Vérification de systèmes communicant à l'aide de canaux non-bornés

Régis Gascon
INRIA Sophia Antipolis
Email: regis.gascon@sophia.inria.fr

Éric Madelaine
INRIA Sophia Antipolis
Email: eric.madelaine@sophia.inria.fr

Vivien Maisonneuve
ENS Cachan
Email: v.maisonneuve@gmail.com

***Résumé*—Nous considérons le problème du calcul des configurations accessibles dans des systèmes de machines communicant à l'aide de canaux FIFO non-bornés. Dans ce but, nous avons implanté un semi-algorithme permettant de manipuler des ensembles infinis de configurations en représentant le contenu des différents canaux de communication à l'aide d'automates finis. Nous utilisons aussi certaines techniques d'accélérations afin d'améliorer ce calcul. Cette opération permet de calculer l'effet d'un nombre quelconque d'itérations d'une boucle dans le système. L'objectif est d'intégrer ce type de méthode dans une plateforme pour la spécification et la vérification formelle de systèmes distribués.**

***Mots clés :* Systèmes infinis, vérification formelle, machines finies communicantes, accessibilité, semi-algorithme, accélération.**

I. Contexte général

La modélisation mathématique de systèmes distribués nécessite parfois l'introduction de paramètres non bornés. Ces paramètres dépendent par exemple des données manipulées (horloges, compteurs,...) ou de la communication à travers l'utilisation de canaux pour stocker les messages. La vérification formelle de propriétés comportementales sur ces systèmes est alors compliquée car il est nécessaire de pouvoir représenter et manipuler des ensembles d'états potentiellement infinis. Plusieurs techniques permettant de palier à ce problème ont cependant été introduites ces dernières années.

La plateforme Vercors [5] permet de générer à partir de descriptions haut niveau (graphique) de systèmes distribués des modèles formels sur lesquels il est possible d'utiliser des moteurs de vérification automatique. Actuellement, le moteur utilisé impose de définir une abstraction finie des domaines des paramètres non bornés du système. Nous voulons utiliser des méthodes de vérification formelle pour les systèmes infinis afin de se passer des abstractions finies des paramètres non-bornés et ainsi démontrer des propriétés plus précises sur le système original. De plus, il est intéressant de pouvoir comparer en pratique l'efficacité de ces deux approches lorsqu'elles sont toutes deux applicables.

II. Présentation du problème

Notre objectif est d'appliquer sur les modèles intermédiaires générés par Vercors des techniques permettant de gérer des queues de requêtes non bornées. Nous nous concentrons ici sur l'aspect infini des canaux en considérant une topologie du système non-paramétrée et un alphabet fini pour les communications ainsi que pour les actions internes. Formellement, nous considérons des systèmes composés de machines à états finis de la forme

$$\mathcal{M} = (Q, q_0, C, \Sigma, A, \delta)$$

telles que :

- Q est un ensemble fini d'états,
- $q_0 \in Q$ est l'état initial de la machine,
- C est un ensemble de canaux de communication,
- Σ est un ensemble fini de caractères qui définit l'alphabet des communications,
- A est un ensemble fini d'actions internes,
- la relation de transition δ est un sous ensemble de $Q \times ((C \times \{?, !\} \times \Sigma) \cup A) \times Q$.

Par exemple, l'*Alternating Bit Protocol* (ABP) peut se modéliser avec deux machines communicantes $\mathcal{M}_1 = (Q^1, q_0^1, C^1, \Sigma^1, A^1, \delta^1)$ et $\mathcal{M}_2 = (Q^2, q_0^2, C^2, \Sigma^2, A^2, \delta^2)$ représentant respectivement le système émetteur et le système récepteur de la manière présentée dans la Fig. 1. L'action E désigne l'opération d'envoi du message et R l'opération de réception : il s'agit de transitions internes. De plus, nous supposons que les canaux sont partagés par les deux machines ($C^1 = C^2 = \{K, L\}$). Une configuration pour ce système

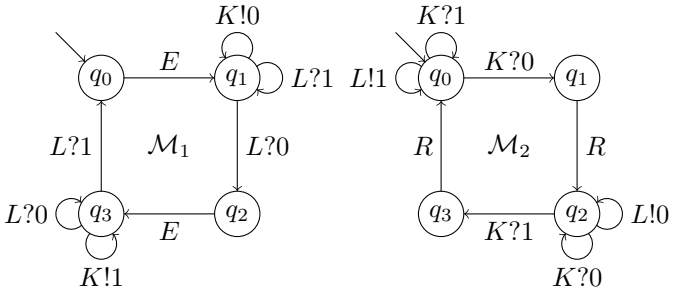

FIG. 1. Modélisation de l'Alternating Bit Protocol

est un élément de $(Q^1 \times Q^2) \times (\Sigma^1 \cup \Sigma^2)^{|C|}$ qui détermine un état pour chacune des machines ainsi que le contenu des canaux. Nous considérons des canaux de types FIFO non-bornés. Ainsi, une opération d'envoi sur un canal telle que $\langle K, !, 0\rangle$ ($K!0$ sur la Fig. 1) ajoute le symbole 0 à la fin du canal K et une transition étiquetée par $\langle K, ?, 0\rangle$ (aussi notée $K?0$) ne peut être franchie que si le symbole en tête de la

file K est 0. Formellement, la sémantique est définie de la manière suivante :

- la configuration initiale est $\langle q_0, q_0, \varepsilon, \varepsilon\rangle$ où ε représente un canal vide.
- une transition $\langle q_a, q_b, K, L\rangle \xrightarrow{op} \langle q'_a, q'_b, K', L'\rangle$ est possible ssi soit $\langle q_a, op, q'_a\rangle \in \delta_1$ et $q_b = q'_b$ ou bien $\langle q_b, op, q'_b\rangle \in \delta_2$ et $q_a = q'_a$, et l'une des conditions suivantes est vérifiée :
  - si $op \in A$ alors $K' = K$ et $L' = L$,
  - si op est de la forme $K!i$ (resp. $L!i$) alors $K' = K \cdot i$ et $L' = L$ (resp. $K' = K$ et $L' = L \cdot i$),
  - si op est de la forme $K?i$ (resp. $L?i$) alors $i \cdot K' = K$ et $L' = L$ (resp. $K' = K$ et $i \cdot L' = L$).

Cette sémantique se généralise facilement à un nombre quelconque de machines communicantes et de canaux. Un exemple d'exécution l'ABP est le suivant :

$$(q_0, q_0, \varepsilon, \varepsilon) \xrightarrow{E} (q_1, q_0, \varepsilon, \varepsilon) \xrightarrow{K!0} (q_1, q_0, 0, \varepsilon)$$
$$\xrightarrow{K?0} (q_1, q_1, \varepsilon, \varepsilon) \xrightarrow{R} (q_1, q_2, \varepsilon, \varepsilon) \xrightarrow{L!0} (q_1, q_2, \varepsilon, L) \cdots$$

III. Contribution

Nous nous intéressons au problème du calcul des configurations accessibles dans un tel système de machines communicantes. Ce problème permet de certifier des propriétés de sûreté simples telles que l'accessibilité d'un bon état du système (ou au contraire l'évitabilité d'un mauvais état). En théorie, ce problème est indécidable puisque une machine utilisant des canaux FIFO non-bornés est équivalente à une machine de Turing. Nous utilisons donc un semi-algorithme permettant de manipuler des ensembles de configurations potentiellement infinis. Pour cela, nous avons choisi d'adapter certaines techniques utilisant une représentation des canaux sous la forme d'automates finis introduite dans [1] et [2].

La convergence du semi-algorithme est améliorée par l'utilisation de techniques d'accélération permettant de calculer directement l'itération d'une séquence d'opérations (boucle). Par exemple, l'itération de la boucle $q_1 \xrightarrow{K!0} q_1$ de l'émetteur de l'ABP sur une configuration $\langle q_1, q', K, L\rangle$ produit l'ensemble de configurations $\langle q_1, q', K \cdot 0^*, L\rangle$. Cet ensemble peut facilement être représenté en codant le contenu du canal K par un automate fini. Il est néanmoins nécessaire de sélectionner les boucles à accélérer. En effet, l'itération de certaines boucles produit un ensemble de configurations dont le langage induit par le contenu des canaux n'est pas régulier. Par conséquent, un tel ensemble de configurations ne peut être représenté à l'aide d'automates finis. Une perspective consiste d'ailleurs à étendre la représentation des canaux afin de tenir compte de ce type de comportement en utilisant par exemple les idées de [3]. Dans notre exemple, nous obtenons l'ensemble des configurations accessibles donné dans la Fig. 2. Sur cet exemple simple, une analyse du résultat permet de voir que les messages arrivent dans l'ordre souhaité : par exemple, aucune configuration où K est de la forme $0^*1^+0^*$ n'est accessible.

Nous avons implanté cette approche en utilisant l'algorithme décrit dans [4]. Ce prototype est programmé en langage Java et a pour objectif d'être intégré dans la plateforme Vercors.

$Q^1 \times Q^2$	K	L
$\langle q_0, q_0\rangle$	1^*	1^*
$\langle q_1, q_0\rangle$	$1^* \cdot 0^*$	1^*
$\langle q_1, q_1\rangle$	0^*	1^*
$\langle q_1, q_2\rangle$	1^*	$1^* \cdot 0^*$
$\langle q_2, q_2\rangle$	0^*	0^*
$\langle q_3, q_0\rangle$	1^*	$0^* \cdot 1^*$
$\langle q_3, q_2\rangle$	$0^* \cdot 1^*$	0^*
$\langle q_3, q_3\rangle$	1^*	0^*

FIG. 2. Configurations accessibles dans l'ABP

Nous utilisons aussi certaines heuristiques pour la sélection des boucles à accélérer lors du calcul. Il est envisagé par la suite d'étendre ce formalisme et ces heuristiques selon les besoins liés à notre plateforme. Une autre perspective concerne l'extension de cette approche à la vérification de propriétés de fiabilité plus riches que la simple accessibilité d'une configuration. Dans certains cas, il est en effet possible de calculer l'ensemble des configurations à partir desquelles une boucle peut être itérée infiniment souvent. L'ajout de ce type de méthodes ouvrirait des perspectives de vérification de propriétés exprimables par des automates de Büchi (i.e. plusieurs logiques temporelles).

IV. Plan de l'exposé

Nous introduirons dans un premier temps le formalisme utilisé ainsi que les principes de base de notre algorithme. En particulier, nous développerons plus en détail la représentation des canaux à l'aide d'automates finis ainsi que les opérations élémentaires permettant de modifier la représentation en fonction de l'effet des transitions. Nous parlerons ensuite de l'accélération et du calcul de leur effet sur la représentation d'un canal. Nous évoquerons le problème de la sélection des boucles qui peuvent être accélérées ainsi que les limites de cette technique dans l'approche utilisée actuellement. Enfin, nous présenterons l'implantation de notre prototype ainsi que les premiers résultats expérimentaux. Nous conclurons en présentant plus précisément les différentes perspectives d'extensions pour notre prototype : par exemple l'extension de la représentation des canaux ou la possibilité de vérifier des problèmes plus riches que la simple accessibilité.

Références

[1] B. Boigelot and P. Godefroid. Symbolic verification of communication protocols with infinite state spaces using QDDs (extended abstract). In *CAV*, vol. 1102 of *LNCS*, pp. 1–12. Springer, 1996.

[2] B. Boigelot, P. Godefroid, B. Willems, and P. Wolper. The power of QDDs (extended abstract). In *SAS*, vol. 1302 of *LNCS*, pp. 172–186. Springer, 1997.

[3] A. Bouajjani and P. Habermehl. Symbolic reachability analysis of FIFO channel systems with nonregular sets of configurations (extended abstract). In *ICALP*, vol. 1256 of *LNCS*, pp. 560–570. Springer, 1997.

[4] S. Roy. Symbolic verification of infinite systems using a finite union of DFAs. In *SEFM*, pp. 56–66. IEEE Computer Society, 2004.

[5] The VERCORS platform : VERification of models for distributed communicating COmponants, with safety and Security. `http ://www-sop.inria.fr/oasis/index.php ?page=vercors`.