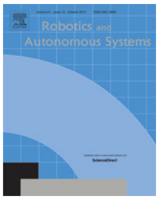
Robotics and Autonomous Systems 83 (2016) 44–56

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



Verification of logical consistency in robotic reasoning

Hongyang Qu, Sandor M. Veres [*]

Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, United Kingdom

HIGHLIGHTS

- Methods to make sure logical consistency in reasoning processes of intelligent software agents.
- Efficient realtime discovery of logical inconsistency by robots.
- Theorems on Boolean evolution systems stability in the reasoning of robotic agents.

ARTICLE INFO

Article history:
Received 21 September 2015
Accepted 14 June 2016
Available online 22 June 2016

Keywords:
Stability analysis
Robot programming
Formal verification

ABSTRACT

Most autonomous robotic agents use logic inference to keep themselves to safe and permitted behaviour. Given a set of rules, it is important that the robot is able to establish the consistency between its rules, its perception-based beliefs, its planned actions and their consequences. This paper investigates how a robotic agent can use model checking to examine the consistency of its rules, beliefs and actions. A rule set is modelled by a Boolean evolution system with synchronous semantics, which can be translated into a labelled transition system (LTS). It is proven that stability and consistency can be formulated as computation tree logic (CTL) and linear temporal logic (LTL) properties. Two new algorithms are presented to perform realtime consistency and stability checks, respectively. Their implementation provides us a computational tool, which can form the basis of efficient consistency checks on-board robots.

1. Introduction

A robotic system's decision making is well known to be in need of some hard decision making at times. A most popular example is Asimov's Laws [1], which demonstrate the difficulties to apply logic by robots in practice. A shortened version of these laws is "1. A robot may not allow a human being to come to harm. 2. A robot must obey the orders given to it by human beings except if the order causes harm to humans. 3. A robot must protect its own existence as long as such protection does not cause harm to humans." Assuming these, what would happen to the robot's decision making if a human commands a robot to kill someone, but at the same time threatens to kill himself if the robot does not obey? In this example, the human introduces a contradiction into the logic of the robot. To avoid this the robot may have a complex rule base to provide it with legal and ethical principles and can be equipped by a meta law which says that "the robot should not allow itself to be dictated by communicated conditions which make its logic contradictory". In this example, one could say that in legal terms the suicide will remain the sole "responsibility" of the threatening person who commands the robot.

The problem is not only the imperfection of Asimov's robotic laws or that an agent programmer can make mistakes. Logical consistency checks are also needed when the robot's perception-based beliefs are wrong. The agent can be programmed to re-examine whether its beliefs may need to be changed as were mistakenly believed to be true or false. This is not unlike enabling the agent to think like Poirot, Miss Marple or Sherlock Holmes when they are reassessing their initial beliefs or impressions. But there are simpler cases: a robot may decide that the book it sees on the table cannot be Tom's as that one is in his home. In this paper, we address the problem of how a robot can quickly and efficiently resolve inconsistencies in order to make the right decisions.

The ability of making fast decisions about logical consistency, and the robot's ability to detect when inconsistency occurs, is an important problem for the future of robotics. It is also of particular importance for logic-based robot control systems, e.g., [2–8]. A typical logic-based robotic system usually contains a belief set, which provides the basis of reasoning for a robot's behaviour [3]. An inconsistent belief set could lead to a wrong plan causing an unexpected result, e.g., an unmanned vehicle can hit an obstacle,

---

[*] Corresponding author.

E-mail addresses: h.qu@sheffield.ac.uk (H. Qu), s.veres@sheffield.ac.uk (S.M. Veres).

http://dx.doi.org/10.1016/j.robot.2016.06.005
0921-8890/© 2016 Published by Elsevier B.V.

instead of avoiding it, if it mistakenly believes that any route of avoidance could cause more damage, due to, for instance, misperception of the environment. Its mis-perception could perhaps be corrected if it had been able to combine environmental prior knowledge with current sensing.

In a rapidly changing environment, Bayesian methods can be used to identify and track movements of objects and establish functional relationships, e.g., [9]. When faced with balanced probabilities for two hypothetical and competing relationships in the robot's environment, it may need to make a decision based on the application of logic using prior knowledge. Discovery of logical inconsistency in geometrical and physical relationships in an environmental model should prompt a robotic agent to revise its perception model of the world. For instance, belief–desire–intention (BDI) agents should carry out consistency checks in their reasoning cycle in languages such as *Jason*, 2*APL* and *Jade* [10–14]. In these systems, the agent programmer should program logical consistency checks and handling of inconsistencies at design stage of the software.

To topic of fast consistency checking by robots has also implications for legal certification of robots. As we humans formulate social and legal behaviour rules in terms of logical implications, the process is likely to be similar for robots and the problem of consistent decisions by robots is an important generic capability. Future legal frameworks for certification of robots need to take into account verifiable decision making by robots.

Consistency checks on a set of logic rules in propositional logic is a textbook problem and has been extended to various types of logic systems in terms of validity, consistency and satisfiability. For instance, [15] provides an authoritative account of the history of logical consistency checking in a propositional logic. Relevant methods and algorithms have long been investigated for database systems and rule-based expert systems, e.g., [16], but none has been specifically designed for robotics. Query Language 4QL [17] and Boolean Networks (BN) [18] are very similar to our modelling formalism *Boolean evolution systems*. The former allows a variable to have four values: *true*, *false*, *unknown* and *inconsistent*. The algorithm that computes the unique well-supported model in [17] can be adapted to check consistency, but it can only deal with one initial evaluation of variables at a time. BN was developed for modelling gene regulatory networks in Biology. In BN, a Boolean variable can only take either *true* or *false*, while in our formalism, a variable can be initialized as *unknown*. Research on BDI reasoning cycles focuses on runtime detection and resolution of conflicting goals, such as [19,20]. No work has been conducted on complex reasoning process, which will be required by autonomous and intelligent robots.

For realtime robotic systems, it is important to increase solver efficiency to be able to deal with large search spaces with complex reasoning process for both offline and online application. In this respect, the use of binary decision diagram (BDD) is very effective by compressing search space through generating a unique and succinct representation of a Boolean formula. BDD has been widely adopted for model checking [21], and applied successfully to verification of large systems. In this paper, we adopt the BDD based symbolic model checking approach [22] to robotics. To our best knowledge, nothing has been reported on its application on consistency and stability checking of decisions by robots.

In this paper, we propose a fast method for discovery of inconsistency in a set of logic rules and statements on relationships in a current world model, past actions, planned actions and behaviour rules of a robotic agent. We do not address the problem of how to resolve logical inconsistency, mainly because we hold the view that, to eliminate inconsistencies, a robot can efficiently improve its world model by non-logic based techniques. Such techniques can include gathering more perception data, active vision, using alternative action plans or analysing and deriving spatial temporal models using probabilities. If a single new perception predicate or predicate derived by logic rules of the robot contradicts its otherwise consistent world model, then the robot may apply a set of logic rules to derive a correction of its belief in terms of the predicate. What to derive and analyse for consistency is however a broad topic and lies outside of the scope of this paper. Here, we focus on fast discovery of inconsistencies which is fundamental for safe operations of autonomous robots. With time it should be a key technical part in the process of legal certification of future autonomous robots.

Our contribution builds on and develops our past efficient state space generation and parallel computation [23] methods further. We have previously developed various state space reduction techniques for symbolic model checking via BDDs, such as symmetry reduction [24,25] and abstraction [26]. The preliminary results of our techniques have been published in [27]. In this paper, we elucidate the setting for which our techniques are designed and demonstrate their way of using it in robotics. We also extend the techniques to deal with a different semantics and develop a new technique to extract counterexamples efficiently when the system is inconsistent or unstable. The counterexamples are useful for system developers to correct robotic reasoning systems; they can provide guidance on how to improve the reasoning process of robots.

We study the efficiency of the agent's ability to examine the consistency of its beliefs and logic rules and, if inconsistency occurs, generate counterexamples to the rules which can then be used by the robot to resolve inconsistency. Our technique can be used both by robot programmers at software design stage and by robots when reasoning. In the former case, system developers can check the logical consistency of reasoning cycles in agent programs at design stage. For each inconsistent check, a counterexample can be produced to help developers understand the source of inconsistency and correct the program. In the latter case, consistency checks are carried out by the robots themselves in realtime and counterexamples are examined to improve reasoning, e.g., bringing in more sensor data to eliminate ambiguity or bring about alternative decisions about future actions.

In Section 2, we introduce the problem in a robotic framework and its characteristics. In Section 3, Boolean evolution systems are formally represented. In Section 4, we translate Boolean evolution systems into *transition systems* which are now widely used in the control systems literature [28,29], which provides the basis of verification. Note that in this paper we abstract robotic behaviour to propositional logic to be able to cope with computational complexity of consistency checking. Section 5 contains our results on stability of Boolean evolution systems in terms of CTL and LTL formulae. An important result states that stability checking can be reduced to a reachability problem which only asks for one fixpoint computation. Similarly, consistency checking can be also converted into simple fixpoint computation. Section 6 presents a case study in a home robotics scenario, which demonstrates the use of uncertain sensory and communication information and a set of rules to satisfy. In Section 7, performance comparison between CTL formulae based solutions and the reachability based algorithms is highlighted and implemented in the symbolic model checker MCMAS [30]. We discuss stability checking under an alternative semantics of evolution in Section 8. We conclude the paper in Section 9.

2. Perception clarification and robot logic

Our predicates-based knowledge representation of a robot, which is derived from sensing events, remembering the past as well as from prediction of a future environment, is schematically depicted in Fig. 1. For new sensory predicates, we assume that the robot is able to identify which ones are uncertain in a probabilistic sense. The following specific problems are to be addressed: