)
- Geffroy, Leroux & Sutre RP'16 ?

Pistes de recherche

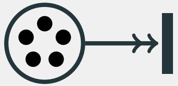
- Prendre en charge extensions de réseaux de Petri : arcs de transfert/remise à zéro
- Combiner approche à algorithme de Karp & Miller
- Structures de données pour ensembles clos par le haut

  (p. ex. « sharing trees » Delzanno *et al.* '04)
- Geffroy, Leroux & Sutre RP'16 ?



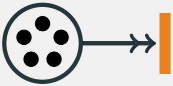
Pistes de recherche

- Prendre en charge extensions de réseaux de Petri : arcs de transfert/remise à zéro
- Combiner approche à algorithme de Karp & Miller
- Structures de données pour ensembles clos par le haut (p. ex. « sharing trees » Delzanno *et al.* '04)
- Geffroy, Leroux & Sutre RP'16 ?



Pistes de recherche

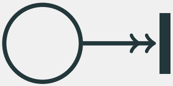
- Prendre en charge extensions de réseaux de Petri : arcs de transfert/remise à zéro
- Combiner approche à algorithme de Karp & Miller
- Structures de données pour ensembles clos par le haut
  (p. ex. « sharing trees » Delzanno *et al.* '04)
- Geffroy, Leroux & Sutre RP'16 ?



Pistes de recherche

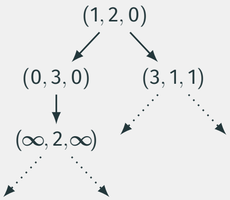
- Prendre en charge extensions de réseaux de Petri : arcs de transfert/remise à zéro
- Combiner approche à algorithme de Karp & Miller
- Structures de données pour ensembles clos par le haut

  (p. ex. « sharing trees » Delzanno *et al.* '04)
- Geffroy, Leroux & Sutre RP'16 ?



Pistes de recherche

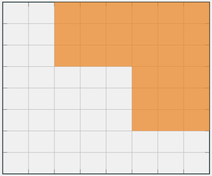
- Prendre en charge extensions de réseaux de Petri : arcs de transfert/remise à zéro
- Combiner approche à algorithme de Karp & Miller
- Structures de données pour ensembles clos par le haut
  (p. ex. « sharing trees » Delzanno *et al.* '04)
- Geffroy, Leroux & Sutre RP'16 ?



Pistes de recherche

- Prendre en charge extensions de réseaux de Petri : arcs de transfert/remise à zéro
- Combiner approche à algorithme de Karp & Miller
- Structures de données pour ensembles clos par le haut

  (p. ex. « sharing trees » Delzanno *et al.* '04)
- Geffroy, Leroux & Sutre RP'16 ?

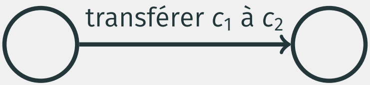


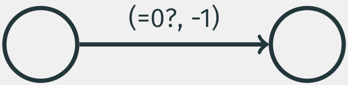
Pistes de recherche

- Prendre en charge extensions de réseaux de Petri : arcs de transfert/remise à zéro
- Combiner approche à algorithme de Karp & Miller
- Structures de données pour ensembles clos par le haut
  (p. ex. « sharing trees » Delzanno *et al.* '04)
- Geffroy, Leroux & Sutre RP'16 ?

Pistes de recherche

- Complexité extensions 2-VASS : avec transferts/remises à zéro, un test à zéro (décidable, Finkel & Sutre '00)
- Complexité VASS à max. deux compteurs simultanément non bornés
- Décidabilité 2-VASS paramétrés (1-VASS $\in$ NEXPTIME, Lechner '15)





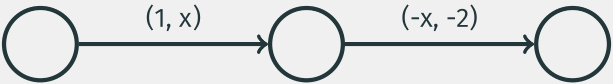
Pistes de recherche

- Complexité extensions 2-VASS : avec transferts/remises à zéro, un test à zéro (décidable, Finkel & Sutre '00)
- Complexité VASS à max. deux compteurs simultanément non bornés
- Décidabilité 2-VASS paramétrés (1-VASS $\in$ NEXPTIME, Lechner '15)

$$(0, 0, 0, 0) \to (3, 2, 1, 0) \to (5, 6, 1, 1) \to (12, 7, 0, 1) \to \cdots$$

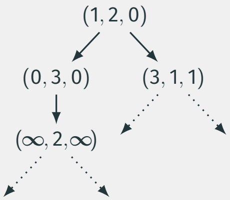
Pistes de recherche

- Complexité extensions 2-VASS : avec transferts/remises à zéro, un test à zéro (décidable, Finkel & Sutre '00)
- Complexité VASS à max. deux compteurs simultanément non bornés
- Décidabilité 2-VASS paramétrés (1-VASS $\in$ NEXPTIME, Lechner '15)



Pistes de recherche

- Généraliser l'algorithme de couverture de Karp & Miller à une classe générique de WSTS à branchement infini
- Étudier implémentabilité du nouvel algorithme de couverture pour modèles concrets
- Étudier WSTS sans beaux préordres, mais avec préordres sans antichaînes infinies



Pistes de recherche

- Généraliser l'algorithme de couverture de Karp & Miller à une classe générique de WSTS à branchement infini
- Étudier implémentabilité du nouvel algorithme de couverture pour modèles concrets
- Étudier WSTS sans beaux préordres, mais avec préordres sans antichaînes infinies

Pistes de recherche

- Généraliser l'algorithme de couverture de Karp & Miller à une classe générique de WSTS à branchement infini
- Étudier implémentabilité du nouvel algorithme de couverture pour modèles concrets
- Étudier WSTS sans beaux préordres, mais avec préordres sans antichaînes infinies

Merci !