ANALYSE ET SOLUTIONS ALGORITHMIQUES DU MODEL CHECKING ET DE LA SYNTHÈSE DE MODÈLES POUR LES MODÈLES PARTIELS

ANALYSIS AND ALGORITHMIC SOLUTIONS OF MODEL CHECKING AND MODEL SYNTHESIS FOR PARTIAL MODELS

Etablissement **Université Paris-Saclay GS Informatique et science du numérique**

École doctorale **Sciences et Technologies de l'Information et de la Communication**

Spécialité **Informatique**

Unité de recherche **IBISC - Informatique, BioInformatique, Systèmes Complexes**

Encadrement de la thèse **Serenella CERRITO (detailResp.pl?resp=36373)**

Début de la thèse le **1 octobre 2021**

Date limite de candidature **30 avril 2021**

Mots clés - Keywords

Information incomplète, Logiques Temporelles,, Logiques multi-agents, Model Checking, Synthèse de modèles, Vérification

Incomplete information, Temporal Logis, Multi-agent logics, Model Checking, Model Synthesis , Verification

Profil et compétences recherchées - Profile and skills required

L'étudiant(e) candidat(e) devra avoir reçu une bonne formation en:
- logique : classique, modale, temporelle
- méthodes de spécification et vérification formelles
- calculabilité et complexité.

Il (elle) doit avoir une très bonne capacité d'abstraction et de raisonnement formel, ainsi qu'une certaine expérience de programmation.

L'étudiant(e) doit être assez à l'aise en Anglais.
The candidate student must have a good training in:
- logic: classical, modal and temporal
- formal specification and verification methods
- calculability and complexity.

He (she) must have a very good capability of abstraction and formal reasoning, and also some experience in programming.

He (she) must be capable to understand and talk English.

Description de la problématique de recherche - Project description

Un problème très important dans le procès de de conception et réalisation d'un système informatique critique est celui de tester automatiquementsi un modèle abstrait donné du système (construit en utilisant une quelque technique formelle, comme des automates ou des structures logiques) vérifie ou pas une quelque propriété P souhaitée pour le système, où P est exprimée dans un quelque langage logique. Ce problème est connu d'habitude comme problème du « model checking ». Dans la majorité des travaux sur le model checking le modèle abstrait M est consideré comme complètement décrit. Toutefois, des nombreux scénarios naturels existent où l'agent (qui peut être un concepteur, or un utilisateur, ou un des acteurs du système) a seulement une information incomplète sur le système M. Ceci peut avoir au moins deux raisons différentes : ou bien le système est encore défini ou construit seulement partiellement, ou bien l'agent a une possibilité d'observation de ce système qui est limitée. Dans les deux cas, on peut supposer que le modèle appartient à une quelque classe de modèles C explicitement définie.

On peut donc considérer deux problèmes différents, bien que connexes, qui se posent dans de tels scénarios où l'information est

incomplète :

1. Compléter la construction partielle de M de façon à obtenir un modèle dans la classe C qui assure la satisfaction de la propriété P, ou bien montrer qu'une telle façon de compléter la construction n'existe pas. Nous appellerons ceci le problème de la « Synthèse de Modèles à partir de Modèles Partiels (Model Synthesis from Partial Models, MSPM). Dans sa version constructive, le problème demande de générer un quelque modèle élément de C qui assure P et qui étend le modèle partiel donné au départ.
2. Vérifier que n'importe quel modèle (appartenant à la classe explicitement définie C) cohérente avec la description partielle possédée par l'agent vérifie la propriété P. Nous appellerons ceci le problème du « Model Checking de Modèles Partiels » (Model Checking of Partial Models, MCPM)

Nous supposons aussi qu'un langage logique L, dans lequel la propriété P est exprimée, ait été fixé. Typiquement, L est une logique modale ou temporelle dont la sémantique est donnée par le biais de systèmes de transitions interprétés, ou d'extensions multi-agents d'une telle logique. Par conséquent, étant donnée L, MCPM et MSPM ont une formulation plus précise comme problèmes logiques de décision :

1. MSPM : étant donnés une formule P de L et une interprétation I de P partiellement décrite, existe-t-il une interprétation I' qui étend I (en un quelque sens précis, par exemple dans une classe donnée C, ou en fonction de L) telle que I' est un modèle de P, c'est-à-dire que P est vraie par rapport à I' ?
2. MCPM : étant donnés une formule P de L et une interprétation I de P partiellement décrite, est-t-il vrai que toutes les interprétations I' qui étendent I (à nouveau, en un quelque sens précis, par exemple dans une classe donnée C) sont des modèles de P, c'est-à-dire qu'elles rendent P vraie ?

Les deux problèmes ont plusieurs applications potentielles à l'IA (et en ingénierie des logiciels et des systèmes) par exemple :

• Le problème MSPM se pose de façon naturelle quand il faut réaliser un système complexe et multi-agents de façon modulaire, et le concepteur de chaque module, ou le superviseur, reçoit une spécification partielle de comment le module composant doit fonctionner et interagir avec le reste du système, mais pas une description exacte de comment le module doit être construit.
• Le problème MSPM se pose de façon naturelle quand le concepteur ou superviseur a une capacité d'observer le système qui est partielle, ou bien une connaissance partielle de celui-ci, et, dans cette condition, il doit assurer que le système se comporte correctement par rapport aux spécifications locales ou globales données.

Il est à observer que les problèmes de décision de la satisfiabilité et le problème du model checking peuvent être vus, respectivement, comme des cas limites de MSPM et MCPM.

Les deux problèmes s'étendent de façon naturelle au cas des systèmes multi-agents, où les agents interagissent de façon stratégique, avec le but d'atteindre leur objectifs individuels et collectifs, sous la contrainte, entre autres, d'une observabilité partielle

L'objectif de cette thèse est l'étude des deux problèmes, MSPM et MCPM, dans leur version constructive, dans le cas de certaines logiques spécifiques qui ont un intérêt particulier en informatique et IA, typiquement des logiques modales et temporelles, où les interprétations sont des systèmes de transitions (structures de Kripke), qui sont particulièrement adaptées à la modélisation du comportement d'un système informatique.

Plus précisément, le projet explorera les cas de la logique dite « Linear Temporal Logic » (LTL), des logiques dites « Branching Time Logics » (CLTL et ses extensions), et leur généralisations multi-agent (comme la logique dite « Alternating Time Temporal Logic » -ATL- et ses extensions), parce-qu'elles permettent l'expression du comportement dynamique de plusieurs systèmes concurrents, distribués, et multi-agents.

Par rapport à MCPM, observons qu'une extension naturelle de ce projet, qu'on pourrait traiter aussi dans le travail de thèse, suppose que l'agent puisse mettre en place des expériences et faire des observations pour apprendre de plus en plus du modèle, jusqu'à quand il obtienne de l'information suffisante à résoudre le problème MCPM.
A very important problem in the process of designing and implementing a safety-critical computer system is to automatically test whether a given abstract model M of the system (built by using some formal technique, as automata or logical structures) verifies some property P required for that system, that is expressed in some logical language. Such a problem is usually known as "model checking problem". In most of the literature on model checking, the abstract model M is taken to be completely described. However, there are many natural scenarios, where the `agent' (which can be a designer, or user, or an agent acting in the system) only has an incomplete information about M. There can be at least two distinct reasons for that: either the system is still only partly defined or constructed, or the agent has only partial observability on that system. In either case, it can be assumed that the model belongs to some explicitly specified class of models C.

Therefore, one can consider two different yet related problems arising in such scenarios with incomplete information:

1. To complete the partial construction of M to a model within the class C so as to ensure the satisfaction of the property P, or to show that such completion does not exist. This we call the Model Synthesis from Partial Models problem (MSPM problem). In its constructive

version, this problem asks for the generation of some model in C ensuring P, extending the initially given partial model.

2. To verify that any model (from a given explicitly described class C) that meets the partial description that the agent has, satisfies the property P. In other words, to verify that any possible completion of the partly described model, within the class C, satisfies P.
This we call the Model Checking of Partial Models problem (MCPM problem).

We further assume that a logical language L is fixed in which the property P is expressed. Typically, L is a modal or temporal logic with semantics over interpreted transition systems, or a multi-agent extension of such logic. Thus, given the logic L, both MCPM and MPSM have a more precise formulation as logical decision problems:

1. MPSM: given a formula P of L, and a partially described model (or, partial interpretation) I of P, is there some interpretation I' extending I (in some precise sense, for instance within a given class C, or possibly depending on L) such that I' is a model of P, i.e. P is true in I'?
2. MCPM: given a formula P of L, and a partially described model (or partial interpretation) I of P, is it the case that all interpretations I' extending I (again, in some precise sense, for instance within a given class C) are models of P, i.e. make P true?

Both problems have various potential applications to AI (and in software engineering), for instance:

- |The MPSM problem naturally arises when a complex, multi-component system is to be built in a modular way, where each module designer is given a partial specification of how its component module must function and interact with the rest of the system, but not exactly how that component should be constructed.
- The MCPM problem naturally applies to systems where the designer or supervisor has only a partial observability on the system, or a partial knowledge about it, and based on that she must ensure that the system behaves correctly with respect to the given local or global specifications.

It should be noted that the problems of deciding satisfiability and validity and the model checking problem can be seen as respective limit cases of MSPM and MPCM.

Both problems naturally extend to multi-agent systems, where agents act and interact strategically, in pursuit of their individual and collective objectives, constrained by partial observability and other limitations.

The objective of this thesis is to study both the problems MPSM and MCPM, in their constructive versions, in the case of some specific logics that are of particular interest in computer science and AI, namely modal and temporal logics, where the interpretations are transitions systems (Kripke's structures), which are particularly suitable to model the behavior of a computer system.
More specifically, the project will explore the cases of linear time logic (LTL), branching time logics (CTL and extensions), and their multi-agent generalizations (such as the alternating-time temporal logic ATL and its extensions), because they enable the expression of the dynamical behavior of various concurrent, distributed, and multi-agent systems. These will constitute one of the main objects of study.

With respect to MCPM, let us observe that a natural extension of this problem, that might also be dealt with in the thesis work, assumes that the agent can set up suitable experiments and make observations to learn more and more from the model, until sufficient information is obtained to solve the MCPM problem.

Thématique / Domaine / Contexte

Synthèse de modèles et Model Checking.

Spécifications et Vérifications Formelles, Intelligence Artificielle.

Conception, vérification et réalisation de systèmes informatiques multi-agents en présence d'information partielle sur ces systèmes.

Objectifs

Les raisons qui font si que l'information sur le système de transitions (interprétation) soit incomplète sont multiples, et elles donneront lieu à des variations des scénarios à étudier :

• L'ensemble des états est donné complètement, mais leur étiquetage, c'est-à-dire l'information sur quelles propositions atomiques sont vraies à chaque état, est incomplète ;

• Chaque état est complétement décrit du point de vue statique, mais la relation de transition entre les états est inconnue, ou seulement partiellement connue ;
• Des états peuvent être ajoutés pour compléter la description du système ;
• Des combinaisons des cas ci-dessus.

Dans le cas spécifique des logiques multi-agents, on pourra considérer encore d'autres variations :

• Les agents sont connus et fixés, mais toutes leurs actions ne le sont pas ; l'ajout de plus d'actions conduira à l'ajout de nouvelles transitions ;
• Il est possible d'ajouter des nouveaux agents, avec leur capacité d'affecter le comportement du système. Une instance importante du problème MSPM dans ce cas est le problème de la synthèse du contrôleur, où le contrôleur cherché peut être vu comme un nouvel agent qui a la tâche d'assurer la satisfaction d'une spécification de sécurité donnée.

Certaines des variations des problèmes MPSM et MCPM esquissées ne sont pas forcément décidables, à priori. Donc une étude de la décidabilité et de la complexité de calcul de la version considérée du problème sera nécessaire. Pour ce cas pour lesquels on montre qu'ils sont décidables, l'objectif est de fournir des algorithmes qui les résolvent, ainsi qu'une analyse de complexité de ces algorithmes. La définition de ces algorithmes sera accompagnée par de l'implémentation, du test, de l'analyse de performance, et par des études de cas pratiques.

Méthode

Une méthodologie prévue pour la conception de ces algorithmes pourrait être l'utilisation des méthodes dites de « tableaux », actuellement utilisées dans la littérature essentiellement pour répondre aux problèmes logiques de satisfiabilité/validité. La définition de ces algorithmes sera accompagnée par de l'implémentation, du test, de l'analyse de performance, et par des études de cas pratiques.

Résultats attendus - Expected results

i) Des résultats théoriques de décidabilité/indécidabilité de certains problèmes;
ii) La définition d'algorithmes permettant de résoudre certains de ces problèmes, avec une analyse de leur complexité ;
iii) Programmation et mise en oeuvre de ces algorithmes, accompagnée par de la programmation, du test, de l'analyse de performance, et des études de cas pratiques.

Précisions sur l'encadrement - Details on the thesis supervision

Co-encadrement avec Valentin Goranko, , Department of Philosophy, Stockholm University valentin.goranko@philosophy.su.se https://www2.philosophy.su.se/goranko/ Le co-encadrement avec V. Goranko se fera en partie à distance (visioconférences) et en partie par le biais de voyages dans les deux directions et de séjours de recherche.

Conditions scientifiques matérielles et financières du projet de recherche

Ce projet de thèse sera faisable, du point de vue financier, seulement si le (la) doctorant(e) obtiendra une bourse.

Objectifs de valorisation des travaux de recherche du doctorant : diffusion, publication et confidentialité, droit à la propriété intellectuelle,...

On vise des publications à des congrès internationaux et à des revues internationales de très bon niveau. Il serait souhaitable aussi de produire des logiciels libres disponibles sur le web.

Collaborations envisagées

L'encadrement se fera en collaboration avec Valentin Goranko (Stockholm University).

Ouverture Internationale

L'encadrement se fera en collaboration avec Valentin Goranko (Stockholm University), qui est un chercheur ayant une forte réputation internationale dans le domaine de la logique et, en particulier, des logiques modales et multi-agents. Il comportera des visites de l'étudiant(e) en Suède ainsi que de séjours de recherche de Valentin Goranko à Evry.

Références bibliographiques

BIBLIOGRAPHIE PARTIELLE

Andersen, H.R.: Partial model checking (extended abstract). In: Proc. of LICS 1995. pp. 398--407. IEEE Computer Society (1995)

Andersen, H.R., Lind-Nielsen, J.: Partial model checking of modal equations: A survey. Intern. J. on Software Tools for Technology Transfer 2(3), 242--259 (1999)

Cerrito, S.: Proofs for temporal logics : focus on the multi-agent case. In A. Arana. Pataut, F., editors, Proofs, Lecture Notes in Logic. Cambrige UP, 2021. 36 pages. To appear.

Cerrito, S., David, A., Goranko, V.: Optimal tableau method for constructive satisfiability testing and model synthesis in the alternating-time temporal logic ATL+. {ACM} Trans. Comput. Log. 17(1), 4:1--4:34 (2015)

Demri, S., Goranko, V., Lange, M.: Temporal Logics in Computer Science. Cambridge Tracts in Theoretical Computer Science, Cambridge University Press (2016)

Goranko, V., Shkatov, D.: Tableau-based decision procedures for logics of strategic ability in multiagent systems. ACM Trans. Comput. Log. 11(1), 1--49 (2009)

Kupferman, O., Vardi, M.Y.: Synthesis with incomplete informatio. In: Barringer, H., Fisher, M., Gabbay, D., Gough, G. (eds.) 2nd Intern. Conf. on Advances in Temporal Logic. pp. 109--127. Springer Netherlands, Dordrecht (2000)

Staruch, B.: Extensions of partial structures and their application to modelling of multiagent systems. In: Monitoring, Security, and Rescue Techniques in Multiagent Systems, MSRAS 2004, Plock, Poland, June 7-9, 2004. Advances in Soft Computing, vol.~28, pp. 293--304. Springer (2005)

Dernière mise à jour le 26 mars 2021