

Action Spécifique n° 22

Vérification des propriétés quantitatives

<http://www-verimag.imag.fr/AS22/>

Rapport final

N. Halbwachs¹ Ph. Schnoebelen²

21 janvier 2003

¹VÉRIMAG, Centre Équation, 2, av. de Vignate, 38610 Gières Cedex. Tél : (+33/0) 4.76.63.48.55.

²LSV, ENS de Cachan, 61, av. Pdt Wilson, 942365 Cachan Cedex. Tél : (+33/0) 1.47.40.75.30.

Table des matières

1	L'AS 22 en bref	5
1.1	Le périmètre de l'Action	5
1.2	La problématique de l'Action	5
1.3	Bilan résumé	6
1.4	Perspectives	7
2	Activités de Recherche	7
2.1	Techniques de Vérification	7
2.1.1	Analyse d'accessibilité de réseaux de Petri avec arithmétique réelle	7
2.1.2	Analyse de systèmes hybrides	8
2.1.3	Vérification de Réseaux Paramétrés	10
2.1.4	Analyse de Programmes Parallèles avec Procédures	10
2.2	Représentations symboliques	10
2.2.1	Algorithmique des polyèdres convexes	10
2.2.2	Représentation symboliques par automates à compteurs	11
2.2.3	Représentations mixtes Numériques/Booléennes	11
2.3	Études de cas et expérimentations	12
2.3.1	L'algorithme TTP/C	12
2.3.2	Analyse du protocole PGM avec TReX	12
2.3.3	Analyse du protocole PGM avec UPPAAL	12
2.4	Développement d'outils	13
2.4.1	TReX	13
3	Activités d'animation	14
3.1	Journées « Vérification des Propriétés Quantitatives »	14
3.2	Workshop à Grenoble en mars 2003	14
	Références	15
A	Bilan Financier	17
B	Programmes des journées VPQ	17
B.1	Journée du 18 décembre 2001 à Cachan	17
B.2	Journée du 28 mars 2002 à Cachan	17
B.3	Journée du 27 juin 2002 à Paris	18
B.4	Journée du 13 novembre 2002 à Cachan	18

1 L'AS 22 en bref

1.1 Le périmètre de l'Action

L'Action Spécifique 22 a réuni, d'octobre 2001 à décembre 2002, une vingtaine de chercheurs du RTP SECC (Systèmes embarqués, complexes ou contraints) autour du thème de la « Vérification des propriétés quantitatives ».

Les partenaires sont essentiellement issus de quatre laboratoires :

LSV (Cachan) : Ph. Schnoebelen, A. Finkel, F. Laroussinie, B. Bérard, L. Fribourg, P. Bouyer, ...

Vérimag (Grenoble) équipes « Systèmes synchrones » et « Systèmes temporisés et hybrides » : N. Halbwachs, S. Yovine, S. Tripakis, E. Asarin, P. Raymond, ...

LIAFA (Paris 7) équipe « Modélisation et Vérification » : A. Bouajjani, P. Habermehl, M. Sighireanu, Y. Jurski, ...

IRCCyN (Nantes) équipe « Systèmes Temps Réel » : O. Roux, Ch. Mauras, ...

et aussi : B. Jeannet (IRISA) et G. Sutre (LaBRI).

Toutefois, l'Action est restée ouverte à une communauté plus large, et nos journées scientifiques ont permis de rassembler des chercheurs venus d'Orsay, de Belgique, de Toulouse, de Besançon, de Poitiers, de Bordeaux, etc.

L'Action a reçu un budget de 45kEUR (cf. bilan financier).

1.2 La problématique de l'Action

À l'origine de la création de l'AS22 se trouvait la volonté de rassembler des chercheurs qui, bien qu'issus de communautés séparées, avaient en commun le besoin de vérifier des « propriétés quantitatives » (le terme choisi pour désigner le terrain de convergence retenu).

Les frontières entre ces communautés étaient celles des modèles sous-jacentes (systèmes temporisés *vs.* systèmes à compteur *vs.* systèmes probabilistes *vs.* ...), celles des questions d'intérêt (vérification paramétrée *vs.* calculs de propriétés numériques *vs.* ...), et celles des technique (model-checking symbolique *vs.* interprétation abstraite).

En stimulant les échanges entre ses participants, l'AS 22 voulait :

- identifier les problèmes communs qui n'étaient pas perçus comme tels ;
- permettre à certaines techniques, mises au point et validées par une des communautés, d'être connues et peut-être adoptées par les autres communautés ;
- créer des synergies entre ces équipes.

L'Action a alors eu trois types d'activité :

- des rencontres régulières entre ses participants, permettant les échanges visés ;
- l'organisation de 4 journées scientifiques qui ont rassemblé à chaque fois de 20 à 40 participants (dont de nombreux chercheurs extérieurs à l'Action) ;
- l'organisation d'un workshop « Vérification des propriétés quantitatives » (du 5 au 7 mars 2003 à Grenoble).

1.3 Bilan résumé

En permettant la création d'un groupe français de chercheurs centrés sur le problème de la vérification de propriétés quantitatives, et en soutenant son activité, l'Action a eu un impact très positif sur la synergie entre nos recherches.

De fait, les confrontations de nos différentes approches, qu'elles soient sur les modèles, sur les algorithmes de vérification, sur les outils, ou sur les résultats, ont permis de dégager des convergences. Nous en donnons quelques exemples :

Les modèles opérationnels : De nombreux systèmes *a priori* fort différents sont en fait représentés formellement au moyen des mêmes modèles opérationnels. Ainsi, les automates à compteurs, classiquement utilisés pour les systèmes gérant des ressources ou pour des systèmes à temps discret, sont aussi utilisés pour représenter des systèmes distribués paramétrés (§ 2.1.3), pour abstraire des programmes parallèles avec procédures récursives (§ 2.1.4).

Les représentations symboliques : Les contraintes sur \mathbb{R}^n sont classiquement utilisées pour le model-checking symbolique de systèmes temporisés (un cas dans lequel on se contente habituellement d'inégalités affines simples) ou hybrides (où on manipule souvent des polyèdres plus compliqués). Les polyèdres sont aussi utilisés pour l'interprétation abstraite d'automates à compteurs (§ 2.2.1). Ici le passage de parties de \mathbb{N}^n à des parties de \mathbb{R}^n est une astuce qui permet de disposer d'algorithmes plus efficaces au prix d'une légère approximation supérieure du résultat.

Il se trouve que les calculs sur \mathbb{R}^n peuvent être utilisés pour évaluer efficacement, *et ceci de façon exacte*, les ensembles d'accessibilité de certains automates à compteurs (des réseaux de Petri) qui sont en fait des parties de \mathbb{N}^n (§ 2.1.1).

Dans l'autre direction, un résultat récent très impressionnant est la preuve qu'on peut représenter de manière effective la relation d'accessibilité des automates temporisés dans la théorie additive des réels (une légère extension de la logique de Presburger qui marie les entiers et les réels) [CJ99].

Les algorithmes : L'intérêt de telle ou telle représentation symbolique dépend fortement de l'algorithmique qui l'accompagne (y compris les structures de données). Les avancées dans ce domaine peuvent profiter à plusieurs modèles, et ensuite à une large gamme de familles de systèmes. Nous décrivons plus bas divers exemples de tels travaux : § 2.1.2, § 2.2.1, § 2.2.3.

Sur ce thème, les travaux décrits dans la section 2.2.2 fournissent un exemple remarquable : les automates à compteurs sont utilisés non pas comme modèles opérationnels mais comme représentations symboliques des ensembles de configurations ! En effet, les systèmes analysés sont modélisés par des automates à piles particuliers et les configurations sont des mots (les contenus des piles). Mais la représentation d'ensembles de mots par des langages réguliers est parfois insuffisante et il faut pouvoir imposer des contraintes numériques (pour représenter par exemples tous les contenus de pile de la forme $a^n b^m$ avec $n \leq m$) qu'on décrit facilement avec des automates à compteurs. Ainsi les algorithmes symboliques sur les automates à compteurs servent

à construire des algorithmes symboliques sur les automates à piles.

1.4 Perspectives

La durée de l'Action a été trop courte pour que les synergies décrites plus haut aient pu déboucher sur des publications communes. Les participants ont toutefois émis le souhait de poursuivre la collaboration amorcée via l'Action.

Une proposition en ce sens est en cours de rédaction. Les porteurs identifiés sont A. Bouajjani (LIAFA, Paris 7) et Ph. Schnoebelen (LSV, Cachan).

1. Il a été décidé de mettre l'accent sur le problème central identifié lors de cette première année : les représentations symboliques pour les ensembles numériques utilisés en vérification, les structures de données et les algorithmes associés. L'objectif que se fixe le groupe est de produire un document de synthèse recensant les meilleures solutions proposées par la communauté mondiale, cernant leur portée (domaine d'application, ...), et identifiant les plus importantes parmi les questions encore non tranchées.
2. Par ailleurs, durant le cours de l'Action nous nous sommes rendus compte que notre problématique avait une portée plus large que la seule vérification symbolique (que ce soit par model-checking ou par interprétation abstraite). Il existe d'autres domaines où les contraintes arithmétiques servent à prédire le comportement de systèmes, par exemple la *shape analysis* en compilation des systèmes embarqués, ou l'analyse des modèles stochastiques. Nous prévoyons d'élargir notre groupe de sorte qu'y figurent des représentants de ces domaines cousins.

2 Activités de Recherche

Cette section, recensant la plupart des techniques, exemples, problèmes, etc., qui ont été présentés lors de nos réunions de travail, donne un aperçu des modèles, algorithmes, propriétés, outils, et études de cas qui ont délimité notre recherche.

2.1 Techniques de Vérification

2.1.1 Analyse d'accessibilité de réseaux de Petri avec arithmétique réelle

Pour certaines classes de réseaux de Petri, l'ensemble des marquages accessibles peut être représenté par une formule de l'arithmétique de Presburger. Ceci a conduit certaines équipes à développer des outils de résolutions permettant le calcul des configurations accessibles pour différents types de modèles, correspondant à des automates étendus. Cependant, la terminaison de ces procédures n'est pas garantie et leur complexité est très élevée.

Par ailleurs, l'arithmétique réelle présente souvent des avantages sur l'arithmétique entière, en termes de complexité. Ce point peut être illustré par le problème de la résolution d'inéquations linéaires, qui est polynomial sur les réels et NP-complet sur les entiers. Ce-

pendant, l'utilisation de réels conduit généralement à une surapproximation de l'ensemble des solutions.

Dans [BF99], nous montrons qu'il est possible d'analyser certains réseaux de Petri en considérant les compteurs comme des variables réelles, et en utilisant la procédure d'élimination de Fourier-Motzkin. Nous obtenons également des conditions suffisantes pour traiter l'adjonction de méta-transitions, qui permettent l'accélération du calcul d'accessibilité. Nous appliquons cette méthode à plusieurs exemples de réseaux de Petri, et nous obtenons en particulier un résultat nouveau de détection de blocage dans le cas du protocole PNCSA (cartes bancaires) qui avait déjà été analysé par des techniques classiques.

2.1.2 Analyse de systèmes hybrides

L'analyse de modèles mathématiques par le biais de méthodes algorithmiques nécessite au préalable d'une étude approfondie pour déterminer si le problème en question est décidable ou non. Si oui, il est donc possible d'implémenter une procédure pour calculer une réponse exacte. Au cas contraire, il devient nécessaire d'appliquer des techniques approchées. Souvent, celles-ci sont aussi utilisées pour aborder des problèmes décidables mais ayant une complexité qui rend l'analyse exacte pratiquement impossible.

Dans le domaine des systèmes hybrides, la plupart des résultats de décidabilité que l'on trouve dans la bibliographie sont fondés sur l'existence d'une partition finie de l'espace d'états. Pourtant, ces preuves ne donnent pas directement des algorithmes exploitables. Les outils d'analyse existants font donc appel à des méthodes approchées basées sur des représentations symboliques le plus souvent polyédriques ou ellipsoïdales. Le grand avantage de ces techniques est qu'elles ont un très ample spectre. Par contre, leur inconvénient majeur est qu'elles n'exploitent pas les propriétés géométriques des systèmes, ce qui s'est traduit en des analyses trop grossières avec beaucoup d'erreur.

Cette réflexion nous a mené à orienter notre travail vers l'étude de systèmes hybrides susceptibles d'être analysés par des techniques algorithmiques basées sur des propriétés géométriques des comportements dynamiques. Nous avons alors proposé le modèle des systèmes linéaires par morceaux. Il s'agit d'automates hybrides où la dynamique continue est définie par des équations (ou des inclusions) linéaires sur une partition du domaine des variables. Nous avons focalisé nos recherches sur les systèmes à deux dimensions, en particulier les inclusions différentielles polygonales par morceaux [AS02].

Propriété d'accessibilité : Nous appelons inclusion différentielle polygonale par morceaux, un système d'inclusions différentielles de la forme : $\dot{x} \in D_i$, quand $x \in P_i \subset \mathbb{R}^2$, où D_i et P_i sont des polygones, et $\{P_i\}_{i \in I}$ est une partition finie de \mathbb{R}^2 .

Nous avons d'abord étudié le problème de l'accessibilité entre deux points p et q donnés : Existe-t-il un segment de trajectoire qui a comme état initial p et comme état final q ? Nous avons prouvé que ce problème est en effet décidable [ASY01].

Ce résultat est basé sur la représentation des trajectoires par des fonctions multivaluées affines par morceaux. Étant donné un polygone P et deux de ses arêtes e et f , l'ensemble

de tous les points accessibles de n'importe quel point $x \in e$, est défini par une fonction multivaluée affine dont les coefficients dépendent de e , f et du polygone D .

Cette technique (qui reprend l'idée des fonctions de Poincaré) permet de définir un système dynamique unidimensionnel dont le comportement qualitatif est équivalent au système continue en dimension deux. Ceci permet de déterminer la limite d'un segment de trajectoire par un calcul de point-fixe d'une fonction multivaluée affine par morceaux.

Ce résultat a donné lieu à un algorithme permettant de répondre oui ou non à la question s'il est possible d'atteindre un point q du plan à partir d'un point p en suivant la dynamique définie par une inclusion différentielle polygonale. Cet algorithme a été implanté dans le cadre de l'outil SPEEDI, dont l'architecture et les fonctionnalités sont décrites brièvement dans [APSY02] et plus en détail dans [Sch02].

Calcul du portrait de phase : La question de l'accessibilité a été le problème le plus souvent abordé dans le cadre des systèmes hybrides. En revanche, la notion de portrait de phase, pourtant centrale dans la théorie des systèmes dynamiques, a été très peu étudiée. Nous avons donc proposé d'approfondir cette question dans le contexte des inclusions polygonales par morceaux.

Il n'est pas clair a priori ce que c'est que le portrait de phase d'un tel système. Pour commencer, nous avons concentré notre attention sur l'étude des ensembles de trajectoires ayant un comportement cyclique. Nous avons étudié deux notions, celles de noyau de viabilité et de noyau de contrôlabilité [ASY02].

On dit qu'une trajectoire x est viable dans un ensemble K si pour tout $t \geq 0$, $x(t) \in K$. On dit que K est un domaine viable si pour tout $x \in K$ il existe une trajectoire viable dans K qui part de x . On appelle noyau de viabilité d'un ensemble K le plus grand domaine viable dans K . Nous avons étudié le noyau de viabilité pour les trajectoires avec signature cyclique, c'est-à-dire, l'ensemble de points à partir desquels il est possible de continuer à "tourner" dans un cycle d'arêtes.

On dit qu'un ensemble K est contrôlable s'il est possible d'aller de n'importe quel point de K à un voisinage aussi petit que l'on veut de n'importe quel autre point de K par un segment de trajectoire sans jamais sortir de K . On appelle noyau de contrôlabilité le plus grand sous-ensemble de K qui est contrôlable. Nous avons montré que la notion de noyau de contrôlabilité est en l'analogie de celle de cycle limite. En effet, nous avons prouvé que toute trajectoire ayant une signature cyclique converge vers le noyau de contrôlabilité du cycle. Les noyaux de contrôlabilité se sont avérés des objets très importants du portrait de phase d'une inclusion polygonale par morceaux. En effet, nous avons démontré que toute trajectoire sans auto-intersections converge vers l'un de ces noyaux. Ceci est l'analogie du théorème de Poincaré-Bendixon pour les trajectoires simples.

Vérification quantitative des systèmes continus et paramétriques : Ce travail concerne les problèmes de la vérification dans des systèmes hybrides pour lesquels les vitesses d'évolution dans certains états sont des paramètres que l'on cherche à déterminer afin que des propriétés données soient vérifiées. Ceci a donné lieu à un exposé de M.

Adélaïde et O. Roux, dans le cadre des réunions de travail de l'AS-22 le 18 décembre 2001 à Cachan et dont le titre était : "Analyse des systèmes hybrides à pente paramétrique".

2.1.3 Vérification de Réseaux Paramétrés

Dans [BM02], nous développons une méthode permettant de vérifier des réseaux de processus avec un nombre arbitraire de composants basée sur (1) la réduction du problème de la vérification du réseau au problème de l'accessibilité dans un automate à compteur, et (2) résoudre le dernier problème en utilisant les techniques existantes d'analyse symbolique d'accessibilité (techniques incomplètes en général, mais suffisamment efficaces en pratique). Nous appliquons cette approche à un cas non trivial d'algorithme paramétré (le TTP/C utilisé dans l'industrie de l'automobile).

2.1.4 Analyse de Programmes Parallèles avec Procédures

Le problème de la vérification des programmes concurrents avec procédures récursives est indécidable. Dans [BET03] nous proposons une approche algorithmique pour l'analyse de tels programmes basée sur le calcul d'abstractions des langages d'actions de synchronisation exécutables par chacune des composantes parallèles du programme. Nous proposons plusieurs instances de notre approche utilisant des abstractions de différentes précisions et de différents coûts. Nous proposons en particulier des abstractions qui oublient l'ordre entre les actions et ne conservent que l'information (quantitative) sur le nombre d'occurrence de chacune des actions.

2.2 Représentations symboliques

2.2.1 Algorithmique des polyèdres convexes

Les polyèdres convexes (systèmes d'inéquations linéaires) constituent une représentation symbolique classique d'ensembles d'états numériques. L'algorithmique des polyèdres se heurte à des problèmes de complexité, dus surtout au fait qu'elle peut être exponentielle en le nombre des variables. Les travaux en cours sur ce thème concernent donc la réduction du nombre des variables (dimension des polyèdres). Deux approches sont étudiées :

- la détection anticipée des équations : il s'agit, avant d'effectuer une opération sur des polyèdres, de déterminer si des équations linéaires sont satisfaites à la fois par les opérands et par le résultat de l'opération. Chaque équation ainsi déterminée permet, par substitution, d'éliminer une variable.
- la détection de produits cartésiens : lorsque des variables sont indépendantes dans un polyèdre, l'espace peut-être séparé en sous-espaces de dimensions plus petites, qui peuvent être traités séparément. Là encore, il s'agit de détecter cette situation de manière anticipée, avant application des opérations.

Ce travail fait l'objet de la thèse de D. Merchat, en cours.

2.2.2 Représentation symboliques par automates à compteurs

Les automates à compteurs peuvent être utilisés non seulement pour modéliser les comportements de systèmes, mais aussi pour représenter (de manière finie) des ensembles (infinis) de leurs configurations.

Dans [BHM01] cette idée est utilisée pour définir un algorithme de vérification pour une classe de programmes avec procédures récursifs à paramètres entiers.

Dans [HL02] différentes classes d'automates à compteurs sont étudiées du point de vue de l'expressivité et de certaines propriétés de fermeture et de décidabilité, nécessaires pour une utilisation en vérification et dans les algorithmes d'analyse d'accessibilité. Les classes considérées dans cette études sont utiles en particuliers dans l'analyse d'automates communiquant par canaux FIFO [BH99] (modèles naturels pour les protocoles de communication).

2.2.3 Représentations mixtes Numériques/Booléennes

Les travaux à l'IRCCyN de Christophe Mauras et David Garriou pour l'année 2002 ont porté essentiellement sur la finalisation d'une représentation symbolique hétérogène mixant des variables booléennes et des contraintes linéaires sur des variables entières. Ces travaux ont fait l'objet de la thèse de David Garriou soutenue le 20 septembre 2002 [Gar02]. L'objectif était de fournir une structure de données permettant de calculer l'accessibilité d'un automate interprété avec des variables numériques, pour lequel le contrôle dépend des valeurs de ces variables numériques. La structure de donnée proposée (appelée diagrammes de décision à contraintes) est fortement inspiré des BDD utilisés dans le cas fini, mais étendus avec des contraintes linéaires. L'algorithmique qui a été développée, même si elle est complexe et coûteuse en temps de calcul, a permis de réaliser quelques expérimentations intéressantes.

Les algorithmes ont été implantés et utilisés dans le cadre d'un outil original de simulation symbolique pour programmes synchrones. L'intérêt particulier de l'Action spécifique pour ces travaux a été de renforcer la collaboration avec l'équipe de N. Halbwachs (Vérimag) et avec B. Jeannet (Irisa). En particulier les travaux les plus récents de D. Garriou ont permis de définir un opérateur d'élargissement sur les diagrammes de décision à contraintes étendant l'opérateur d'élargissement de Halbwachs et Cousot pour les polyèdres. L'ensemble de ces travaux a été présenté par C. Mauras à Cachan le 28 mars 2002 lors d'un des séminaires de l'Action spécifique.

Parmi les perspectives actuelles, la plus intéressante semble être la comparaison, entre les techniques de calculs symboliques utilisées ici et celles utilisées en programmation logique par contraintes dont en particulier les techniques d'analyse statique. Pour cela, on étudie actuellement le système Timed Default CC de Saraswat et Gupta, qui offre un cadre unifiant permettant d'exprimer la programmation synchrone dans une théorie des programmes concurrents à contraintes (CC), et ainsi d'utiliser les nombreux solveurs disponibles dans le domaine de la programmation logique par contraintes.

2.3 Études de cas et expérimentations

2.3.1 L’algorithme TTP/C

Dans [BM02], nous avons effectué une vérification automatique d’un protocole utilisé dans le domaine de l’industrie du transport. Le protocole, appelé TTP/C, permet la communication entre plusieurs composants (stations) embarqués dans une voiture automobile, et a pour but d’assurer l’exécution sûre de différentes actions de conduite. Le TTP/C comprend un algorithme qui gère l’appartenance des stations au groupe des stations actives. Cet algorithme permet aux stations de se rendre compte qu’elles sont défailtantes (qu’elles sont perçues comme défailtantes par les autres) et de se retirer du groupe. L’algorithme permet aussi aux stations inactives de rejoindre le groupe.

La propriété essentielle que doit satisfaire l’algorithme est qu’après toute occurrence d’une faute (défaillance d’une station), il y a stabilisation dans une configuration où il n’existe qu’un seul groupe de stations actives communiquant entre elles (il n’y a pas formation de plusieurs cliques).

Nous avons montré qu’il est possible de vérifier cette propriété automatiquement pour un nombre arbitraire de stations et un nombre arbitraire de fautes.

2.3.2 Analyse du protocole PGM avec TReX

Le PGM est un protocole multicast qui a été conçu par le IETF. C’est un protocole de la couche transport (au dessus de IP) permettant la transmission rapide de données d’une source vers plusieurs (un grand nombre de) destinataires. La transmission se fait à travers un réseau de noeuds intermédiaires reliés par des canaux non fiables selon une architecture arborescente. Le protocole ne doit pas être complètement fiable mais il doit assurer le fait que pour chaque message émis par la source, chaque destinataire reçoit soit le message (correctement), soit une indication de perte définitive (le message n’a pas pu être récupéré).

Dans [BS02], nous considérons le problème de la vérification paramétrique du protocole : une relation entre les différents paramètres du protocole est dérivée manuellement puis vérifiée automatiquement en utilisant des outils de model-checking. Cette relation exprime les contraintes sur les paramètres (taille des fenêtres d’émission et de réception, le délai de propagation dans le réseau, la vitesse d’émission de la source, etc.) sous lesquelles le protocole est complètement fiable (dans le sens où aucun message ne sera perdu définitivement).

2.3.3 Analyse du protocole PGM avec UPPAAL

Le protocole PGM (Pragmatic General Multicast) fait partie de la seconde génération de protocoles multicast, comportant des contraintes pour la réception des messages. Pour éviter la surcharge du réseau par des messages d’acquittement positifs, ceux-ci sont supprimés. Lorsqu’une perte est détectée, des acquittement négatifs sont émis, suivis par la retransmission des paquets perdus.

La propriété que doit satisfaire le protocole est la suivante : la perte définitive d'un paquet doit être détectée.

Pour vérifier cette propriété, nous avons proposé dans [BBP02] une modélisation simplifiée du protocole, utilisant le formalisme du model-checker UPPAAL. Cette modélisation incorpore les aspects temporisés du protocole, en particulier les constantes relatives aux fréquences d'émission de paquets et aux délais de détection des pertes. Ces paramètres s'avèrent cruciaux pour la satisfaction de la propriété : l'analyse avec UPPAAL fait apparaître des cas d'échec, dus notamment aux relations entre leurs différentes valeurs.

2.4 Développement d'outils

2.4.1 TReX

TReX (*Tool for the REachability analysis of compleX systems*, [ABS01]) permet d'analyser automatiquement des modèles basés sur des automates équipés de différentes sortes de variables et de structures de données non bornées. Ces modèles sont des automates temporisés paramétrés, étendus par des compteurs entiers, et communiquant par des canaux FIFO (non fiables).

L'outil TReX admet des modèles décrits dans le format IF (développé à Vérimag), ce qui permet d'utiliser SDL comme langage de spécification (grâce à la connexion SDL-IF de l'environnement IF).

TReX permet d'effectuer les opérations suivantes :

- Construction d'une représentation de l'ensemble des configurations accessibles à partir d'un ensemble de configurations initiales,
- Vérification à la volée de propriétés de sûreté,
- Construction d'un graphe symbolique, qui est par définition une abstraction finie du système analysé. Ce graphe symbolique peut être minimisé, visualisé, et utilisé en model-checking (en utilisant les outils de l'environnement IF).

Les algorithmes implémentés dans l'outil TReX sont basés sur l'analyse d'accessibilité. Ils utilisent (1) des structures de représentation finies d'ensembles infinis de configurations, et (2) des procédures d'exploration des espaces de configurations dont la terminaison est aidée (ou forcée) par des techniques d'accélération (de calcul de point fixe) [ABJ98, AAB99, AAB00].

TReX utilise plusieurs bibliothèques de manipulation de contraintes sur les configurations. En particulier, TReX utilise les P-DBM (Parametric Difference Bound Matrices) qui permettent d'analyser des automates à horloges et à compteurs paramétrés.

L'outil TReX a été utilisé pour analyser des systèmes avec des contraintes paramétriques complexes (par exemple le protocole BRP [AAB99, AAB00] et le *IEEE 1394 Root Contention Protocol* [CAS01]).

La version actuelle de TReX (version 1.3) est disponible à l'adresse : <http://www.liafa.jussieu.fr/~sighirea>

3 Activités d'animation

3.1 Journées « Vérification des Propriétés Quantitatives »

Dans le cadre de l'Action, 4 journées scientifiques ont été organisées :

- le 18 décembre 2001, à l'ENS Cachan
- le 28 mars 2002, à l'ENS Cachan
- le 27 juin 2002, au LIAFA
- le 13 novembre 2002, à l'ENS Cachan

Chacune de ces journées a été annoncée par les *mailing-lists* habituelles et a réuni une assistance variant de 20 à 40 personnes (venues de toute la France et parfois de l'étranger). Le programme de ces journées est donné en Annexe A. En général, on pourra trouver, via les pages web de l'Action, une sorte de *proceedings online* (des pointeurs sur des transparents ou sur des papiers, en tout cas un résumé) correspondant aux exposés.

3.2 Workshop à Grenoble en mars 2003

Par ailleurs, l'Action va organiser un workshop à Grenoble, les 5, 6 et 7 mars 2003, conjointement avec les "Journées Systèmes Infinis" (un groupe de travail du GDR ALP qui partage une partie de nos centres d'intérêt).

Références

- [AAB99] P. A. Abdulla, A. Annichini, and A. Bouajjani. Symbolic verification of lossy channel systems : Application to the bounded retransmission protocol. In *Proc. 5th Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'99), Amsterdam, The Netherlands, Mar. 1999*, volume 1579 of *Lecture Notes in Computer Science*, pages 208–222. Springer, 1999.
- [AAB00] A. Annichini, E. Asarin, and A. Bouajjani. Symbolic techniques for parametric reasoning about counter and clock systems. In *Proc. 12th Int. Conf. Computer Aided Verification (CAV'2000), Chicago, IL, USA, July 2000*, volume 1855 of *Lecture Notes in Computer Science*, pages 419–434. Springer, 2000.
- [ABJ98] P. A. Abdulla, A. Bouajjani, and B. Jonsson. On-the-fly analysis of systems with unbounded, lossy FIFO channels. In *Proc. 10th Int. Conf. Computer Aided Verification (CAV'98), Vancouver, BC, Canada, June-July 1998*, volume 1427 of *Lecture Notes in Computer Science*, pages 305–318. Springer, 1998.
- [ABS01] A. Annichini, A. Bouajjani, and M. Sighireanu. TReX : A tool for reachability analysis of complex systems. In *Proc. 13th Int. Conf. Computer Aided Verification (CAV'2001), Paris, France, July 2001*, volume 2102 of *Lecture Notes in Computer Science*, pages 368–372. Springer, 2001.
- [APSY02] E. Asarin, G. J. Pace, G. Schneider, and S. Yovine. SPeeDI – a verification tool for polygonal hybrid systems. In *Proc. 14th Int. Conf. Computer Aided Verification (CAV'2002), Copenhagen, Denmark, July 2002*, volume 2404 of *Lecture Notes in Computer Science*, pages 354–358. Springer, 2002.
- [AS02] E. Asarin and G. Schneider. Widening the boundary between decidable and undecidable hybrid systems. In *Proc. 13th Int. Conf. Concurrency Theory (CONCUR'2002), Brno, Czech Republic, Aug. 2002*, volume 2421 of *Lecture Notes in Computer Science*, pages 193–208. Springer, 2002.
- [ASY01] E. Asarin, G. Schneider, and S. Yovine. On the decidability of the reachability problem for planar differential inclusions. In *Proc. 4th Int. Workshop Hybrid Systems : Computation and Control (HSCC'2001), Roma, Italy, Mar. 2001*, volume 2034 of *Lecture Notes in Computer Science*, pages 89–104. Springer, 2001.
- [ASY02] E. Asarin, G. Schneider, and S. Yovine. Towards computing phase portraits of polygonal differential inclusions. In *Proc. 5th Int. Workshop Hybrid Systems : Computation and Control (HSCC'2002), Stanford, CA, USA, Mar. 2002*, volume 2034 of *Lecture Notes in Computer Science*, pages 49–61. Springer, 2002.
- [BBP02] B. Bérard, P. Bouyer, and A. Petit. Analysing the PGM protocol with UP-PAAL. In *Proc. 2nd Workshop on Real-Time Tools (RT-TOOLS'02), Copenhagen, Denmark, Aug. 2002*, 2002. 12 pages. Proceedings published as Tech. Report 2002-025, Dept. Information Technology, Uppsala Univ., Sweden.

- [BET03] A. Bouajjani, J. Esparza, and T. Touili. A generic approach to the static analysis of concurrent programs with procedures. In *Proc. 30th ACM Symp. Principles of Programming Languages (POPL'2003)*, New Orleans, LA, USA, Jan. 2003. ACM Press, 2003.
- [BF99] B. Bérard and L. Fribourg. Reachability analysis of (timed) Petri nets using real arithmetic. In *Proc. 10th Int. Conf. Concurrency Theory (CONCUR'99)*, Eindhoven, The Netherlands, Aug. 1999, volume 1664 of *Lecture Notes in Computer Science*, pages 178–193. Springer, 1999.
- [BH99] Ahmed Bouajjani and Peter Habermehl. Symbolic Reachability Analysis of Fifo-Channel Systems with Nonregular Sets of Configurations. *Theoretical Computer Science*, 221(1-2) :211–250, 1999.
- [BHM01] A. Bouajjani, P. Habermehl, and R. Mayr. Automatic verification of recursive procedures with one integer parameter. In *Proc. 26th Int. Symp. Math. Found. Comp. Sci. (MFCS'2001)*, Mariánské Lázně, Czech Republic, Aug. 2001, volume 2136 of *Lecture Notes in Computer Science*, pages 198–211. Springer, 2001.
- [BM02] A. Bouajjani and A. Merceron. Parametric verification of a group membership algorithm. In *Proc. 7th Int. Symp. Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'2002)*, Oldenburg, Germany, Sep. 2002, volume 2469 of *Lecture Notes in Computer Science*, pages 311–330. Springer, 2002.
- [BS02] Marc Boyer and Mihaela Sighireanu. Synthesis and Verification of Constraints in the PGM protocol. Technical report, LIAFA - University of Paris 7, October 2002.
- [CAS01] A. Collomb-Annichini and M. Sighireanu. Parameterized reachability analysis of the IEEE 1394 Root Contention Protocol using TReX. In *Proc. 1st Workshop on Real-Time Tools (RT-TOOLS'01)*, Aalborg, Denmark, Aug. 2001. Aalborg University, 2001.
- [CJ99] H. Comon and Y. Jurski. Timed automata and the theory of real numbers. In *Proc. 10th Int. Conf. Concurrency Theory (CONCUR'99)*, Eindhoven, The Netherlands, Aug. 1999, volume 1664 of *Lecture Notes in Computer Science*, pages 242–257. Springer, 1999.
- [Gar02] D. Garriou. étude et mise en œuvre de la simulation symbolique de programmes réactifs synchrones. Thèse de Doctorat, École Centrale de Nantes, September 2002.
- [HL02] Peter Habermehl and Sébastien Leveque. Automates à contraintes et automates d'Ibarra. Technical report, LIAFA - University of Paris 7, September 2002.
- [Sch02] G. Schneider. Algorithmic analysis of polygonal hybrid systems. Thèse de Doctorat, Université Joseph Fourier, Grenoble, July 2002.

A Bilan Financier

Le LIAFA ne figurait pas parmi les participants à l'origine de l'Action. Comme pour G. Sutre et B. Jeannet, les dépenses des participants du LIAFA (des missions uniquement) ont été prises en charge par Vérimag et le LSV.

Partenaire	Dotation	Missions	Fonct.	Equip.	Reste
Vérimag ¹	16 007	15 056	488	0	463
IRCCYN (Mauras)	2 202	2 202	0	0	0
IRCCYN (Roux)	5 182	5 182	0	0	0
LSV	21 952	14 083	2 589	5 001	279

B Programmes des journées VPQ

B.1 Journée du 18 décembre 2001 à Cachan

- Application de l'analyse de relation linéaires à la vérification de programmes réactifs. N. Halbwachs (VÉRIMAG, Grenoble)
- Vérification des propriétés quantitatives des automates temporisés. Y. Jurski (LIAFA, Paris)
- Analyse des systèmes hybrides à pente paramétrique. M. Adélaïde (IRCYNN, Nantes)
- Arithmétique réelle et problème de l'accessibilité dans les réseaux de Petri temporisés. B. Bérard (LSV, Cachan)
- L'outil OpenKronos. S. Yovine (VÉRIMAG, Grenoble)

B.2 Journée du 28 mars 2002 à Cachan

- BDD interprétés pour la simulation symbolique et la découverte de propriétés numériques de programmes synchrones. Christophe Mauras (IRCCYN, Nantes)
- Accélérer permet souvent de terminer!. Alain Finkel (LSV, Cachan)
- Interprétation abstraite de types de données hétérogènes : partitionnement dynamique et transformateurs de prédicats approchés. Bertrand Jeannet (IRISA, Rennes)
- Des exemples de preuves de théorèmes dans le calcul des durées par analyse de relations linéaires. Nicolas Halbwachs (VÉRIMAG, Grenoble)
- TAXYS = Esterel + Kronos. Sergio Yovine (VÉRIMAG, Grenoble)
- Vérification de procédures récursives avec paramètres entiers. Ahmed Bouajjani (LIAFA, Paris)
- Vérification de propriétés quantitatives de structures de Kripke avec durées. François Laroussinie (LSV, Cachan)

¹7 000 Euros, provenant d'autres crédits, ont été réservés pour financer le colloque "Propriétés quantitatives et systèmes infinis" des 5-7 mars 2003.

B.3 Journée du 27 juin 2002 à Paris

- Presburger Model Checker. Mamoun Filali (IRIT, Toulouse)
- Parametric Verification of a Membership Group Algorithm. Agathe Merceron (LIAFA, Paris 7)
- Résolution de requêtes temporelles. Philippe Schnoebelen (LSV, Cachan)
- Vérification du protocole PGM : une expérience avec Uppaal. Béatrice Bérard (LSV, Cachan)
- PGM : étude avec le IF-toolset. Marc Boyer (LIAFA, Paris 7)
- Fault Diagnosis for Timed Automata. Stavros Tripakis (VÉRIMAG, Grenoble)
- Vérification et analyse qualitative de systèmes hybrides : dimension 2. Eugene Asarin (VÉRIMAG, Grenoble)

B.4 Journée du 13 novembre 2002 à Cachan

- Vérification probabiliste des systèmes non-fiables. Philippe Schnoebelen (LSV, Cachan)
- Déterminisation d'automates à contraintes et comparaison avec les automates d'Ibarra. Peter Habermehl (LIAFA, Paris 7)
- Vérification probabiliste de LTL. Jean-Michel Couvreur (LSV, Cachan & LaBRI, Bordeaux)
- Vérification de programmes récursifs parallèles. Tayssir Touili (LIAFA, Paris 7)
- Speedup Prediction for Selective Compilation of Embedded Java Programs. Sergio Yovine (VÉRIMAG, Grenoble)
- Untameable Timed Automata!. Patricia Bouyer (LSV, Cachan)