STV+Reductions: Towards Practical Verification of Strategic Ability Using Model Reductions

Demonstration Track

Damian Kurpiewski
Institute of Computer Science, Polish Academy of Sciences
d.kurpiewski@ipipan.waw.pl

Wojciech Jamroga
Institute of Computer Science, Polish Academy of Sciences and University of Luxembourg
w.jamroga@ipipan.waw.pl

Witold Pazderski
Institute of Computer Science, Polish Academy of Sciences
w.pazderski@ipipan.waw.pl

Yan Kim
University of Luxembourg
yan.kim@uni.lu

ABSTRACT
We present a substantially expanded version of our tool STV for strategy synthesis and verification of strategic abilities. The new version adds user-definable models and support for model reduction through partial order reduction and checking for bisimulation.

KEYWORDS
formal methods; model checking; alternating-time temporal logic

ACM Reference Format:

1 INTRODUCTION
Formal analysis of multi-agent systems is becoming increasingly important as the procedures, protocols, and technology that surround us get more and more complex. Alternating-time temporal logic ATL [3, 4, 43] is probably the most popular logic to describe interaction in MAS. Formulas of ATL allow to express statements about what agents (or groups of agents) can achieve. For example, \(\langle\langle\text{taxi}\rangle\rangle G \neg \text{fatality}\) says that the autonomous cab can drive in such a way that nobody is ever killed, and \(\langle\langle\text{taxi}, \text{pass}\rangle\rangle F \text{destination}\) expresses that the cab and the passenger have a joint strategy to arrive at the destination, no matter what any other agents do.

Algorithms and tools for verification of such properties have been in development for over 20 years [1, 2, 7, 8, 11, 13–15, 21, 26, 32, 33, 37, 38, 40]. Unfortunately, model checking of agents with imperfect information is \(\Delta^P_2\)-complete to PSPACE-complete for memoryless strategies [9, 22, 43] and \(\text{EXPTIME}\)-complete to undecidable for agents with perfect recall [18, 20]; also, the problem does not admit simple incremental solutions [10, 16, 17]. This has been confirmed in experiments [11, 12, 21, 26, 40] and case studies [23, 25, 28].

Much of the complexity is due to the size of the model, and in particular to state space explosion [5]. To address the problem, we have extended our experimental tool STV (STrategic Verifier) [33] with support for model reductions. Two methods are used: (i) checking for equivalence of models according to a handcrafted relation of \(A\)-bisimulation [6], and (ii) fully automated partial order reduction (POR) [29, 31]. We also add a simple model specification language that allows the user to define their own inputs for verification, which was not available in the previous version [33].

The purpose of the extension is twofold. First, it should facilitate practical verification of MAS, as the theoretical and experimental results for POR and bisimulation-based reduction suggest [6, 31]. No less importantly, it serves a pedagogical objective, as we put emphasis on visualisation of the reductions, so that the tool can be also used in the classroom to show how the reduction works. Finally, checking strategic bisimulation by hand is difficult and prone to errors; here, the user can both see the idea of the bisimulation, and automatically check if it is indeed correct.

2 APPLICATION DOMAIN
STV+Reductions is aimed at verification of agents’ abilities – in particular, synthesis of memoryless imperfect information strategies that guarantee a given temporal goal. This includes both model checking of functionality requirements (understood as the ability of legitimate users to achieve their goals), and security properties defined by the inability of an intruder to compromise the system.

A good example of a specific domain is formal verification of voting procedures and elections, with a number of classical requirements, such as election integrity, ballot secrecy, receipt-freeness, and voter-verifiability [42, 44]. Some recent case studies [23, 25, 28] have shown that practical verification of such scenarios is still outside of reach. Some tools do not support intuitive specification and validation of models; some others have limited property specification languages. In all cases, the state-space explosion is a major obstacle that prevents verification of anything but toy models.

3 SCENARIOS
The new version of STV provides a flexible specification language for asynchronous models. The following examples are included: Train–Gate–Controller (TGC) [2, 31, 45], Two-Stage Voting [6], and
Asynchronous Simple Voting [31]. Some built-in synchronous models are also included, such as TianJi [37], Castles [41], Bridge Endplay [24], and Drones [27].

4 FORMAL BACKGROUND

Models. The main part of the input is given by an asynchronous multi-agent system (AMAS) [30, 31, 35], i.e., a network of local automata (one automaton per agent). From the AMAS, the global model is generated, where nodes are tuples of local states. The knowledge/uncertainty of an agent is defined by the agent’s local state. An example AMAS is shown in Figure 1(left). The global model generated from the AMAS is shown in Figure 1(right).

Strategies. A strategy is a conditional plan that specifies what the agent(s) are going to do in every possible situation. Here, we consider the case of imperfect information memoryless strategies, represented by functions from the agent’s local states (formally, abstraction classes of its indistinguishability relations) to its available actions. The outcome of a strategy from state q consists of all the infinite paths starting from q and consistent with the strategy.

Formulas. Given a model M and a state q in the model, the formula $\langle A \rangle \varphi$ holds in $(M, q)$ iff there exists a strategy for A that makes $\varphi$ true on all the outcome paths starting from any state indistinguishable from q. For more details, we refer the reader to [4, 43].

Model reduction and bisimulation. State space explosion is a major factor that prevents practical model checking [5]. A possible way out is model reduction, i.e., using a smaller equivalent model for verification instead of the original one. A suitable notion of A-bisimulation has been proposed in [6]. Unfortunately, synthesizing a reduced A-bisimilar model is at least as hard as the verification itself [6]. However, checking if a handcrafted relation is an A-bisimulation can be done in polynomial time, which offers valuable help especially for larger models.

Partial-order reduction. A fully automated model reduction is possible if the state space explosion is due to asynchronous interleaving of agents’ actions. The method is called partial order reduction, and has been recently extended to verification of strategic abilities under imperfect information [31]. The reduced model for the TGC scenario is highlighted in blue color in Figure 1(right).

5 TECHNOLOGY

STV+Reductions does explicit-state model checking. That is, the global states and transitions of the model are represented explicitly in the memory of the verification process. The tool includes the following new functionalities.

User-defined input. The user can load and parse the input specification from a text file that defines: the local automata in the AMAS, the formula to be verified, the propositional variables, persistent propositions, agent names relevant for POR, and/or the mapping for bisimulation checking. Based on that, the global model is generated and displayed in the GUI and can be verified by means of fix-point approximation [26] or dominance-based strategy search [34].

When using partial-order reduction, the reduced model is also displayed, and highlighted in the full model.

Partial-order reduction. The fully automated reduction method is based on POR [39] and implemented according to the algorithms proposed in [31, 36]. The reduced model is generated based on the AMAS specification, together with two additional parameters: the coalition and the set of proposition variables.

Bisimulation checking. The tool allows to check if two models are A-bisimilar for a given coalition A [6]. Apart from the specification of the two models, the bisimulation relation between the corresponding states must also be provided, along with the selected coalition.

6 USAGE

The current version of STV+Reductions is available for download here, and allows to:

- Select and display a model specification from a text file,
- Generate and display the explicit state-transition graph,
- Generate and display the reduced model using POR,
- Select specifications of two models and a relation from text files, and check if the models are A-bisimilar wrt the relation,
- Verify the selected full or reduced model by means of fixpoint approximation or dominance-based verification (DominDFS),
- Alternatively, run the verification for a predefined parameterized model and formula,
- Display the verification result, including the relevant truth values and the winning strategy.

7 CONCLUSIONS

Model checking strategic abilities under imperfect information is notoriously hard. STV+Reductions addresses the state explosion problem by an implementation of partial-order reduction and bisimulation checking. This should not only facilitate verification, but also make the techniques easier to use and understand.

Acknowledgements. The authors acknowledge the support of the Luxembourg National Research Fund (FNR) and the National Centre for Research and Development (NCBiR), Poland, under the CORE/PoLLux project STV (POLUX-VII/1/2019).