PERSÉE : 2003–2006

Techniques symboliques pour la vérification automatique
des systèmes critiques hétérogènes

ACI Sécurité Informatique

Philippe Schnoebelen

http://www.lsv-ens-cachan.fr/aci-persee/

Plan de la présentation

- Participants
- Contexte scientifique : vérification des systèmes critiques
- – analyse symbolique d'accessibilité
- – abstractions et méthodes symboliques
- – environnement intégré

Les participants

LSV (Cachan)

Ph. Schnoebelen, A. Finkel, D. Nowak, S. Bardin, ...

LIAFA (Paris 7)

A. Bouajjani, M. Sighireanu, P. Habermehl, ...

LaBRI (Bordeaux)

G. Sutre, A. Griffault, K. Musumbu, F. Herbreteau, ...

Trois model-checkers symboliques : FAST, TReX, AltaRica

Pas de collaboration antérieure

Contexte : Vérification des systèmes critiques

- Systèmes complexes : temps-réel, concurrents, réactifs, ...

Contexte : Vérification des systèmes critiques

- Systèmes complexes : temps-réel, concurrents, réactifs, ...

- Domaine d'application : systèmes embarqués, protocoles, systèmes sur puce, systèmes de contrôle-commande, algorithmes distribués, ...

Contexte : Vérification des systèmes critiques

• Systèmes complexes : temps-réel, concurrents, réactifs, ...

• Domaine d'application : systèmes embarqués, protocoles, systèmes sur puce, systèmes de contrôle-commande, algorithmes distribués, ...

• Notre approche : vérification automatique (model-checking)
→ systèmes à infinité d'états
→ méthodes symboliques (regular model-checking)
→ techniques d'accélération des calculs itératifs de points-fixes

Contexte : Vérification des systèmes critiques

• Systèmes complexes : temps-réel, concurrents, réactifs, ...

• Domaine d'application : systèmes embarqués, protocoles, systèmes sur puce, systèmes de contrôle-commande, algorithmes distribués, ...

• Notre approche : vérification automatique (model-checking)
→ systèmes à infinité d'états
→ méthodes symboliques (regular model-checking)
→ techniques d'accélération des calculs itératifs de points-fixes

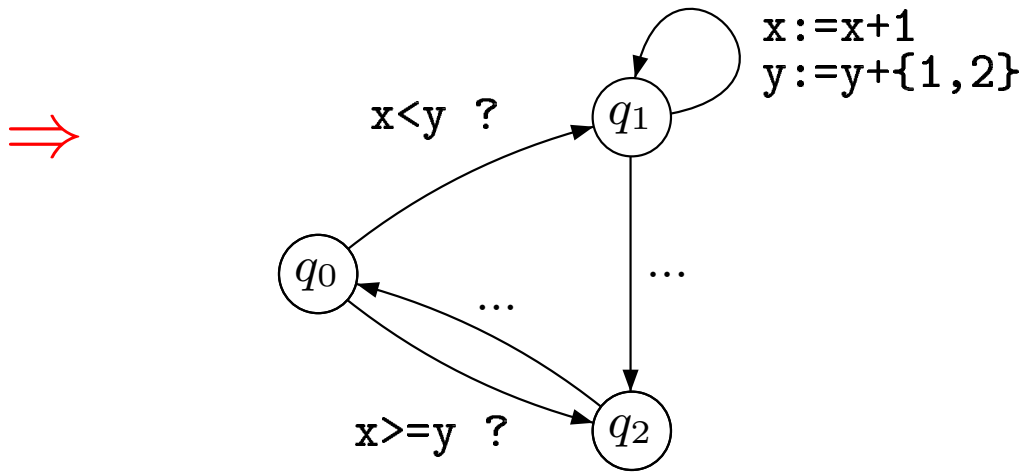
• Objectifs du projets : rapprocher les techniques mises en oeuvre dans les trois équipes partenaires. Concrétiser le rapprochement via l'échange de librairies, d'algorithmes, d'études de cas, ..., au sein d'un environnement de vérification unifié.

Ingrédients pour la vérification symbolique

```
channel c
   process P1(u)
    send u -> c
      ...
||  process P2(v)
    receive c <- n
    if #P1<#P2 ···
```

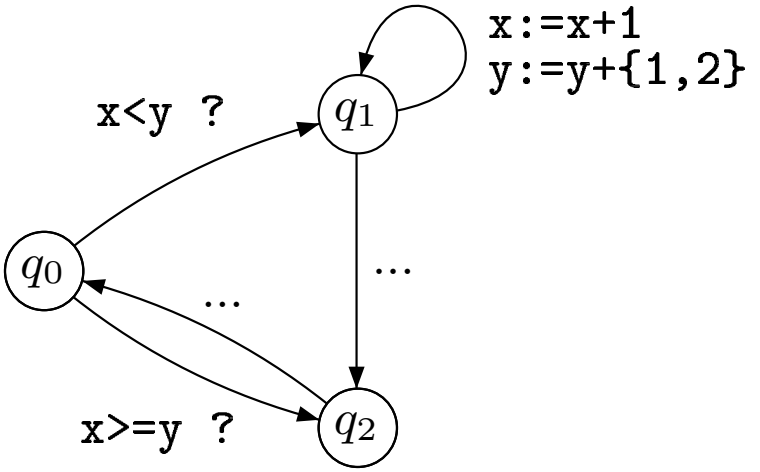
Ingrédients pour la vérification symbolique

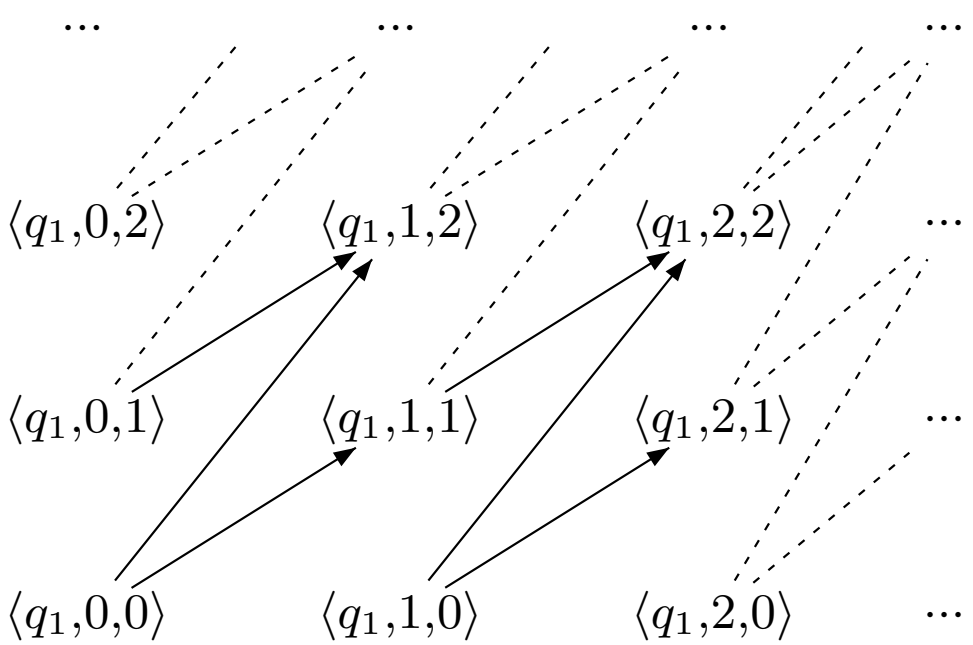
```
channel c
   process P1(u)
    send u -> c
      ...
||  process P2(v)
    receive c <- n
    if #P1<#P2 ···
```



Ingrédients pour la vérification symbolique

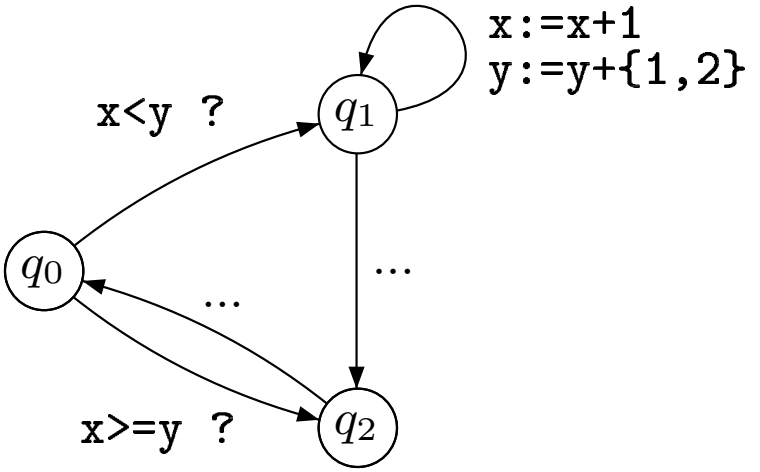
```
channel c
   process P1(u)
    send u -> c
      ...
||  process P2(v)
    receive c <- n
    if #P1<#P2 ···
```

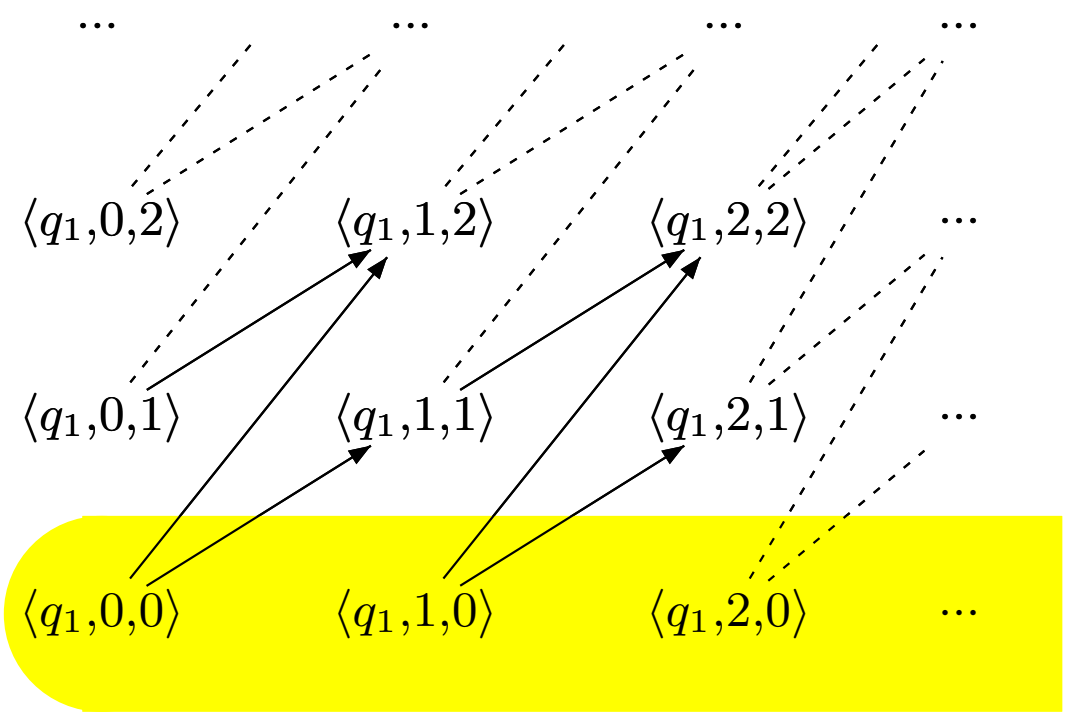




Ingrédients pour la vérification symbolique

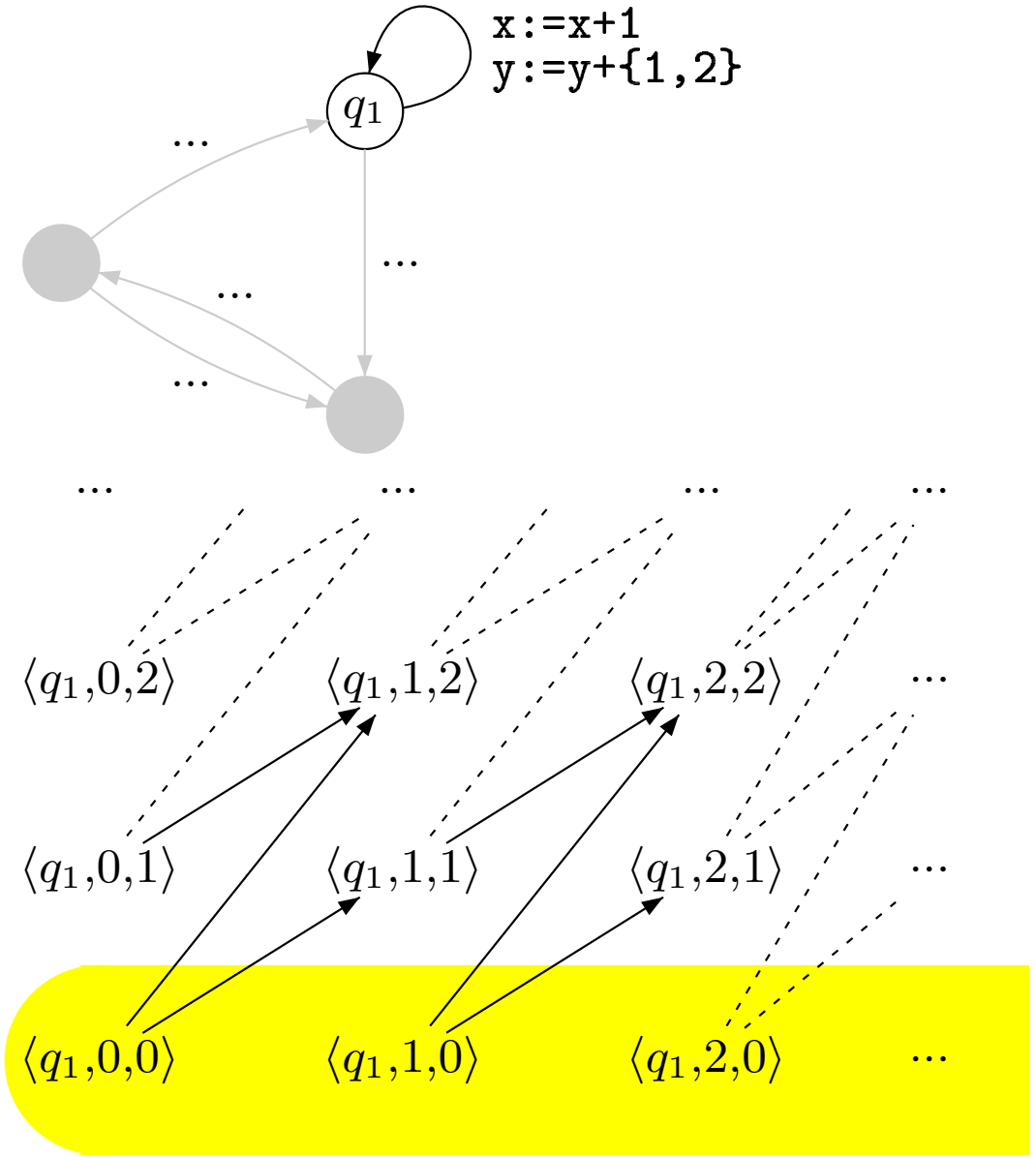
```
channel c
   process P1(u)
    send u -> c
      ...
||  process P2(v)
    receive c <- n
    if #P1<#P2 ···
```





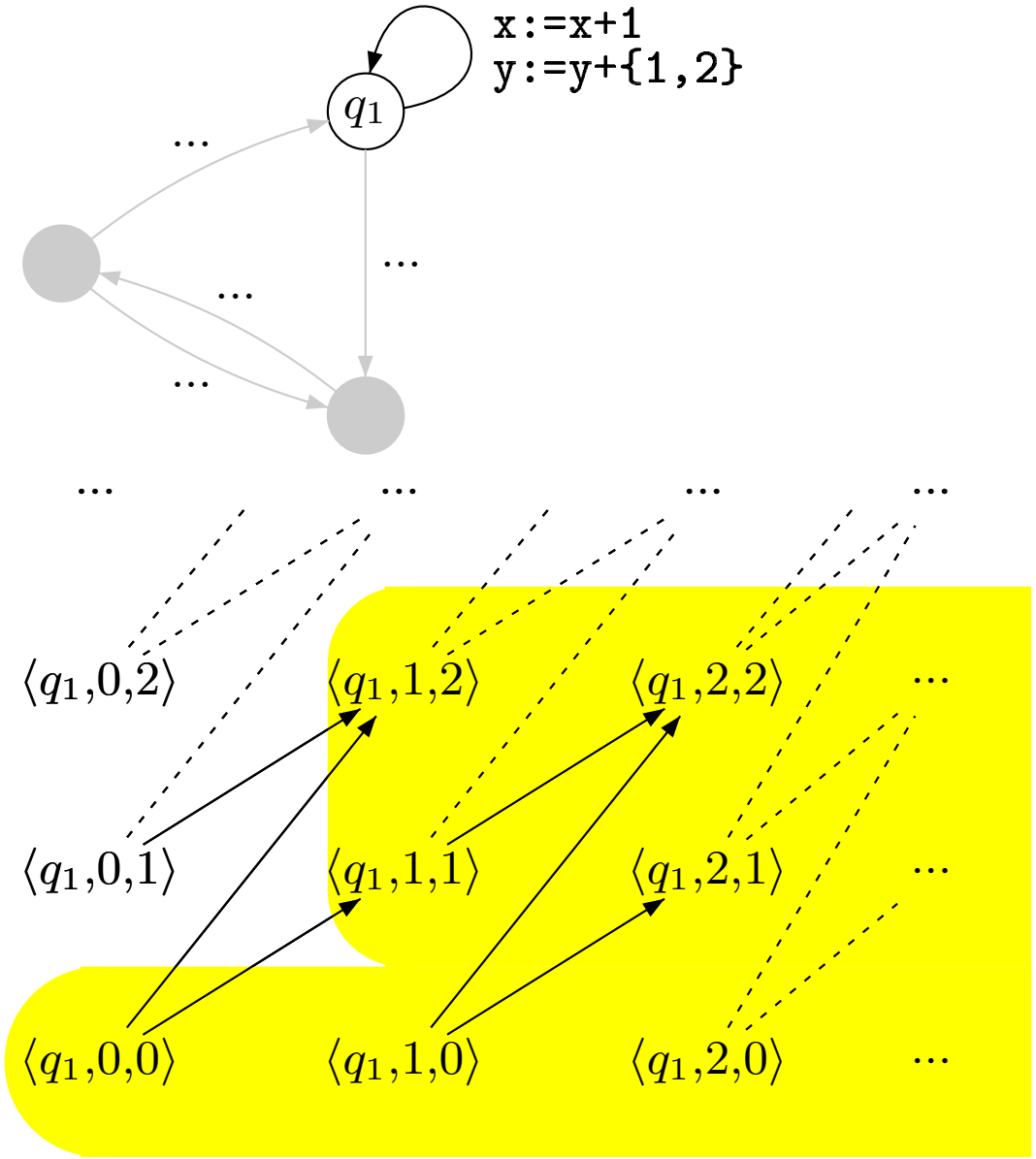
$$\mathcal{S}=\langle q_1,\geq 0,=0\rangle$$

Techniques d'accélération



$\mathcal{S}=\langle q_1,\geq 0,=0\rangle$

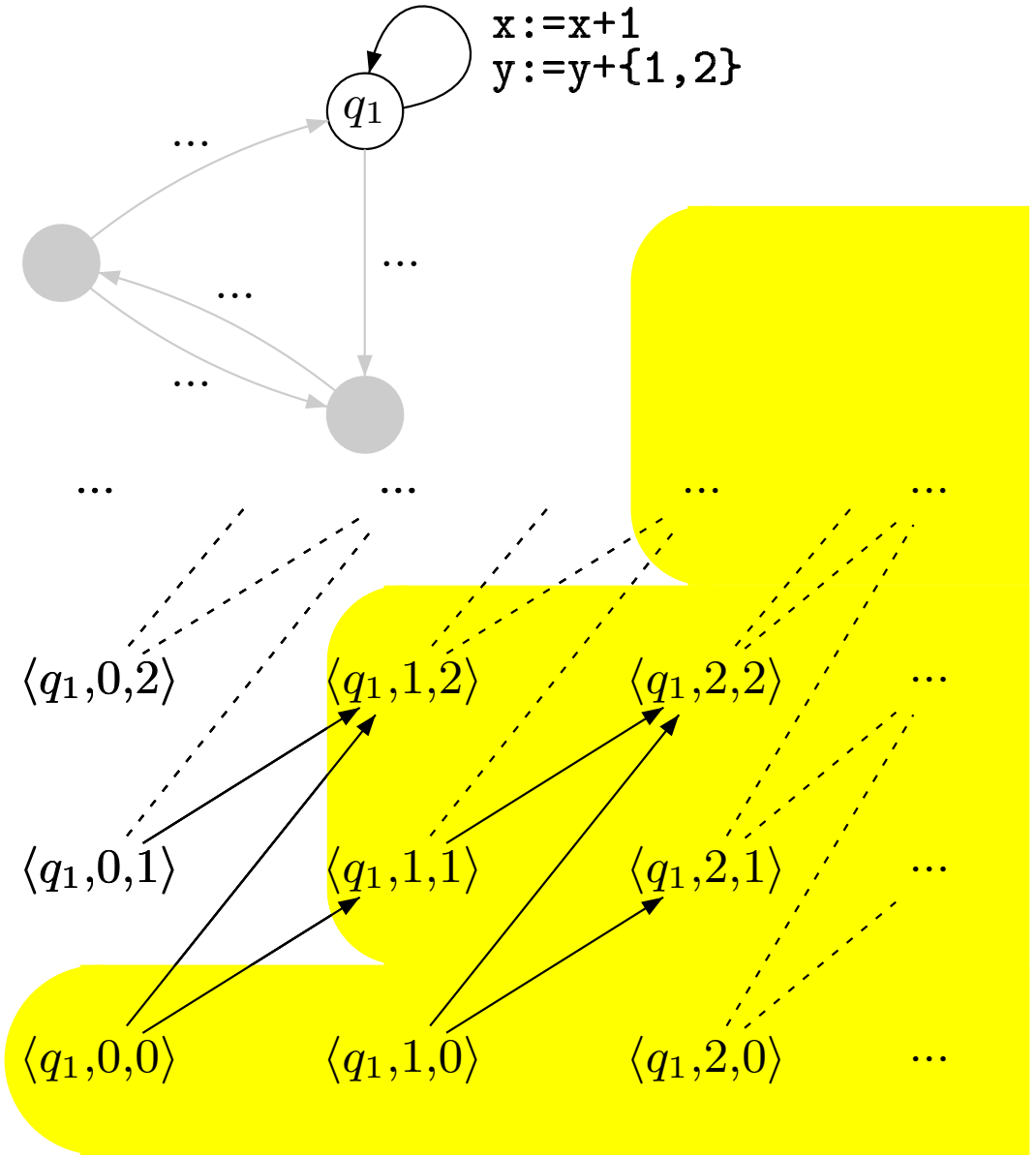
Techniques d'accélération



$\mathcal{S}'=\langle q_1,\geq 0,=0\rangle+\langle q_1,\geq 1,\in\{1,2\}$

$\mathcal{S}=\langle q_1,\geq 0,=0\rangle$

Techniques d'accélération

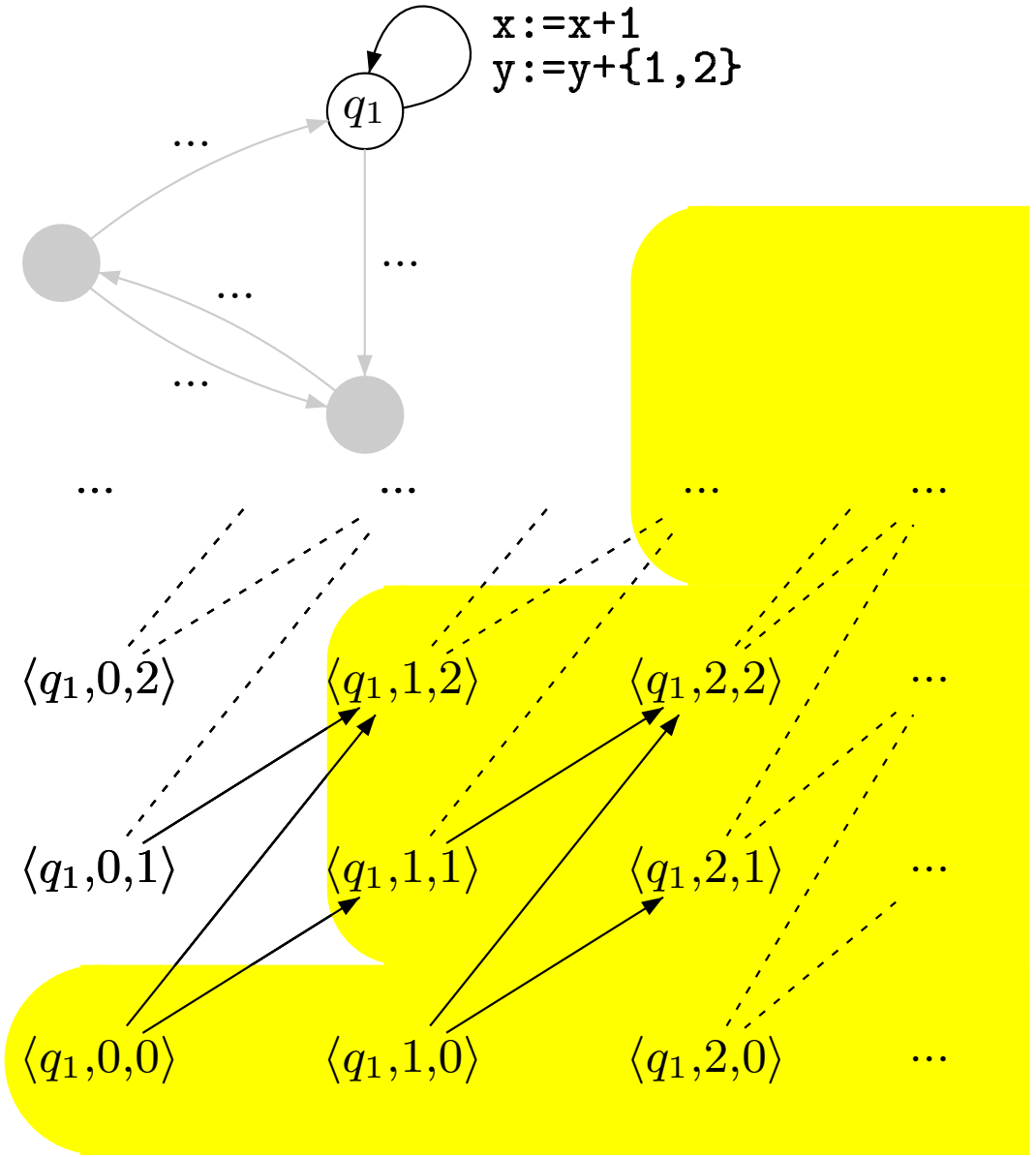


$\mathcal{S}''=\cdots$

$\mathcal{S}'=\langle q_1,\geq 0,=0\rangle+\langle q_1,\geq 1,\in\{1,2\}$

$\mathcal{S}=\langle q_1,\geq 0,=0\rangle$

Techniques d'accélération



$\mathcal{S}^{\infty} = \langle q_1,a,b\rangle \mid b \leq 2a$

$\mathcal{S}'' = \ldots$

$\mathcal{S}' = \langle q_1,\geq 0,=0\rangle + \langle q_1,\geq 1,\in\{1,2\}$

$\mathcal{S} = \langle q_1,\geq 0,=0\rangle$

Analyse symbolique d'accessibilité

Calcul symbolique

Principaux résultats – 1

- Une théorie de l'« accélération plate » :

— Stratégies complètes et caractérisation de la complétude [C+B]

— Explique et généralise de nombreux résultats sur les systèmes semilinéaires [C+B]

Principaux résultats – 1

• Une théorie de l'« accélération plate » :

— Stratégies complètes et caractérisation de la complétude [C+B]

— Explique et généralise de nombreux résultats sur les systèmes semilinéaires [C+B]

• Au delà des propriétés de sûreté :

— Vivacité dans le Regular model checking [P]

— NPLCS (canaux FIFO + pertes stochastiques) [C]

Principaux résultats – 2

- Combinaison de représentations symboliques :

— Un cadre théorique avec accélération [C]

Principaux résultats – 2

• Combinaison de représentations symboliques :

— Un cadre théorique avec accélération [C]

• De nouvelles représentations symboliques :

— SMS (graphes avec contraintes arithmétiques) pour l'allocation dynamique de pointeurs [C]

— Récursivité + parallélisme : réseaux d'automates à piles [P]

Principaux résultats – 2

• Combinaison de représentations symboliques :

— Un cadre théorique avec accélération [C]

• De nouvelles représentations symboliques :

— SMS (graphes avec contraintes arithmétiques) pour l'allocation dynamique de pointeurs [C]

— Récursivité + parallélisme : réseaux d'automates à piles [P]

• Algorithmique des représentations symboliques :

— Représentation par automates partagés [B→C]

— Algorithmique des UBA [C→B]

Abstractions et méthodes symboliques

Vérification de programmes

Principaux résultats – 3

• « Abstract regular model checking » [P]

— Combine « regular model checking » et l'approche « abstract - check - refine »

— NB1 : le raffinement est basé sur les contre-exemples

— NB2 : abstraction sur des automates avec inférence d'ensembles réguliers

Principaux résultats – 3

• « Abstract regular model checking » [P]

— Combine « regular model checking » et l'approche « abstract - check - refine »

— NB1 : le raffinement est basé sur les contre-exemples

— NB2 : abstraction sur des automates avec inférence d'ensembles réguliers

• « Abstract - check - refine » sur des modèles infinis (piles + multithread) [P]

Environnement intégré

Mise en oeuvre et partage

Développements

Objectifs = Environnement de vérification symbolique permettant d'utiliser et de piloter les représentation symboliques et les stratégies de model-checkers différents (FAST et TReX)

Développements

Objectifs = Environnement de vérification symbolique permettant d'utiliser et de piloter les représentation symboliques et les stratégies de model-checkers différents (FAST et TReX)

- Travail effectué :

— Spécification formelle des interfaces entre outils

— Extension du langage AltaRica pour décrire les modèles

— Adaptation de l'architecture de TReX

Développements

Objectifs = Environnement de vérification symbolique permettant d'utiliser et de piloter les représentation symboliques et les stratégies de model-checkers différents (FAST et TReX)

• Travail effectué :

— Spécification formelle des interfaces entre outils

— Extension du langage AltaRica pour décrire les modèles

— Adaptation de l'architecture de TReX

• En cours : intégration des composants et exemples de vérification utilisant toute la chaîne

Développements

Objectifs = Environnement de vérification symbolique permettant d'utiliser et de piloter les représentation symboliques et les stratégies de model-checkers différents (FAST et TReX)

• Travail effectué :

— Spécification formelle des interfaces entre outils

— Extension du langage AltaRica pour décrire les modèles

— Adaptation de l'architecture de TReX

• En cours : intégration des composants et exemples de vérification utilisant toute la chaîne

• Au delà : communication entre model-checkers au niveau des structures de données symboliques ?

Conclusions et perspectives

- Collaboration réussie :

— échanges riches, ouverts et fructueux

— collaboration concrétisée en termes de mobilité

— concrétisée en termes de recherches croisées

— à concrétiser en termes de publications communes

- A suscité le projet RNTL AVERILES 2005–2008

— LSV + LIAFA + VERIMAG + EDF + CRIL Technology

— Thème = regular model checking et accélération pour l'allocation de pointeurs