Information and Computation 209 (2011) 766–781

Contents lists available at ScienceDirect

Information and Computation

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

Model-checking games for fixpoint logics with partial order models

Julian Gutierrez *, Julian Bradfield

LFCS, School of Informatics, University of Edinburgh, Informatics Forum, 10 Crichton Street, Edinburgh EH8 9AB, UK

ARTICLE INFO

Article history:
Available online 15 December 2010

Keywords:
Fixpoint modal logics
Model-checking games
Concurrency

ABSTRACT

In this paper, we introduce model-checking games that allow *local* second-order power on *sets* of independent transitions in the underlying partial order models where the games are played. Since the interleaving semantics of such models is not considered, some problems that may arise when using interleaving representations are avoided and new decidability results for partial order models of concurrency are achieved. The games are shown to be *sound* and *complete*, and therefore determined. While in the interleaving case they coincide with the local model-checking games for the μ-calculus, in a partial order setting they verify properties of a number of fixpoint modal logics that can specify, in concurrent systems with partial order semantics, several properties not expressible with the μ-calculus. The games underpin a novel decision procedure for model-checking all temporal properties of a class of infinite and regular event structures, thus improving, in terms of temporal expressive power, previous results in the literature.

1. Introduction

Model-checking games [12,35], also called Hintikka evaluation games, are played by two players, a "Verifier" Eve ($\exists$) and a "Falsifier" Adam ($\forall$). These *logic games* [2] are played in a formula ϕ and a mathematical model $\mathfrak{M}$. In a game $\mathcal{G}(\mathfrak{M}, \phi)$ the goal of Eve is to show that $\mathfrak{M} \models \phi$, while the goal of Adam is to refute such an assertion. Solving these games amounts to answering the question of whether or not Eve has a strategy to win the game $\mathcal{G}(\mathfrak{M}, \phi)$. These games have a long history in mathematical logic and in the last two decades have become an active area of research in computer science, both from theoretical and practical view points. Good introductions to the subject can be found in [12,33].

In concurrency and program verification, most usually ϕ is a modal or a temporal formula and $\mathfrak{M}$ is a Kripke structure or a labelled transition system (LTS), i.e., a graph structure, and the two players play the game $\mathcal{G}(\mathfrak{M}, \phi)$ *globally* by picking single elements of $\mathfrak{M}$, according to the game rules defined by ϕ. This setting works well for concurrent systems with *interleaving* semantics since one always has a notion of global state enforced by the nondeterministic sequential computation of atomic actions, which in turn allows the players to choose only single elements of the structure $\mathfrak{M}$. However, when considering concurrent systems with partial order models [26], explicit notions of *locality* and *concurrency* have to be taken into account. A possible solution to this problem – the traditional approach – is to use the one-step interleaving semantics of such models in order to recover the *globality* and *sequentiality* of the semantics of formulae.

This solution is, however, problematic for at least five reasons. Firstly, interleaving models usually suffer from the state space explosion problem [4]. Secondly, interleaving interpretations cannot be used to give completely satisfactory game semantics to logics with partial order models as all information on independence in the models is lost in the interleaving simplification [1]. Thirdly, although temporal properties can still be verified with the interleaving simplification, properties involving concurrency, causality and conflict, natural to partial order models of concurrency, can no longer be verified [28].

* Corresponding author.
E-mail addresses: J.E.Gutierrez@ed.ac.uk (J. Gutierrez), jcb@inf.ed.ac.uk (J. Bradfield).

0890-5401/$ - see front matter © 2011 Julian Gutierrez and Julian Bradfield. Published by Elseiver Inc. All rights reserved.
doi:10.1016/j.ic.2010.12.002

From a more practical standpoint, partial order reduction methods [9,11] or unfolding techniques [8] cannot be applied directly to interleaving models in order to build less complex model checkers based on these techniques. Finally, the usual techniques for verifying interleaving models cannot always be used to verify partial order ones since such problems may become undecidable [21,27].

For these reasons, we believe that the study of verification techniques for partial order models continues to deserve much attention since they can help alleviate some of the limitations related with the use of interleaving models. We therefore abandon the traditional approach to defining model-checking games for logics with partial order models and propose a new class of games called 'trace local monadic second-order (LMSO) model-checking games', where *sets* of independent elements of the structure at hand can be *locally* recognized. These games avoid the need of using the one-step interleaving semantics of partial order models, and thus define a more natural framework for analysing fixpoint modal logics with noninterleaving semantics. Moreover, their use in the temporal verification of a class of regular event structures [34] improves previous results in the literature [21,27]. We do so by allowing a free interplay of fixpoint operators and local second-order power on *conflict-free* sets of transitions.

The logic we consider is Separation Fixpoint Logic (SFL) [14], a μ-calculus (L_μ) [19] extension that can express causal properties in partial order models [26], e.g., transition systems with independence, Petri nets or event structures, and allows for doing *dynamic* local reasoning. The notion of locality in SFL, namely separation or disjointness of independent sets of resources, was inspired by the one defined *statically* for Separation Logic [29]. Since SFL is as expressive as L_μ in an interleaving context, nothing is lost with respect to the main approaches to logics for concurrency with interleaving semantics. Instead, logics and techniques for interleaving concurrency are extended to a partial order setting with SFL.

The structure of the paper is as follows: in Section 2, we introduce the partial order models of concurrency that are used in the paper and in Section 3 the syntax and semantics of SFL is defined. In Section 4, trace LMSO model-checking games are defined, and in Section 5 their soundness and completeness is proved. In Section 6, we show that the games are decidable and their coincidence with the local model-checking games for L_μ in the interleaving case. In Section 7, the game is used to effectively model-check a class of regular and infinite event structures. Finally, in Section 8 a summary of related work is given, and in Section 9 the paper concludes.

2. Preliminaries

This section introduces the background material that is needed in the following sections, namely the partial order models of our interest.

2.1. Partial order models of concurrency

In concurrency there are two main approaches to modelling concurrent behaviour. On the one hand, interleaving models represent concurrency as the nondeterministic combination of all possible sequential behaviours in the system. On the other hand, partial order models represent concurrency explicitly by means of an independence relation on the set of actions, transitions or events in the system that can be executed concurrently.

We are interested in partial order models of concurrency for several reasons. In particular, because they can be seen as a generalization of the interleaving models as will be explained later on in this section. This allows us to define the model-checking games presented here in a uniform way for several different models of concurrency, regardless of whether they have an interleaving or a partial order semantics. In the following, we present the three partial order models of concurrency that we consider here, namely Petri nets, transition systems with independence and event structures [26]. We also present some basic relationships between these three models, and how they generalize two important models for interleaving concurrency, which are also embraced in the uniform framework for model-checking we propose here. For further information the reader is referred to [26,30] where one can find a more comprehensive presentation.

2.1.1. Petri nets

A labelled *net* $\mathcal{N}$ is a tuple $(P, A, \mathcal{W}, \mathcal{F}, \Sigma)$, where P is a set of places, A is a set of actions, $\mathcal{W} \subseteq (P \times A) \cup (A \times P)$ is a relation between places and actions, and $\mathcal{F}$ is a labelling function $\mathcal{F} : A \to \Sigma$ from actions to a set Σ of action labels. Places and actions are called nodes; given a node n, ${}^\bullet n = \{x \mid (x, n) \in \mathcal{W}\}$ is the preset of n and $n^\bullet = \{y \mid (n, y) \in \mathcal{W}\}$ is the postset of n. These elements define the static structure of a net.[1] The notion of computation state in a net (its dynamic part) is that of a 'marking', which is a set or a multiset of places; in the former case such nets are called *safe*. Hereafter we only consider safe nets. Finally, a *Petri net* $\mathfrak{N}$ is a tuple $(\mathcal{N}, M_0)$, where $\mathcal{N} = (P, A, \mathcal{W}, \mathcal{F}, \Sigma)$ is a net and $M_0 \subseteq P$ is its initial marking.

As mentioned before markings define the dynamics of nets; they do so in the following way. We say that a marking M enables an action t iff ${}^\bullet t \subseteq M$. If t is enabled at M, then t can occur and its occurrence leads to a successor marking M', where $M' = (M \setminus {}^\bullet t) \cup t^\bullet$, written as $M \xrightarrow{t} M'$. Let $\xrightarrow{t}$ be the relation between all successive markings, and $\longrightarrow^*$ the reflexive

[1] The reader acquainted with net theory may have noticed that we use the word 'action' instead of 'transition', more common in the literature on (Petri) nets. We chose this notation in order to avoid confusion later on in the document.