Proposition de recherche doctorale

Exploitation des syme?tries pour le model checking de syste?mes re?partis : du mode?le au codage

Mots clés :

- **Directeur de thèse** : Fabrice Kordon
- **Co-encadrant(s)** :
- **Unité de recherche** : Laboratoire d'informatique de Paris 6
- **Ecole doctorale** : École Doctorale Informatique, Télécommunications, Électronique de Paris
- **Domaine scientifique principal**: Divers

Résumé du projet de recherche (Langue 1)

{{{Equipe d'accueil}}} La the?se se de?roule au sein de l'e?quipe MoVe (Mode?lisation et Ve?rification) au sein du de?partement "Re?seaux et syste?mes re?partis" du laboratoire LIP6. Cette e?quipe se compose actuellement de 22 enseignants-chercheurs permanents (dont 9 habilite?s a? diriger des recherches) et 10 doctorants. L'e?quipe MoVe s'inte?resse a? la mode?lisation et a? l'analyse des syste?mes re?partis a? composants interope?rables. Son objectif est de de?velopper des techniques de mode?lisation, des outils de ve?rification et des ge?ne?rateurs de programmes permettant a? des inge?nieurs de de?velopper sans risque des applications re?parties complexes. Les tech- niques et outils de?veloppe?s se placent dans le contexte d'une approche MDD (Model Driven Development). {{{1 Introduction}}} L'e?volution des technologies de communication a accru l'utilisation de syste?mes re?partis dans de nombreux secteurs d'activite? : industries, administrations, applications grand public... En particulier, les domaines critiques sont de plus en plus concerne?s : transports, applications me?dicales ({life critical}), militaires, spatiales ({mission critical}), mais aussi commerciales ou industrielles ({business critical}). La criticite? de ces domaines ne?cessite de garantir le fonctionnement des syste?mes en question, et renforce l'inte?re?t d'une e?tape de ve?rification formelle dans le cycle de de?veloppement des produits. La mode?lisation et la ve?rification constituent d'une part une assistance a? la conception, par le formalisme qu'elles offrent, et d'autre part permettent un haut degre? de fiabilite? des syste?mes de?veloppe?s. Les syste?mes re?partis, dont l'utilisation se re?pand tout en reque?rant de plus en plus de fiabilite?, sont une cible de choix pour ces techniques. Pour toutes ces raisons, la demande d'outils de mode?lisation et de ve?rification est aujourd'hui en forte croissance. Cette demande et le manque de maturite? des outils actuels font de la ve?rification un champ de recherches actif. {{{2 Contexte scientifique}}} Le model checking est l'une des principales approches de ve?rification. Il repose sur l'e?laboration d'un mode?le du syste?me a? ve?rifier, sur lequel sont e?nonce?es les proprie?te?s a? prouver (a? l'aide d'une logique approprie?e). Ces proprie?te?s sont ve?rifie?es par l'exploration exhaustive des e?volutions possibles du mode?le (l'espace d'e?tats). La ge?ne?ration et le parcours de l'espace d'e?tats sont facilement automatisables, mais les algorithmes nai?fs sont victimes de l'explosion combinatoire du nombre d'e?tats de?s que les mode?les deviennent re?alistes. La mise en œuvre pratique de l'automatisation requiert donc l'utilisation de diffe?rentes techniques d'optimisation. Deux axes sont a? conside?rer : -* l'exploitation des syme?tries du syste?me([1]),pour construire des e?tats symboliques repre?sentant des familles d'e?tats concrets aux proprie?te?s communes. Cette notion est a? l'origine de la the?orie des Symmetric Nets (SN), variante des Coloured Petri Nets ([2],[3]) -* l'utilisation de techniques de codage et de compression, gra?ce a? l'emploi de diagrammes de de?cision notamment, pour optimiser la charge de la me?moire ([4]). Ces deux techniques tre?s diffe?rentes peuvent e?tre conside?re?es de manie?re inde?pendante. Elles n'en sont pas moins comple?mentaires, et particulie?rement inte?ressantes dans la perspective d'une imple?mentation. La premie?re suppose une re?flexion au niveau des formalimes utilise?s, pour repe?rer les syme?tries du syste?me, qui peuvent aboutir a? des redondances d'information. Les e?tats syme?triques du syste?me sont regroupe?s au sein de classes d'e?quivalence d'e?tats, re?duisant ainsi l'espace d'e?tats a? explorer. La seconde approche est essentiellement algorithmique, et repose sur l'optimisation du codage et de l'utilisation de la me?moire. Elle s'appuie sur l'utilisation de diagrammes de de?cision pour coder les e?tats. Le travail de the?se s'inse?re e?galement dans les efforts de l'e?quipe MoVe pour combattre le proble?me de l'explosion combinatoire. Des travaux ont de?ja? e?te? mene?s ces dernie?res anne?es au sein de l'e?quipe. Certains concernent l'exploitation des syme?tries : de?tection des syme?tries, syme?tries structurelles, syme?tries globales et locales... D'autres se sont concentre?s sur l'algorithmique ne?cessaire a? l'exploration d'un espace d'e?tats : e?quivalence et traduction entre diffe?rents formalismes (colore?s - P/T), codages exploitant les diffe?rents niveaux de syme?trie, codages hie?rarchiques... MoVe se distingue notamment par deux ide?es originales : -* l'e?tude des syme?tries partielles et locales ([5],[6],[7]) -* les diagrammes de de?cision hie?rarchiques ([8]). L'e?quipe inte?gre ses outils de mode?lisation et ve?rification des re?seaux de Petri au sein de sa plateforme FrameKit. Ce dispositif oriente l'e?quipe vers un souci de faisablilite? : les nouveaux concepts conduisent a? la re?alisation de prototypes qui peuvent e?tre confronte?s a? des e?tudes de cas. {{{3 Proble?matique}}} La the?se vise a? approfondir les travaux re?sume?s par [9] mene?s au sein de l'e?quipe : la ge?ne?ration de l'espace d'e?tats d'un SN en utilisant un codage en Data Decision Diagrams (DDD). Ceux-ci e?tendent les Binary Decision Diagrams (BDD) en autorisant la valuation des arcs avec des entiers. Ils introduisent e?galement les homomorphismes, outils de manipulation souples et puissants ([10]). Plus pre?cise?ment, le travail se focalise sur deux principaux axes : -* e?tablir un cadre ge?ne?ral pour l'exploitation des syme?tries structurelles d'un mode?le. Cette re?ification doit permettre de dissocier la syme?trie d'un formalisme particulier. -* approfondir les techniques de codage et d'optimisation de la me?moire, notamment en attachant un codage a? chaque niveau de syme?trie identifie? dans le processus de re?ification. Sont donc conduites en paralle?le la re?flexion sur les mode?les et celle sur leurs codages. Ainsi, l'optimisation de l'utilisation me?moire est affine?e : les codages aux diffe?rents niveaux de syme?trie peuvent e?voluer inde?pendamment, pour exploiter au mieux les diffe?rents types de contraintes.

{{L’exploitation des syme?tries structurelles}} Le processus de re?ification vise en premier lieu l’identification des caracte?ristiques de la syme?trie d’un mode?le inde?pendamment du formalisme. Il doit donc permettre de valider simplement le caracte?re syme?trique de nouveaux formalismes. Mais il doit aller plus loin : identifier diffe?rents niveaux de syme?tries, correspondant a? des classes de formalismes, chaque niveau e?tant caracte?rise? inde?pendamment des formalismes. {{Optimisation de l’utilisation de la me?moire}} L’inte?re?t des syme?tries est l’impact qu’elles ont sur la repre?sentation en me?moire. La syme?trie induit en effet une contrainte qu’il n’est pas ne?cessaire de reproduire dans le codage. L’identification de diffe?rents niveaux de syme?tries pour les mode?les va donc naturellement de pair avec la de?finition d’un codage pour chaque niveau, exploitant au mieux les particularite?s du niveau en question. On aboutit donc a? un codage variable, de?pendant du degre? de syme?trie du mode?le. On envisage d’aller plus loin, en utilisant un codage dynamique : pour un me?me mode?le, le codage de?pend de la position dans l’espace d’e?tat. Les mode?les peuvent en effet pre?senter localement des syme?tries plus ou moins fortes. L’ide?e est de ne pas niveler la syme?trie globale par le bas. La mise en place de codages dynamiques peut e?tre facilite?e par l’utilisation de PolyDD, diagrammes de de?cision polymorphes ([11]), afin de garantir la cohe?rence d’un codage me?me dynamique. Enfin, la re?flexion en paralle?le sur les formalismes et les codages doit permettre de mieux comprendre les interactions entre les deux techniques. Elle mettra en lumie?re d’e?ventuels effets symbiotiques, positifs ou ne?gatifs, pour une meilleure compre?hension des relations entre formalismes et codages. {{{4 De?marche}}} Le travail de the?se s’appuie sur le formalisme des Symmetric Nets with Bags (SNB), variante des SN dans laquelle les pt pas la syme?trie, et ne bride pas les performances de la ve?rification. Le travail de the?se de?bute par l’e?tude de ces SNB, en tant que cas particulier devant conduire au processus de re?ification e?voque? plus haut. Cette de?marche aura e?te? initie?e en stage de M2, a? travers l’imple?mentation d’un ge?ne?rateur de l’espace d’e?tats d’un SN, en utilisant des Hierarchical Set Decision Diagrams (SDD) comme base de codage. Ceux-ci, introduits dans [8], sont des diagrammes de de?cision hie?rarchiques, et vont plus loin que les DDD. Ce pre?lude aura permis au candidat de se familiariser avec le sujet, et d’amorcer sa reflexion, sur les codages notamment, et le lien entre syme?tries et repre?sentations. Le travail devrait ensuite suivre les e?tapes suivantes : -* l’identification de diffe?rents types de syme?tries, et des caracte?ristiques d’un formalisme propices a? leur exploitation -* la de?termination de codages pour repre?senter ces syme?tries, de?bouchant sur un codage dynamique des e?tats. A? ce stade, le processus de re?ification est acheve?, et il conviendrait de le synthe?tiser au sein d’un formalisme refle?tant les niveaux de syme?trie identifie?s. Ce formalisme pourra e?tre utilise? pour en re?e?crire d’autres de?ja? existants afin d’en exhiber les syme?tries structurelles, mais aussi pour de?finir de manie?re ge?ne?rique de nouveaux formalisme a? partir d’un degre? de syme?trie souhaite?. Le travail de the?se pourra s’achever par l’imple?mentation d’un outil de ve?rification automatique reprenant ce formalisme et exploitant les codages variables, voire dynamiques, e?voque?s plus haut. La re?flexion sur les syme?tries s’appuiera sans doute sur l’expe?rience de MoVe en matie?re de syme?tries partielles ([6],[7]). De me?me, la de?termination de "bons" codages devra s’appuyer sur les pre?ce?dentes re?alisations ([9]). {{{Re?fe?rences}}}

[1] T. Junttila. {On the Symmetry Reduction Method for Petri Nets and Similar Formalisms}. Research report, Espoo, Finland, September 2003. [2] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. {Stochastic well-formed colored nets and symmetric modeling applications}. IEEE Transactions on Computers, 42(11) :1343–1360, 1993. [3] K. Jensen. {Coloured petri nets}. Petri nets : central models and their properties, pages 248–299, 1992. [4] J.R. Burch, E.M. Clarke, et al. {Symbolic model checking : 10^20 States and beyond}. Information and computation, 98(2) :142–170, 1992. [5] E. Allen Emerson and Richard J. Trefler. {From asymmetry to full symmetry : New techniques for symmetry reduction in model checking}. In In Conference on Correct Hardware Design and Verification Methods, pages 142–156. Springer, 1999. [6] Serge Haddad, Jean-Michel Ilie?, and Khalil Ajami. {A model checking method for partially symmetric systems}. In Proceedings of the FIP TC6 WG6.1 Joint International Conference on Formal Description Techniques for Distributed Systems and Communication Protocols (FORTE XIII) and Protocol Specification, Testing and Verification (PSTV XX), FORTE/PSTV 2000, pages 121–136, Deventer, The Netherlands, The Netherlands, 2000. Kluwer, B.V. [7] Souheib Baarir and Alexandre Duret-Lutz. {Emptiness check of powerset buchi automata using inclusion tests}. In Proceedings of the Seventh International Conference on Application of Concurrency to System Design, pages 41–50, Washington, DC, USA, 2007. IEEE Computer Society. [8] J.M. Couvreur and Y. Thierry-Mieg. {Hierarchical decision diagrams to exploit model structure}. Lecture notes in computer science, 3731 :443, 2005. [9] Y. Thierry-Mieg, J.M. Ilie?, and D. Poitrenaud. {A symbolic symbolic state space representation}. Formal Techniques for Networked and Distributed Systems–FORTE 2004, pages 276–291, 2004. [10] J.M. Couvreur, E. Encrenaz, E. Paviot-Adet, D. Poitrenaud, and P.A. Wacrenier. {Data decision diagrams for Petri net analysis}. Application and Theory of Petri Nets 2002, pages 129–158, 2002. [11] A. Linard, E. Paviot-Adet, F. Kordon, D. Buchs, and S. Charron. {polyDD : Towards a Framework Generalizing Decision Diagrams}. In 10th International Conference on Application of Concurrency to System Design (ACSD’2010), pages 124–133, Braga, Portugal, June 2010. IEEE Computer Society.