VeriFAST : architecture d'un model-checker accéléré générique

S. Bardin, G. Cécé, A. Finkel, F. Herbreteau, J. Leroux,
D. Nowak, G. Sutre, A. Vincent

Laboratoire Spécification et Vérification

LSV, Ecole Normale Sup. de Cachan
LaBRI, Université de Bordeaux 1
LIFC, Université de Franche-Comté

LaBRI

Outline

1. Introduction
2. Vérification symbolique accélérée
3. Automates étendus à actions gardées typées
4. Architecture

Outline

1. Introduction
2. Vérification symbolique accélérée
3. Automates étendus à actions gardées typées
4. Architecture

Motivations

- Implantation des techniques de vérification symbolique accélérée

On aimerait pouvoir :

Motivations

- Implantation des techniques de vérification symbolique accélérée

On aimerait pouvoir :

- Comparer (objectivement) ces techniques
  - représentations symboliques
  - accélérations

Motivations

- Implantation des techniques de vérification symbolique accélérée

On aimerait pouvoir :

- Comparer (objectivement) ces techniques
  - représentations symboliques
  - accélérations
- Combiner ces techniques
  - analyse de modèles hétérogènes (plusieurs types de données)

Motivations

- Implantation des techniques de vérification symbolique accélérée

On aimerait pouvoir :

- Comparer (objectivement) ces techniques
  - représentations symboliques
  - accélérations
- Combiner ces techniques
  - analyse de modèles hétérogènes (plusieurs types de données)
- Faciliter l'expérimentation de nouvelles techniques
  - nouveaux types de données et/ou d'opérations

Motivations (suite)

- Constat :
  - les outils existants ne sont pas facilement extensibles
  - les techniques existantes partagent de nombreux concepts

Motivations (suite)

- Constat :
  - les outils existants ne sont pas facilement extensibles
  - les techniques existantes partagent de nombreux concepts
- Groupe VeriFAST :
  - conception (et implantation) d'un model-checker accéléré générique
  - comparaison, combinaison et expérimentation
  - réutilisation de bibliothèques de représentations symboliques

Outline

1. Introduction
2. Vérification symbolique accélérée
3. Automates étendus à actions gardées typées
4. Architecture

Automates étendus homogènes

- Automate étendu : $(Q, V, \mathtt{Op}, \rightarrow)$
  - Q : localités (de contrôle), V : variables, $\mathtt{Op}$: opérations
  - $\rightarrow \subseteq Q \times \mathtt{Op} \times Q$: transitions (de contrôle)

Automates étendus homogènes

- Automate étendu : $(Q, V, \mathtt{Op}, \rightarrow)$
  - Q : localités (de contrôle), V : variables, $\mathtt{Op}$: opérations
  - $\rightarrow \subseteq Q \times \mathtt{Op} \times Q$: transitions (de contrôle)
- Sémantique des données : $(V, \mathbb{D}, \delta)$
  - $\delta : \mathbb{D}^V \times \mathtt{Op} \rightarrow 2^{\mathbb{D}^V}$

Automates étendus homogènes

- Automate étendu : $(Q, V, \mathtt{Op}, \rightarrow)$
  - Q : localités (de contrôle), V : variables, $\mathtt{Op}$: opérations
  - $\rightarrow \subseteq Q \times \mathtt{Op} \times Q$: transitions (de contrôle)
- Sémantique des données : $(V, \mathbb{D}, \delta)$
  - $\delta : \mathbb{D}^V \times \mathtt{Op} \rightarrow 2^{\mathbb{D}^V}$
- Sémantique opérationnelle : $(S, \mathtt{Op}, \rightarrow)$
  - $S = Q \times \mathbb{D}^V$
  - $\rightarrow \subseteq S \times \mathtt{Op} \times S$ défini par :

$$(q, d) \xrightarrow{\mathtt{op}} (q', d') \text{ si } q \xrightarrow{\mathtt{op}} q' \text{ et } d' \in \delta(d, \mathtt{op})$$

Automates étendus homogènes

- Automate étendu : $(Q, V, \mathtt{Op}, \rightarrow)$
  - Q : localités (de contrôle), V : variables, $\mathtt{Op}$: opérations
  - $\rightarrow \subseteq Q \times \mathtt{Op} \times Q$: transitions (de contrôle)
- Sémantique des données : $(V, \mathbb{D}, \delta)$
  - $\delta : \mathbb{D}^V \times \mathtt{Op} \rightarrow 2^{\mathbb{D}^V}$
- Sémantique opérationnelle : $(S, \mathtt{Op}, \rightarrow)$
  - $S = Q \times \mathbb{D}^V$
  - $\rightarrow \subseteq S \times \mathtt{Op} \times S$ défini par :

$$(q, d) \xrightarrow{\mathtt{op}} (q', d') \text{ si } q \xrightarrow{\mathtt{op}} q' \text{ et } d' \in \delta(d, \mathtt{op})$$

- On souhaite calculer : $post^*(q_0, \mathbb{D}^V)$

Représentation symbolique

- Représentation symbolique : $(R, \bot, \sqcup, \widehat{\delta}, [\![\cdot]\!])$

Représentation symbolique

- Représentation symbolique : $(R, \bot, \sqcup, \widehat{\delta}, [\![\cdot]\!])$
- Ensemble R de régions, interprétation donnée par $[\![\cdot]\!] : R \to 2^{\mathbb{D}^V}$

Représentation symbolique

- Représentation symbolique : $(R, \bot, \sqcup, \widehat{\delta}, [\![\cdot]\!])$
- Ensemble R de régions, interprétation donnée par $[\![\cdot]\!] : R \to 2^{\mathbb{D}^V}$
- pré-ordre d'inclusion induit $\sqsubseteq$ entre régions : $r \sqsubseteq r'$ ssi $[\![r]\!] \subseteq [\![r']\!]$

Représentation symbolique

- Représentation symbolique : $(R, \bot, \sqcup, \widehat{\delta}, [\![\cdot]\!])$
- Ensemble R de régions, interprétation donnée par $[\![\cdot]\!] : R \to 2^{\mathbb{D}^V}$
- pré-ordre d'inclusion induit $\sqsubseteq$ entre régions : $r \sqsubseteq r'$ ssi $[\![r]\!] \subseteq [\![r']\!]$
- région vide $\bot$ ($[\![\bot]\!] = \emptyset$)

Représentation symbolique

- Représentation symbolique : $(R, \bot, \sqcup, \widehat{\delta}, [\![\cdot]\!])$
- Ensemble R de régions, interprétation donnée par $[\![\cdot]\!] : R \to 2^{\mathbb{D}^V}$
- pré-ordre d'inclusion induit $\sqsubseteq$ entre régions : $r \sqsubseteq r'$ ssi $[\![r]\!] \subseteq [\![r']\!]$
- région vide $\bot$ ($[\![\bot]\!] = \emptyset$)
- Union de régions : $[\![r \sqcup r']\!] = [\![r]\!] \cup [\![r']\!]$

Représentation symbolique

- Représentation symbolique : $(R, \bot, \sqcup, \widehat{\delta}, [\![\cdot]\!])$
- Ensemble R de régions, interprétation donnée par $[\![\cdot]\!] : R \to 2^{\mathbb{D}^V}$
- pré-ordre d'inclusion induit $\sqsubseteq$ entre régions : $r \sqsubseteq r'$ ssi $[\![r]\!] \subseteq [\![r']\!]$
- région vide $\bot$ ($[\![\bot]\!] = \emptyset$)
- Union de régions : $[\![r \sqcup r']\!] = [\![r]\!] \cup [\![r']\!]$
- $\widehat{\delta}$ symbolique : $R \times \mathtt{Op} \to R$ satisfaisant :

$$[\![\widehat{\delta}(r, \mathtt{op})]\!] = \{d' \in \mathbb{D}^V \mid \exists\, d \in \mathbb{D}^V, d' \in \delta(\mathtt{op}, d)\}$$

Représentation symbolique

- Représentation symbolique : $(R, \bot, \sqcup, \widehat{\delta}, [\![\cdot]\!])$
- Ensemble R de régions, interprétation donnée par $[\![\cdot]\!] : R \to 2^{\mathbb{D}^V}$
- pré-ordre d'inclusion induit $\sqsubseteq$ entre régions : $r \sqsubseteq r'$ ssi $[\![r]\!] \subseteq [\![r']\!]$
- région vide $\bot$ ($[\![\bot]\!] = \emptyset$)
- Union de régions : $[\![r \sqcup r']\!] = [\![r]\!] \cup [\![r']\!]$
- $\widehat{\delta}$ symbolique : $R \times \mathtt{Op} \to R$ satisfaisant :

$$[\![\widehat{\delta}(r, \mathtt{op})]\!] = \{d' \in \mathbb{D}^V \mid \exists\, d \in \mathbb{D}^V, d' \in \delta(\mathtt{op}, d)\}$$

- On peut rajouter $\top$ et $\sqcap$ (et $\setminus$).

Exemples

- Automates à files :
  - QDD
  - CQDD
  - SRE
  - SLRE

Exemples

- Automates à files :
  - QDD
  - CQDD
  - SRE
  - SLRE
- Automates à compteurs :
  - Ensembles clos supérieurement (inférieurement)
  - Ensembles semilinéaires
  - Formules de Presburger
  - NDD

Exemples

- Automates à files :
  - QDD
  - CQDD
  - SRE
  - SLRE
- Automates à compteurs :
  - Ensembles clos supérieurement (inférieurement)
  - Ensembles semilinéaires
  - Formules de Presburger
  - NDD
- Automates temporisés et hybrides
  - DBM
  - Unions finies de polyèdres convexes
  - RVAs

Accéleration

- Calcul de $post^*$ par itération de point fixe peut ne générer que des régions “finies”
- Idée: calculer en une étape l’effet de l’itération d’un circuit de contrôle donné

Accélération

- Calcul de $post^*$ par itération de point fixe peut ne générer que des régions “finies”
- Idée: calculer en une étape l’effet de l’itération d’un circuit de contrôle donné
- Accélération d’une séquence $\sigma \in \mathtt{Op}^+$: fonction $\widehat{\sigma} : R \to R$ telle que

$$[\![\widehat{\sigma}(r)]\!] = \{d' \in \mathbb{D}^V \mid \exists d \in \mathbb{D}^V, \exists i \in \mathbb{N}, d' \in \delta(\sigma^i, op)\}$$

Accélération

- Calcul de $post^*$ par itération de point fixe peut ne générer que des régions "finies"
- Idée: calculer en une étape l'effet de l'itération d'un circuit de contrôle donné
- Accélération d'une séquence $\sigma \in \mathtt{Op}^+$: fonction $\widehat{\sigma} : R \to R$ telle que

$$[\![\widehat{\sigma}(r)]\!] = \{d' \in \mathbb{D}^V \mid \exists d \in \mathbb{D}^V, \exists i \in \mathbb{N}, d' \in \delta(\sigma^i, op)\}$$

- Stratégies d'accélération : statiques (méta-transitions) et/ou dynamiques (heuristiques)

Outline

1. Introduction
2. Vérification symbolique accélérée
3. Automates étendus à actions gardées typées
4. Architecture

Structuration des opérations

- Affectations gardées : $\varphi \rightarrow v_1 := t_1 \,||\, v_2 := t_2 \,||\, \cdots \,||\, v_n := t_n$
  - φ : prédicat
  - v_i : variables
  - t_i : termes

Structuration des opérations

- Affectations gardées : $\varphi \rightarrow v_1 := t_1 \,||\, v_2 := t_2 \,||\, \cdots \,||\, v_n := t_n$
  - φ : prédicat
  - v_i : variables
  - t_i : termes
- logique du premier ordre typée

Structuration des opérations

- Affectations gardées : $\varphi \rightarrow v_1 := t_1 \,||\, v_2 := t_2 \,||\, \cdots \,||\, v_n := t_n$
  - φ : prédicat
  - v_i : variables
  - t_i : termes
- logique du premier ordre typée
- les types de base et les mots-clés (fonctions et relations) sont définis dans l'automate étendu
  - généricité du formalisme de description des systèmes

Structuration des opérations

- Affectations gardées : $\varphi \rightarrow v_1 := t_1 \,||\, v_2 := t_2 \,||\, \cdots \,||\, v_n := t_n$
  - φ : prédicat
  - v_i : variables
  - t_i : termes
- logique du premier ordre typée
- les types de base et les mots-clés (fonctions et relations) sont définis dans l'automate étendu
  - généricité du formalisme de description des systèmes
- la sémantique de l'automate dépend de l'interprétation des types de base et des mots-clés
  - exemple : FIFO exacte / FIFO avec pertes

Automates étendus à actions gardées typées

- Signature (“langage de description”) :
  - $\mathbb{D}_i$: types de base (par ex. int)
  - f_i : fonctions (et constantes) de base (par ex. $plus$)
  - R_i : relations de base (par ex. inf)

Automates étendus à actions gardées typées

- Signature (“langage de description”) :
  - $\mathbb{D}_i$: types de base (par ex. int)
  - f_i : fonctions (et constantes) de base (par ex. $plus$)
  - R_i : relations de base (par ex. inf)
- Automate étendu à actions gardées typées :
  - L : signature
  - Q : localités (de contrôle), V : variables typées
  - $I(q)$: invariant de la localité $q \in Q$
  - transitions de contrôles étiquetées par des affectations gardées
  - le typage doit être respecté

Automates étendus à actions gardées typées

- Signature (“langage de description”) :
  - $\mathbb{D}_i$: types de base (par ex. int)
  - f_i : fonctions (et constantes) de base (par ex. $plus$)
  - R_i : relations de base (par ex. inf)
- Automate étendu à actions gardées typées :
  - L : signature
  - Q : localités (de contrôle), V : variables typées
  - $I(q)$: invariant de la localité $q \in Q$
  - transitions de contrôles étiquetées par des affectations gardées
  - le typage doit être respecté
- Interprétation : L-structure (par exemple $\langle int \mapsto \mathbb{N}, plus \mapsto +, inf \mapsto \leq \rangle$)

Automates étendus à actions gardées typées

- Signature (“langage de description”) :
  - $\mathbb{D}_i$: types de base (par ex. int)
  - f_i : fonctions (et constantes) de base (par ex. $plus$)
  - R_i : relations de base (par ex. inf)
- Automate étendu à actions gardées typées :
  - L : signature
  - Q : localités (de contrôle), V : variables typées
  - $I(q)$: invariant de la localité $q \in Q$
  - transitions de contrôles étiquetées par des affectations gardées
  - le typage doit être respecté
- Interprétation : L-structure (par exemple $\langle int \mapsto \mathbb{N}, plus \mapsto +, inf \mapsto \leq \rangle$)
- Sémantique opérationnelle donnée relativement à une interprétation

Exemple

```
Domain Int ;;
val 0, 3, 30 : Int ;;
val elapse, ++ : Int -> Int ;;
rel =, <, <= : Int * Int ;;

System Description Timeout ::
  states
    WAIT {x, y : Int},
    OK,
    ERR;

  WAIT[initial (=(x,0)&&=(y,0))]::
    t1 :: =(x,30) && <=(y,3) -> x:=0, y:=++(y) -> WAIT ;;
    t2 :: <(x,30) && <(y,3) -> x:=elapse(x), y:=y -> WAIT ;;
    t3 :: true -> nop -> OK ;;
    t4 :: =(x,30) && <(3,y) -> nop -> ERR ;;
  ;;
;;
```

Outline

1. Introduction
2. Vérification symbolique accélérée
3. Automates étendus à actions gardées typées
4. Architecture

Aperçu global de l'architecture

- module `TypedLogic`
  - termes, prédicats, typage, signature
  - seul module dont l'implantation est fixée
- modules de type `SYSTEM_DESCRIPTION`
  - représentation d'automates étendus à actions gardées typées
- modules de type `REGION`
  - $(R, \bot, \top, \equiv, \sqsubseteq, \sqcup, \sqcap)$
- modules de type `SYMBOLIC_REPRESENTATION`
  - module Region + fonctions $(post, pre)$
- modules de type `ACCELERATION`
  - module Region + fonction $accelerate$

Aperçu global de l'architecture (suite)

- modules de type `ACCELERATED_SYMBOLIC_REPRESENTATION`
  - module Region + module Acceleration (même module Région)
- modules de type `MODEL_CHECKER`
  - module SystemDescription + module AcceleratedSymbRepr + fonction $post^*$
- foncteurs de type `MODEL_CHECKER_MAKER`
  - arguments : modules SystemDescription, AcceleratedSymbRepr
  - sortie : module ModelChecker (ajout de la fonction $post^*$)
- foncteurs pour composer les modules de représentation symbolique accélérée

Intégration

- Interprétation des mots-clés donnée dans les représentations symboliques
  - extensibilité du langage de description de systèmes
  - expérimentation de nouvelles représentations symboliques et accélérations
- Représentations symboliques sous forme de “plugins”
  - composition cartésienne des ces représentations symboliques
- Stratégies de calcul de $post^*$: dans les model-checkers
- Permet de comparer les techniques symboliques