

Accessibilité dans les systèmes à compteurs continus: théorie et pratique

Vincent Antaki

Université de Montréal

antakivi@iro.umontreal.ca

29 juin 2015

Contexte : La vérification formelle

La vérification formelle vise à confirmer qu'un système a le comportement attendu.

Introduction

Les VASS
continus

Implémentation

Conclusion

Contexte : La vérification formelle

La vérification formelle vise à confirmer qu'un système a le comportement attendu.

- Circuits combinatoires et circuits numériques

Contexte : La vérification formelle

La vérification formelle vise à confirmer qu'un système a le comportement attendu.

- Circuits combinatoires et circuits numériques
- Moteur d'inférences logiques

Contexte : La vérification formelle

La vérification formelle vise à confirmer qu'un système a le comportement attendu.

- Circuits combinatoires et circuits numériques
- Moteur d'inférences logiques
- Logiciels

Contexte : La vérification formelle

La vérification formelle vise à confirmer qu'un système a le comportement attendu.

- Circuits combinatoires et circuits numériques
- Moteur d'inférences logiques
- Logiciels

Formalise des modèles et des questions :

Contexte : La vérification formelle

La vérification formelle vise à confirmer qu'un système a le comportement attendu.

- Circuits combinatoires et circuits numériques
- Moteur d'inférences logiques
- Logiciels

Formalise des modèles et des questions :

- Existe-t-il une séquence infinie dans cet automate ?

Contexte : La vérification formelle

La vérification formelle vise à confirmer qu'un système a le comportement attendu.

- Circuits combinatoires et circuits numériques
- Moteur d'inférences logiques
- Logiciels

Formalise des modèles et des questions :

- Existe-t-il une séquence infinie dans cet automate ?
- Le graphe suivant possède-t-il un chemin d'un état x à un état y ?

Contexte : La vérification formelle

La vérification formelle vise à confirmer qu'un système a le comportement attendu.

- Circuits combinatoires et circuits numériques
- Moteur d'inférences logiques
- Logiciels

Formalise des modèles et des questions :

- Existe-t-il une séquence infinie dans cet automate ?
- Le graphe suivant possède-t-il un chemin d'un état x à un état y ?
- L'automate accepte-t-il un ensemble de mots X ?

Contexte : La vérification formelle

La vérification formelle vise à confirmer qu'un système a le comportement attendu.

- Circuits combinatoires et circuits numériques
- Moteur d'inférences logiques
- Logiciels

Formalise des modèles et des questions :

- Existe-t-il une séquence infinie dans cet automate ?
- Le graphe suivant possède-t-il un chemin d'un état x à un état y ?
- L'automate accepte-t-il un ensemble de mots X ?
- Une expression est-elle satisfiable dans un système de logique donné ?

Les réseaux de Petri (PN)

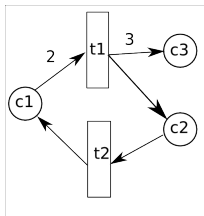
Un réseau de Petri $P = (P, T, Pre, Post)$ où

Introduction

Les VASS
continus

Implémentation

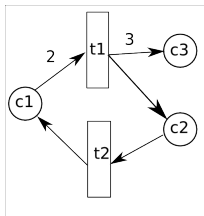
Conclusion



Les réseaux de Petri (PN)

Un réseau de Petri $P = (P, T, Pre, Post)$ où

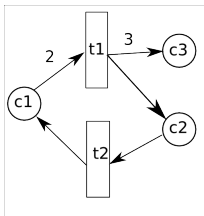
- P un ensemble fini de *places* (compteurs);



Les réseaux de Petri (PN)

Un réseau de Petri $P = (P, T, Pre, Post)$ où

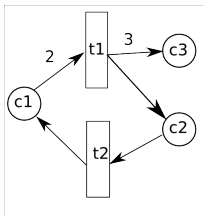
- P un ensemble fini de *places* (compteurs);
- T un ensemble fini de *transitions*;



Les réseaux de Petri (PN)

Un réseau de Petri $P = (P, T, Pre, Post)$ où

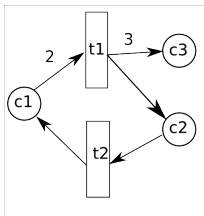
- P un ensemble fini de *places* (compteurs);
- T un ensemble fini de *transitions*;
- Pre une matrice $|P| \times |T|$ indiquant le coût pour activer chaque transition;



Les réseaux de Petri (PN)

Un réseau de Petri $P = (P, T, Pre, Post)$ où

- P un ensemble fini de *places* (compteurs) ;
- T un ensemble fini de *transitions* ;
- Pre une matrice $|P| \times |T|$ indiquant le coût pour activer chaque transition ;
- $Post$ une matrice $|P| \times |T|$ indiquant l'ajout dans les compteurs suite à l'activation de chaque transition

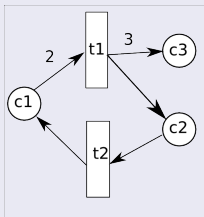


Exécution dans un PN

Exemple

Séquence d'activation : $t_1 t_2$

Exécution : (2, 0, 0)

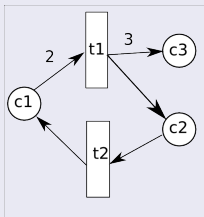


Exécution dans un PN

Exemple

Séquence d'activation : $t_1 t_2$

Exécution : $(2, 0, 0)(0, 1, 3)$

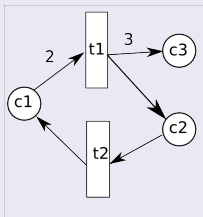


Exécution dans un PN

Exemple

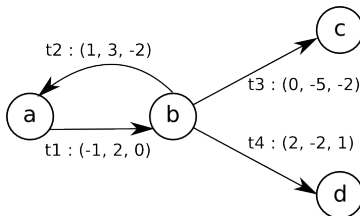
Séquence d'activation : $t_1 t_2$

Exécution : $(2, 0, 0)(0, 1, 3)(1, 0, 3)$



Les systèmes d'additions de vecteurs avec états (VASS)

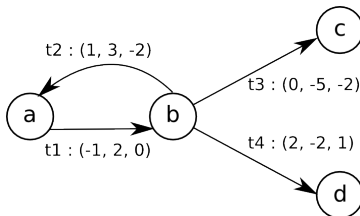
Un d -VASS est une paire (Q, δ) où :



Les systèmes d'additions de vecteurs avec états (VASS)

Un d -VASS est une paire (Q, δ) où :

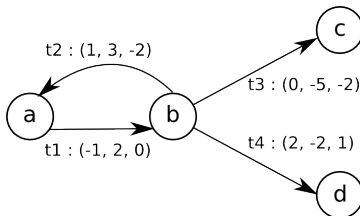
- d la *dimension* (nombre de compteurs) ;



Les systèmes d'additions de vecteurs avec états (VASS)

Un d -VASS est une paire (Q, δ) où :

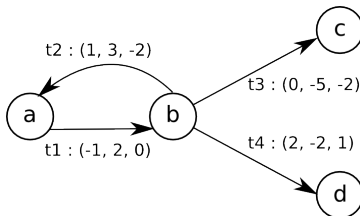
- d la *dimension* (nombre de compteurs) ;
- Q un ensemble fini d'*états* ;



Les systèmes d'additions de vecteurs avec états (VASS)

Un d -VASS est une paire (Q, δ) où :

- d la *dimension* (nombre de compteurs) ;
- Q un ensemble fini d'*états* ;
- $\delta \subseteq Q \times \mathbb{Z}^d \times Q$ un ensemble fini de *transitions*.

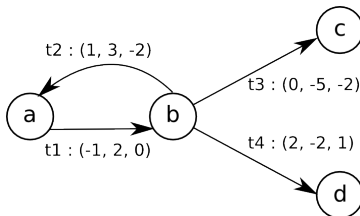


Les systèmes d'additions de vecteurs avec états (VASS)

Un d -VASS est une paire (Q, δ) où :

- d la *dimension* (nombre de compteurs) ;
- Q un ensemble fini d'*états* ;
- $\delta \subseteq Q \times \mathbb{Z}^d \times Q$ un ensemble fini de *transitions*.

Les VASS sont équivalents aux PN.

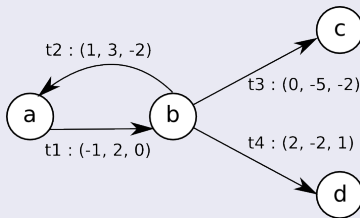


Exécution dans un VASS

Exemple

Séquence d'activation : $t_2 t_1 t_3$

Exécution : $(b, (0, 0, 4))$

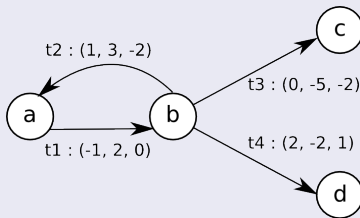


Exécution dans un VASS

Exemple

Séquence d'activation : $t_2 t_1 t_3$

Exécution : $(b, (0, 0, 4))(a, (1, 3, 2))$

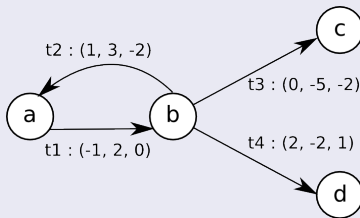


Exécution dans un VASS

Exemple

Séquence d'activation : $t_2 t_1 t_3$

Exécution : $(b, (0, 0, 4))(a, (1, 3, 2))(b, (0, 5, 2))$

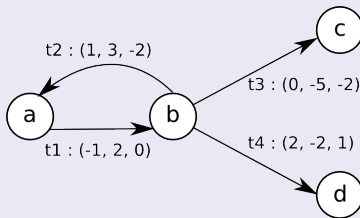


Exécution dans un VASS

Exemple

Séquence d'activation : $t_2 t_1 t_3$

Exécution : $(b, (0, 0, 4))(a, (1, 3, 2))(b, (0, 5, 2))(c, (0, 0, 0))$



Le problème d'accessibilité

Étant donné un système S , une configuration initiale x_0 et une configuration finale x , existe-t-il une exécution $x_0 \xrightarrow{*} x$ dans S ?

Le problème d'accessibilité

Étant donné un système S , une configuration initiale x_0 et une configuration finale x , existe-t-il une exécution $x_0 \xrightarrow{*} x$ dans S ?

| Modèle | Complexité | | |
|--------|---------------------------|-------------------------|---------------------------|
| | $d = 1$ | $d = 2$ | $d \geq 3$ |
| VASS | NP-complet [HKOW09] | PSPACE-complet [BFG+14] | $\in F_{\omega^3}$ [LS15] |
| PN | $\in F_{\omega^3}$ [LS15] | | |
| | | | |

Le problème d'accessibilité

Étant donné un système S , une configuration initiale x_0 et une configuration finale x , existe-t-il une exécution $x_0 \xrightarrow{*} x$ dans S ?

| Modèle | Complexité | | |
|--------------------|---------------------------|-------------------------|---------------------------|
| | $d = 1$ | $d = 2$ | $d \geq 3$ |
| VASS | NP-complet [HKOW09] | PSPACE-complet [BFG+14] | $\in F_{\omega^3}$ [LS15] |
| PN | $\in F_{\omega^3}$ [LS15] | | |
| CPN | P-complet [FH13] | | |
| \mathbb{Z} -VASS | NP-complet [HH14] | | |

Les réseaux de Petri continus (CPN)

Introduction

Les VASS
continus

Implémentation

Conclusion

Les réseaux de Petri continus (CPN)

Introduction

Les VASS continus

Implémentation

Conclusion

- Mêmes données qu'un PN ($P, T, Pre, Post$)

Les réseaux de Petri continus (CPN)

Introduction

Les VASS continus

Implémentation

Conclusion

- Mêmes données qu'un PN $(P, T, Pre, Post)$
- Marquage dans les réels $m \in \mathbb{R}_{\geq 0}^P$

Les réseaux de Petri continus (CPN)

Introduction

Les VASS continus

Implémentation

Conclusion

- Mêmes données qu'un PN $(P, T, Pre, Post)$
- Marquage dans les réels $m \in \mathbb{R}_{\geq 0}^P$
- Chaque transition est appliquée avec un coefficient $\alpha \in (0, 1]$ qui pondère l'effet de la transition sur les compteurs

Les réseaux de Petri continus (CPN)

Introduction

Les VASS continus

Implémentation

Conclusion

- Mêmes données qu'un PN $(P, T, Pre, Post)$
- Marquage dans les réels $m \in \mathbb{R}_{\geq 0}^P$
- Chaque transition est appliquée avec un coefficient $\alpha \in (0, 1]$ qui pondère l'effet de la transition sur les compteurs
- Peut converger avec des séquences d'activation infinies

Les réseaux de Petri continus (CPN)

Introduction

Les VASS continus

Implémentation

Conclusion

- Mêmes données qu'un PN $(P, T, Pre, Post)$
- Marquage dans les réels $m \in \mathbb{R}_{\geq 0}^P$
- Chaque transition est appliquée avec un coefficient $\alpha \in (0, 1]$ qui pondère l'effet de la transition sur les compteurs
- Peut converger avec des séquences d'activation infinies
- Problème d'accessibilité P-Complet : Fraca & Haddad 2013

Le Projet

Objectif :

Le Projet

Objectif :

- Développement de module pour un outil permettant de résoudre partiellement le problème d'accessibilité dans les VASS

Le Projet

Objectif :

- Développement de module pour un outil permettant de résoudre partiellement le problème d'accessibilité dans les VASS

Résultats :

Le Projet

Objectif :

- Développement de module pour un outil permettant de résoudre partiellement le problème d'accessibilité dans les VASS

Résultats :

- Deux modèles définis (CVASSM et CVASSU)

Le Projet

Objectif :

- Développement de module pour un outil permettant de résoudre partiellement le problème d'accessibilité dans les VASS

Résultats :

- Deux modèles définis (CVASSM et CVASSU)
- CVASSM prouvés équivalents aux CPN

Le Projet

Objectif :

- Développement de module pour un outil permettant de résoudre partiellement le problème d'accessibilité dans les VASS

Résultats :

- Deux modèles définis (CVASSM et CVASSU)
- CVASSM prouvés équivalents aux CPN
- CVASSU constituant une nouvelle relaxation des VASS

Le Projet

Objectif :

- Développement de module pour un outil permettant de résoudre partiellement le problème d'accessibilité dans les VASS

Résultats :

- Deux modèles définis (CVASSM et CVASSU)
- CVASSM prouvés équivalents aux CPN
- CVASSU constituant une nouvelle relaxation des VASS
- Implémentation d'un algorithme résolvant le problème d'accessibilité pour les CPN

Les systèmes d'addition de vecteurs continus avec états

- Même données qu'un VASS (Q, δ, d)

Les systèmes d'addition de vecteurs continus avec états

- Même données qu'un VASS (Q, δ, d)
- Les compteurs ont un nombre de jetons $x \in \mathbb{R}_{\geq 0}$

Les systèmes d'addition de vecteurs continus avec états

- Même données qu'un VASS (Q, δ, d)
- Les compteurs ont un nombre de jetons $x \in \mathbb{R}_{\geq 0}$
- Chaque transition avec un coefficient $\alpha \in (0, 1]$ qui pondère l'effet de la transition sur les compteurs

Les systèmes d'addition de vecteurs continus avec états

- Même données qu'un VASS (Q, δ, d)
- Les compteurs ont un nombre de jetons $x \in \mathbb{R}_{\geq 0}$
- Chaque transition avec un coefficient $\alpha \in (0, 1]$ qui pondère l'effet de la transition sur les compteurs

Avec états uniques (CVASSU) :

- Change entièrement d'état lorsqu'on emprunte une transition, indépendamment du coefficient α

Les systèmes d'addition de vecteurs continus avec états

- Même données qu'un VASS (Q, δ, d)
- Les compteurs ont un nombre de jetons $x \in \mathbb{R}_{\geq 0}$
- Chaque transition avec un coefficient $\alpha \in (0, 1]$ qui pondère l'effet de la transition sur les compteurs

Avec états uniques (CVASSU) :

- Change entièrement d'état lorsqu'on emprunte une transition, indépendamment du coefficient α

Avec états multiples (CVASSM) :

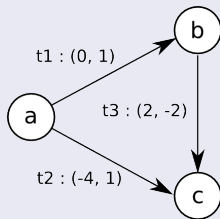
- α pondère le changement d'état permettant ainsi au modèle d'être partiellement dans plusieurs états.

CVASS avec état unique (CVASSU)

Exemple d'exécution

Séquence d'activation : $1t_10.5t_3$

Exécution : $(a, (2, 0))$

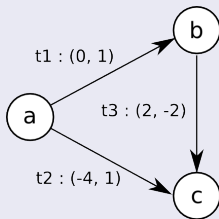


CVASS avec état unique (CVASSU)

Exemple d'exécution

Séquence d'activation : $1t_10.5t_3$

Exécution : $(a, (2, 0))$ $(b, (2, 1))$

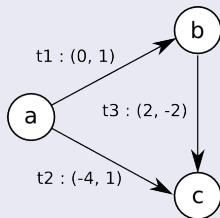


CVASS avec état unique (CVASSU)

Exemple d'exécution

Séquence d'activation : $1t_1$ $0.5t_3$

Exécution : $(a, (2, 0))$ $(b, (2, 1))$ $(c, (3, 0))$

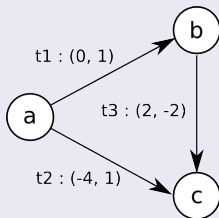


CVASS avec états multiples (CVASSM)

Exemple d'exécution

Séquence d'activation : $0.5t_1 0.5t_2 0.5t_3$

Exécution : $((1, 0, 0), (2, 0))$

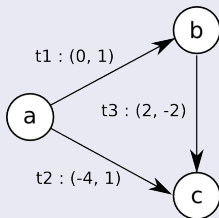


CVASS avec états multiples (CVASSM)

Exemple d'exécution

Séquence d'activation : $0.5t_1 0.5t_2 0.5t_3$

Exécution : $((1, 0, 0), (2, 0)) ((0.5, 0.5, 0), (2, 0.5))$

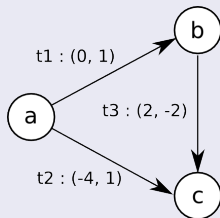


CVASS avec états multiples (CVASSM)

Exemple d'exécution

Séquence d'activation : $0.5t_1$ $0.5t_2$ $0.5t_3$

Exécution : $((1, 0, 0), (2, 0))$ $((0.5, 0.5, 0), (2, 0.5))$
 $((0, 0.5, 0.5), (0, 1))$

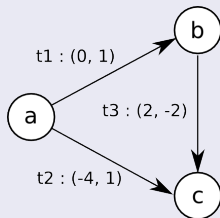


CVASS avec états multiples (CVASSM)

Exemple d'exécution

Séquence d'activation : $0.5t_1 0.5t_2 0.5t_3$

Exécution : $((1, 0, 0), (2, 0)) ((0.5, 0.5, 0), (2, 0.5))$
 $((0, 0.5, 0.5), (0, 1)) ((0, 0, 1), (1, 0))$



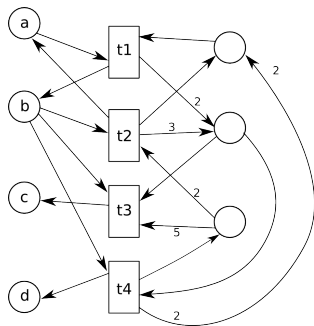
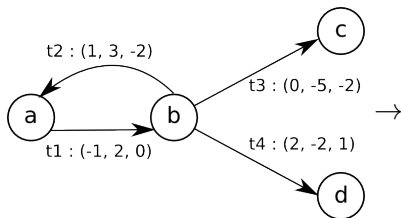
Conversion CVASSM à CPN

Introduction

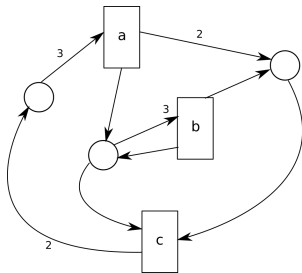
Les VASS
continus

Implémentation

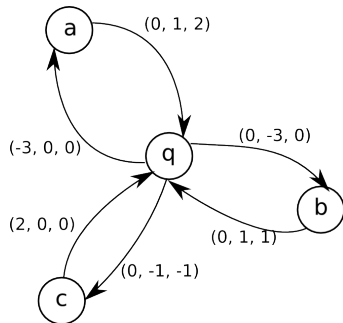
Conclusion



Conversion CPN à CVASSM



→



Ordonnancement d'une séquence d'activation

Exemple

- On suppose que l'on commence et termine dans l'état q

Ordonnancement d'une séquence d'activation

Exemple

- On suppose que l'on commence et termine dans l'état q
- Soit une séquence désordonnée :

$0.5\delta_{qc}$ $0.5\delta_{qb}$ $0.25\delta_{bq}$ $0.25\delta_{qc}$ $0.25\delta_{bq}$ $0.75\delta_{cq}$

Ordonnancement d'une séquence d'activation

Exemple

- On suppose que l'on commence et termine dans l'état q
- Soit une séquence désordonnée :

$0.5\delta_{qc}$ $0.5\delta_{qb}$ $0.25\delta_{bq}$ $0.25\delta_{qc}$ $0.25\delta_{bq}$ $0.75\delta_{cq}$

① $0.5\delta_{qc}$ $0.5\delta_{cq}$ $0.5\delta_{qb}$ $0.25\delta_{bq}$ $0.25\delta_{qc}$ $0.25\delta_{cq}$ $0.25\delta_{bq}$

Ordonnancement d'une séquence d'activation

Exemple

- On suppose que l'on commence et termine dans l'état q
- Soit une séquence désordonnée :

$0.5\delta_{qc}$ $0.5\delta_{qb}$ $0.25\delta_{bq}$ $0.25\delta_{qc}$ $0.25\delta_{bq}$ $0.75\delta_{cq}$

① $0.5\delta_{qc}$ $0.5\delta_{cq}$ $0.5\delta_{qb}$ $0.25\delta_{bq}$ $0.25\delta_{qc}$ $0.25\delta_{cq}$ $0.25\delta_{bq}$

② $0.5\delta_{qc}$ $0.5\delta_{cq}$ $0.5\delta_{qb}$ $0.5\delta_{bq}$ $0.25\delta_{qc}$ $0.25\delta_{cq}$

Algorithme de Fraca et Haddad

Introduction

Les VASS
continus

Implémentation

Conclusion

Algorithm 2: Decision algorithm for reachability

Reachable($\langle \mathcal{N}, \mathbf{m}_0 \rangle, \mathbf{m}$): status
Input: a CPN system $\langle \mathcal{N}, \mathbf{m}_0 \rangle$, a marking \mathbf{m}
Output: the reachability status of \mathbf{m}
Output: the Parikh image of a witness in the positive case
Data: *nbsol*: integer; \mathbf{v}, \mathbf{sol} : vectors; T' : subset of transitions

```

1 if  $\mathbf{m} = \mathbf{m}_0$  then return (true,0)
2  $T' \leftarrow T$ 
3 while  $T' \neq \emptyset$  do
4    $nbsol \leftarrow 0$ ;  $\mathbf{sol} \leftarrow \mathbf{0}$ 
5   for  $t \in T'$  do
6     solve  $\exists \mathbf{v} \ \mathbf{v} \geq \mathbf{0} \wedge \mathbf{v}[t] > 0 \wedge \mathbf{C}_{P \times T'} \mathbf{v} = \mathbf{m} - \mathbf{m}_0$ 
7     if  $\exists \mathbf{v}$  then  $nbsol \leftarrow nbsol + 1$ ;  $\mathbf{sol} \leftarrow \mathbf{sol} + \mathbf{v}$ 
8   end
9   if  $nbsol = 0$  then return false else  $\mathbf{sol} \leftarrow \frac{1}{nbsol} \mathbf{sol}$ 
10   $T' \leftarrow \llbracket \mathbf{sol} \rrbracket$ 
11   $T' \leftarrow T' \cap \max\text{FS}(\mathcal{N}_{T'}, \mathbf{m}_0[\bullet T' \bullet])$ 
12   $T' \leftarrow T' \cap \max\text{FS}(\mathcal{N}_{T'}^{-1}, \mathbf{m}[\bullet T' \bullet])$  /* deleted for lim-reachability */
13  if  $T' = \llbracket \mathbf{sol} \rrbracket$  then return (true,  $\mathbf{sol}$ )
14 end
15 return false
```

Algorithme de Fraca et Haddad

Introduction

Les VASS
continus

Implémentation

Conclusion

Algorithm 2: Decision algorithm for reachability

Reachable($\langle \mathcal{N}, \mathbf{m}_0 \rangle, \mathbf{m} \rangle$): status
Input: a CPN system $\langle \mathcal{N}, \mathbf{m}_0 \rangle$, a marking \mathbf{m}
Output: the reachability status of \mathbf{m}
Output: the Parikh image of a witness in the positive case
Data: $nbsol$: integer; \mathbf{v}, \mathbf{sol} : vectors; T' : subset of transitions

```

1 if  $\mathbf{m} = \mathbf{m}_0$  then return (true,0)
2  $T' \leftarrow T$ 
3 while  $T' \neq \emptyset$  do
4    $nbsol \leftarrow 0$ ;  $\mathbf{sol} \leftarrow \mathbf{0}$ 
5   for  $t \in T'$  do
6     solve  $\exists \mathbf{v} \ \mathbf{v} \geq \mathbf{0} \wedge \mathbf{v}[t] > 0 \wedge \mathbf{C}_{P \times T'} \mathbf{v} = \mathbf{m} - \mathbf{m}_0$ 
7     if  $\exists \mathbf{v}$  then  $nbsol \leftarrow nbsol + 1$ ;  $\mathbf{sol} \leftarrow \mathbf{sol} + \mathbf{v}$ 
8   end
9   if  $nbsol = 0$  then return false else  $\mathbf{sol} \leftarrow \frac{1}{nbsol} \mathbf{sol}$ 
10   $T' \leftarrow \llbracket \mathbf{sol} \rrbracket$ 
11   $T' \leftarrow T' \cap \max\text{FS}(\mathcal{N}_{T'}, \mathbf{m}_0[\bullet T' \bullet])$ 
12   $T' \leftarrow T' \cap \max\text{FS}(\mathcal{N}_{T'}^{-1}, \mathbf{m}[\bullet T' \bullet])$  /* deleted for lim-reachability */
13  if  $T' = \llbracket \mathbf{sol} \rrbracket$  then return (true,  $\mathbf{sol}$ )
14 end
15 return false
```

Système à résoudre

Résoudre $\exists v$ tel que $v \geq 0 \wedge v[t] > 0 \wedge C_{P \times T'} v = m - m_0$

Comment résoudre ce système avec un solveur qui ne traite pas les inégalités strictes ?

Système à résoudre

Résoudre $\exists v$ tel que $v \geq 0 \wedge v[t] > 0 \wedge C_{P \times T'} v = m - m_0$

Comment résoudre ce système avec un solveur qui ne traite pas les inégalités strictes ?

①

Maximiser t

Sujet à $C_{P \times T'} v = m - m_0 \wedge v \geq 0$

②

Minimiser t

Sujet à $C_{P \times T'} v = m - m_0 \wedge v \geq 0 \wedge v[t] \geq 1$

Solveur exact

- Les solveurs conventionnels utilisent l'arithmétique à virgule flottante, insuffisante dans le cadre de la vérification formelle.

Solveur exact

- Les solveurs conventionnels utilisent l'arithmétique à virgule flottante, insuffisante dans le cadre de la vérification formelle.
- Solveur exact en arithmétique rationnelle plus lent

Solveur exact

- Les solveurs conventionnels utilisent l'arithmétique à virgule flottante, insuffisante dans le cadre de la vérification formelle.
- Solveur exact en arithmétique rationnelle plus lent
- QSopt_ex : Simplexe exact en C++

Questions ouvertes et travaux futurs

Questions ouvertes et travaux futurs

- Solveurs alternatifs et tests de performances

Questions ouvertes et travaux futurs

Introduction

Les VASS
continus

Implémentation

Conclusion

- Solveurs alternatifs et tests de performances
- Accessibilité dans les CVASSU

Questions ouvertes et travaux futurs

Introduction

Les VASS
continus

Implémentation

Conclusion

- Solveurs alternatifs et tests de performances
- Accessibilité dans les CVASSU
- Prochains modules

Merci