Information and Computation 262 (2018) 241–264

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Model checking for fragments of the interval temporal logic HS at the low levels of the polynomial time hierarchy ☆

Laura Bozzelli [a], Alberto Molinari [b], Angelo Montanari [b,*], Adriano Peron [a], Pietro Sala [c]

[a] Department of Electronic Engineering and Information Technologies, University of Napoli "Federico II", Italy
[b] Department of Mathematics, Computer Science, and Physics, University of Udine, Italy
[c] Department of Computer Science, University of Verona, Italy

ARTICLE INFO

Article history:
Received 21 April 2017
Available online 6 September 2018

Keywords:
Interval temporal logic
Model checking
Computational complexity

ABSTRACT

Some temporal properties of reactive systems, such as actions with duration and temporal aggregations, which are inherently interval-based, can not be properly expressed by the standard, point-based temporal logics LTL, CTL and CTL*, as they give a state-by-state account of system evolution. Conversely, interval temporal logics—which feature intervals, instead of points, as their primitive entities—naturally express them.

We study the model checking (MC) problem for Halpern and Shoham's modal logic of time intervals (HS), interpreted on Kripke structures, under the homogeneity assumption. HS is the best known interval-based temporal logic, which has one modality for each of the 13 ordering relations between pairs of intervals (Allen's relations), apart from equality. We focus on MC for some HS fragments featuring modalities for (a subset of) Allen's relations meet, met-by, started-by, and finished-by, showing that it is in $\mathbf{P}^{\mathbf{NP}}$. Additionally, we provide some complexity lower bounds to the problem.

1. Introduction

Our dependence on hardware and software systems is continuously increasing under many aspects of our everyday life. Embedded systems are employed for critical applications, e.g., air traffic and railway control systems, telephone networks, and nuclear plants monitoring. Security protocols are at the basis of e-commerce websites and services, and are exploited in all applications which have to ensure user privacy. Biomedical instruments and equipment are endowed with automatic or proactive functionalities, and are supposed to help humans and to prevent human error. Thus, the essential requirements of safety, reliability, and correctness for these systems suggest to support their development steps, namely, design, implementation, verification, and testing, by structured methodologies possibly founded on *formal methods*—some of them are even becoming integral part of standards—as well as by suitable specification languages and automatic verification techniques and tools.

☆ This paper is an extended and revised version of [40] and [10].

* Corresponding author.
E-mail addresses: lr.bozzelli@gmail.com (L. Bozzelli), molinari.alberto@gmail.com (A. Molinari), angelo.montanari@uniud.it (A. Montanari), adrperon@unina.it (A. Peron), pietro.sala@univr.it (P. Sala).

https://doi.org/10.1016/j.ic.2018.09.006
0890-5401/

A well known technique in this setting is *model checking*. Model checking (MC) allows one to verify the desired properties of a system against a model of its behaviour [19]. Properties are usually specified by means of temporal logics, such as LTL and CTL, and systems are represented as labelled state-transition graphs (Kripke structures). MC algorithms perform, in a fully automatic way, an (implicit or explicit) exhaustive enumeration of all the states reachable by the system, and either terminate positively—proving that all properties are met—or produce a counterexample—witnessing that some behaviour falsifies a property, which is extremely useful for debugging purposes.

MC can be applied during the early stages of the development cycle, allowing one to analyze even partial specifications, in such a way that it is not necessary to completely describe a system before information can be obtained regarding its correctness. MC has been applied in a variety of practical scenarios, including, for instance, communication and security protocols [2,3], embedded reactive systems [18], computer device drivers [57], database-backed web applications [24], concurrency control and transaction atomicity [35], automated verification of UML design of applications [20], testing of railway control systems [5,44], and verification of clinical guidelines [22].

The MC problem has been investigated for a long time only in the context of *point-based temporal logics*, like LTL, CTL, and CTL*, which predicate over single system/computation states [21,46,47,51]. For instance, LTL allows one to reason about changes in the truth value of formulas in a Kripke structure over a linearly-ordered temporal domain, where each moment in time has a unique possible future. More precisely, one has to consider all possible paths in a Kripke structure and to analyse, for each of them, how proposition letters, labelling the states, change from one state to the next one along the path. A systematic investigation of MC for *interval temporal logics* has been initiated very recently.

Interval temporal logics (ITLs) [27,53,54] feature intervals, instead of points, as their primitive entities. This makes them a highly expressive formalism for temporal representation and reasoning, with the ability of easily "mastering" advanced temporal features, such as actions with duration, accomplishments, and temporal aggregations, which can not be properly dealt with by standard, point-based temporal logics. ITLs have been applied in a variety of computer science fields, including artificial intelligence (reasoning about action and change, qualitative reasoning, planning, multi-agent systems, and computational linguistics), theoretical computer science (formal verification), and databases (temporal and spatio-temporal databases) [6,23,25,34,43,48,58]. However, the great expressiveness of ITLs is a double-edged sword: in most cases, the satisfiability problem for ITLs turns out to be undecidable, and, in the few decidable ones, the standard proof machinery, like Rabin's theorem, is usually not applicable.

The best known ITL is *Halpern and Shoham's modal logic of time intervals* (HS, for short) [27], which has one modality for each of the 13 possible ordering relations between pairs of intervals (the so-called Allen's relations [1]), apart from equality. In [27], the authors prove that the satisfiability problem for HS, interpreted over all relevant (classes of) linear orders, is undecidable. The investigation has been later extended to many HS fragments, leading to the conclusion that undecidability prevails over them as well (see [12] for an up-to-date account of undecidable fragments). However, meaningful exceptions exist, including the interval logic of temporal neighbourhood $\mathsf{A\overline{A}}$ and the interval logic of sub-intervals D [13–15,41,42].

In this paper, we focus on the *MC problem for HS*, which has entered the research agenda only recently [31–33,36–39] (it is worth pointing out that, in contrast to the case of point-based, linear temporal logics, there is no easy reduction from the MC problem to validity/satisfiability for ITL). In interval-based MC, in order to verify interval properties of computations, one needs to collect information about the states of a system into computation stretches. To this aim, we interpret each finite path of a Kripke structure (a trace) as an interval, and we define the labelling of an interval on the basis of the proposition letters which hold on the sequence of states that compose it. Different ways of defining interval labelling have been proposed in the literature. A short account of them is given in the related work section below.

1.1. Related work

In [36], Molinari et al. gave a first characterization of MC for full HS interpreted over finite Kripke structures, under the *homogeneity assumption* [49], according to which a proposition letter holds over an interval if and only if it holds at all its states. In that paper, the authors showed that finite Kripke structures can be suitably mapped into interval-based structures, called abstract interval models, over which HS formulas can be interpreted. Then, they proved a small model theorem showing (with a non-elementary procedure) the decidability of MC for full HS, which was later proved to be **EXPSPACE**-hard by Bozzelli et al. in [8].[1]

The MC problem for some large fragments of HS was studied in [8,37,38]. In [38], Molinari et al. devised an **EXPSPACE** MC algorithm for the HS fragment $\mathsf{A\overline{A}B\overline{B}E}$ (resp., $\mathsf{A\overline{A}E\overline{B}E}$) of Allen's relations *meets*, *met-by*, *starts*, *finishes*, and *started-by* (resp., *finished-by*), which exploits the possibility of finding, for each trace (of unbounded length), an equivalent bounded-

[1] It is worth pointing out that the homogeneity assumption, which allows us to interpret HS formulas on Kripke structures in a fairly natural way, changes the status of the satisfiability problem for HS and its fragments. In particular, in [9] Bozzelli et al. showed that, when interpreted over the (infinite) fullpaths of a finite Kripke structure (which is not the way we interpret it here), LTL and HS have the same expressive power, but the latter is provably at least exponentially more succinct. As a byproduct, the satisfiability problem for full HS, under such a trace-based semantics, turns out to be decidable. Thus, under the homogeneity assumption, the relevant issue for the satisfiability problem of HS and its fragments becomes its complexity, rather than its decidability. We addressed it for the interval logic of sub-intervals D in [11].