Sujet de Thèse
Combiner SOG et Réduction d'Ordre Partiel pour Une Vérification Modulaire et Parallèle de la Logique LTL

Sami Evangelista et Kaïs Klai

LIPN, CNRS UMR 7030
Université Paris 13
99 avenue Jean-Baptiste Clément
F-93430 Villetaneuse, France

{sami.evangelista,kais.klai}@lipn.univ-paris13.fr

1 Contexte scientifique et problématique

1.1 Vérification formelle

La mise au point des applications distribuées critiques est un problème complexe pour lequel il est recommandé d'utiliser des techniques de description formelle afin de spécifier sans ambiguité le comportement des applications considérées. Il faut aussi des outils de vérification automatiques afin de valider le bon fonctionnement de ces applications. L'évolution des systèmes distribués se caractérise par une complexité croissante et un rôle toujours plus critique. La vérification de leurs propriétés est reconnue comme un problème difficile du fait de l'explosion combinatoire de leur espace d'états. Les logiciels orchestrant de tels systèmes doivent réagir correctement et en particulier face aux situations critiques. Les méthodes formelles de spécification de systèmes ont pour objectif d'assurer la fiabilité de ces logiciels, c'est-à-dire leur bonne spécification et l'absence d'erreur. Idéalement, pour concevoir le logiciel d'un système concurrent donné, il faudrait spécifier formellement ce système à l'aide d'un modèle mathématique à partir duquel on pourrait raisonner et vérifier les propriétés attendues. En réalité, ce processus se heurte à plusieurs problèmes d'ordre pratique et théorique. Un premier obstacle apparait au niveau de la définition du langage de spécification utilisé. Si celui-ci est trop expressif, alors on ne peut pas, mathématiquement, l'analyser automatiquement. Les réseaux de Petri sont reconnus pour être suffisamment expressifs pour décrire la réalité des systèmes, tout en restant raisonnablement analysables. Un second obstacle à la vérification formelle des systèmes complexe, en particulier celles basées sur la technique du model checking[CGP99], est l'explosion combinatoire des états possibles des systèmes. Malgré les avancées spectaculaires de la technologie des ordinateurs, il arrive que l'on soit incapable d'analyser intégralement des systèmes par manque d'espace mémoire ou de temps.

Cette thèse s'intègre dans la thématique de la vérification formelle par model checking de systèmes concurrents ou distribués. On s'intéressera en particulier à la vérification de propriétés exprimées avec la logique temporelle linéaire LTL [MP92]. Etant donné que le model checking est basé sur une exploration exhaustive de l'espace d'états accessibles, il est toujours d'actualité de proposer des approches permettant de combattre l'explosion combinatoire de cet espace d'états. Nous nous intéressons à la combinaison de deux approches orthogonales qui ont fait séparément leur preuve dans le domaine de la vérification : Les graphes d'observation [HIK04,KP08b] et l'ordre partiel [GW93].

1.2 Graphes Symboliques d'Observation (SOG)

Une spécification LTL fait rarement référence à tous les aspects d'un système. Souvent elle concerne un sous-ensemble de propositions atomiques parmi toutes celles présentes dans le système. Lors de la vérification d'une propriété LTL, il est possible de réduire la taille de la représentation du système grâce à une observation partielle, guidée par la formule à vérifiée, du système. L'approche des graphes d'observations symboliques est basée sur cette constatation : seules les propositions atomiques apparaissant dans la formule à vérifier sont observées. Le SOG est ainsi défini comme un graphe où les noeuds sont des ensembles d'états explicites impliquant des éléments non observés du système, et dont l'étude n'influence pas la propriété à vérifier. Ils sont représentés et manipulés de manière efficace grace à des structures symboliques comme les diagrammes de décision binaires (type BDD [Bry92]).

Les SOGs permettent donc d'abstraire l'espace d'états d'un système tout en préservant ses propriétés temporelles (en l'occurrence exprimées avec la logique LTL). La technique des SOGs a montré son efficacité dans le cadre de la vérification de systèmes concurrents grâce au gain obtenu, en temps de construction et en consommation mémoire (voir [KP08b,KP08a,DLKPTM11] pour les résultats expérimentaux), par rapport aux approches existentes.

1.3 Technique de Réduction d'Ordre Partiel

Une de sources d'explosion combinatoire de l'espace d'états est l'entrelacement des actions concurrentes. Un système concurrent possédant deux transitions concurrentes, a et b, peut, en principe, les exécuter selon deux ordonnancements différents, ab et ba. Avec n actions concurrentes indépendantes, le nombre d'ordonnancements possibles est $n!$ et le nombre d'états distincts à considérer est $\mathcal{O}(2^n)$, ce qui rend couteux de l'exploration utilisée par un model checker.

La réduction d'ordre partiel [GW93] est une technique largement étudiée pour surmonter ce problème : elle permet de réduire l'espace d'états du système en exploitant l'indépendance entre les évènements/actions concurrentes. Ainsi, un seul représentant de tus les ordonnancement possibles sera représenté dans le graphe et les autres peuvent en être déduits par permutation des actions indépendantes, ce qui se révèle non seulement plus naturel mais aussi plus efficace.

Dans notre exemple, l'ensemble d'ordonnancements $\{ab, ba\}$ serait substitué par un singleton (par exemple $\{ab\}$) .

2 Objectifs de la thèse

Les techniques de vérification basées sur la réduction d'ordre partiel et celles s'appuyant sur les SOGs visent à réduire le coût de la vérification en exploitant deux aspects orthogonaux du système : la concurrence et l'observation partielle de la spécification. Elles ont été étudiées de manière indépendante et l'objectif de cette thèse est de les intégrer au sein d'une seule technique afin de tirer le meilleur de chacune de ces deux approches. En particulier, l'objectif est de

- Approche globale
  - Définir une approche de construction de SOG en appliquant la technique d'ordre partiel. Il s'agira de concevoir des algorithmes symboliques (s'appliquant sur les diagrammes de décision) d'algorithmes d'ordre partiel explicites existants.
  - Envisager l'application de la réduction d'ordre partiel sur des actions observées (notamment dans un context modulaire)
  - Implémenter un Model Checker LTL, basé sur cette nouvelle réduction, dans l'outil Helena [Eva05]. Nous intéresserons à une logique mixte (even and state-based LTL) permettant de considérer des formules impliquant des propositions atomiques d'états et/ou d'actions.
- Approche modulaire/Distribuée
  - Il s'agira d'étendre l'approches précédente à un contexte modulaire permettant la mise en place d'une technique de vérification de haut en bas (décomposition) et de bas en haut (composition) permettant de réduire le cout d'une vérification globale.
  - En se basant sur les travaux récents de l'équipe (voir par exemple [CEP18,OKAZ19]), l'objectif de cette étape et d'intégrer la réduction par ordre partiel dans nos approches de model checking parallel dans le cadre d'un outil de vérification modulaire et parallèle basé sur le SOG.
- Approche parallèle

Références

Bry92. Randal E. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24(3) :293–318, 1992.

CEP18. Camille Coti, Sami Evangelista, and Laure Petrucci. State compression based on one-sided communications for distributed model checking. In *23rd International Conference on Engineering of Complex Computer Systems, ICECCS 2018, Melbourne, Australia, December 12-14, 2018*, pages 41–50, 2018.

CGP99. E. M. Clarke, Orna Grumberg, and Doron Peled. *Model Checking*. MIT Press, 1999.

DLKPTM11. Alexandre Duret-Lutz, Kais Klai, Denis Poitrenaud, and Yann Thierry-Mieg. Self-loop aggregation product - a new hybrid approach to on-the-fly ltl model checking. In *Automated Technology for Verification and Analysis, 9th International Symposium, ATVA 2011, Taipei, Taiwan, October 11-14, 2011. Proceedings*, volume 6996 of *Lecture Notes in Computer Science*, pages 336–350, 2011.

Eva05. Sami Evangelista. High level petri nets analysis with helena. In *Applications and Theory of Petri Nets 2005, 26th International Conference, ICATPN 2005, Miami, USA, June 20-25, 2005, Proceedings*, volume 3536 of *Lecture Notes in Computer Science*, pages 455–464. Springer, 2005.

GW93. Patrice Godefroid and Pierre Wolper. Partial-Order Methods for Temporal Verification. In *CONCUR'1993*, volume 715, pages 233–246, 1993.

HIK04. Serge Haddad, Jean-Michel Ilié, and Kais Klai. Design and evaluation of a symbolic and abstraction-based model checker. In *Automated Technology for Verification and Analysis : Second International Conference - ATVA*, volume 3299 of *Lecture Notes in Computer Science*, pages 196–210. Springer, 2004.

KP08a. Kais Klai and Laure Petrucci. Modular construction of the symbolic observation graph. In *ACSD*, pages 88–97. IEEE, 2008.

KP08b. Kais Klai and Denis Poitrenaud. MC-SOG : An LTL model checker based on symbolic observation graphs. In *Petri Nets*, 2008.

MP92. Zohar Manna and Amir Pnueli. *The temporal logic of reactive and concurrent systems.* Springer-Verlag New York, Inc., New York, NY, USA, 1992.

OKAZ19. Hiba Ouni, Kais Klai, Chiheb Ameur Abid, and Belhassen Zouari. Towards parallel verification of concurrent systems using the symbolic observation graph. In *To appear in 19th International Conference on Application of Concurrency to System Design (ACSD 2019)*, 2019.