Theoretical Computer Science 551 (2014) 84–101

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Recent advances in unfolding technique

Blai Bonet[a], Patrik Haslum[b], Victor Khomenko[c,*], Sylvie Thiébaux[b], Walter Vogler[d]

[a] Departamento de Computación, Universidad Simón Bolívar, Caracas, Venezuela
[b] The Australian National University & NICTA, Canberra, Australia
[c] School of Computing Science, Newcastle University, Newcastle upon Tyne, UK
[d] Institut für Informatik, Universität Augsburg, Augsburg, Germany

ARTICLE INFO

Article history:
Received 10 December 2013
Received in revised form 6 May 2014
Accepted 6 July 2014
Available online 15 July 2014
Communicated by G. Rozenberg

Keywords:
Adequate order
Unfolding prefix
Petri net
Well-foundedness
Concurrency

ABSTRACT

We propose a new, and to date the most general, framework for Petri net unfolding, which broadens its applicability, makes it easier to use, and increases its efficiency. In particular: (i) we propose a user-oriented view of the unfolding technique, which simply tells which information will be preserved in the final prefix and how to declare an event a cut-off in the algorithm, while hiding the technical parameters like the adequate order; (ii) the notion of the adequate order is generalised to a well-founded relation, and the requirement that it must refine $\subset$ is replaced by a weaker one; and (iii) the order in which the unfolding algorithm selects the possible extensions of the prefix is entirely disentangled from the cut-off condition. We demonstrate the usefulness of the developed theory on some case studies.

1. Introduction

A distinctive characteristic of reactive concurrent systems is that their sets of local states have descriptions which are both short and manageable, and the complexity of their behaviour comes from highly complicated interactions with the external environment rather than from complicated data structures and manipulations thereon. One way of coping with this complexity problem is to use formal methods and, especially, computer aided verification tools implementing model checking (see, e.g. [4]) — a technique in which the verification of a system is carried out using a finite representation of its state space.

The main drawback of model checking is that it suffers from the *state space explosion* problem. That is, even a relatively small system specification can (and often does) yield a very large state space. To alleviate this problem, a number of techniques have been proposed. Among them, a prominent technique is McMillan's (finite prefixes of) Petri net unfoldings (see, e.g. [12,26]). It relies on the partial order view of concurrent computation, and represents system states implicitly, using an acyclic net. More precisely, given a Petri net Σ, the unfolding technique aims at building a labelled acyclic net, called *(unfolding) prefix*, satisfying two key properties:

* Corresponding author.
E-mail addresses: bonet@ldc.usb.ve (B. Bonet), patrik.haslum@anu.edu.au (P. Haslum), Victor.Khomenko@ncl.ac.uk (V. Khomenko), sylvie.thiebaux@anu.edu.au (S. Thiébaux), vogler@informatik.uni-augsburg.de (W. Vogler).

http://dx.doi.org/10.1016/j.tcs.2014.07.003
0304-3975/© 2014 Elsevier B.V. All rights reserved.

– *Completeness:* each reachable marking of Σ is represented by at least one 'witness', i.e. a marking of the prefix reachable from its initial marking; similarly, for each possible firing of a transition at any reachable state of Σ there is a suitable 'witness' event in the prefix.
– *Finiteness:* the prefix is finite and thus can be used as an input to model checking algorithms, e.g. those searching for deadlocks. The finiteness is commonly achieved by identifying a set of *cut-off* events beyond which the unfolding is not generated.

A prefix satisfying these two properties can be used for model checking as a condensed representation of the state space of a system.

Practical algorithms exist for building complete prefixes [12,19], which ensure that the number of (non-cut-off) events in such a prefix can never exceed the number of reachable states of the Petri net. However, complete prefixes are often much smaller than the corresponding state graphs, especially for highly concurrent Petri nets, because they represent concurrency directly rather than by multidimensional 'diamonds' as it is done in state graphs. For example, if the original Petri net consists of 100 transitions which can fire once in parallel, the state graph will be a 100-dimensional hypercube with 2^{100} vertices, whereas the complete prefix will coincide with the net itself. In many applications, e.g. in asynchronous circuit design, the Petri net models usually exhibit a lot of concurrency, but have rather few choice points, and so their unfolding prefixes are often exponentially smaller than the corresponding state graphs; in fact, in many of the experiments conducted in [19] they are just slightly bigger then the original Petri nets themselves. Therefore, unfolding prefixes are well-suited for alleviating the state space explosion problem.

In this paper, we investigate a number of important generalisations of unfolding, which broaden its applicability and increase its efficiency. We generalise the notion of a *cutting context* [22] (a parametric framework for identifying cut-off events) in the following directions:

– The *adequate order* $\lhd$, which is commonly used to define cut-off events of the prefix, does not have to be part of the cutting context — merely the *existence* of a suitable relation $\lhd$ is sufficient for ensuring that the prefix is complete. This provides a *user-oriented view* of the unfolding technique, which simply tells which information will be preserved in the final prefix and how to declare an event a cut-off in the algorithm, while hiding the technical parameters like the adequate order.
– The adequate order is replaced by a well-founded relation. Moreover, the requirement that it must refine $\subset$ is replaced by a weaker one; cf. the definition of semi-adequate orders in [2].
– The order in which the unfolding algorithm selects the possible extensions of the prefix is entirely disentangled from the cut-off condition.

The latter generalisation in particular has important applications, see Section 8. For instance, it enables efficient search strategies to be used to construct the unfolding or perform reachability analysis, without sacrificing completeness. Without disentanglement, even a simple depth-first search strategy leads to incompleteness [11]. Disentanglement also enables the unfolding technique to find *optimal* solutions to reachability problems, even for notions of optimality which, such as makespan, cannot be captured by adequate (or semi-adequate) orderings.

The paper is organised as follows. Section 2 explains basic notions related to Petri nets and unfolding prefixes. In Section 3 we present a new unfolding algorithm, where the selection of a possible extension is entirely disentangled from the cut-off condition. In Section 4, we define our new notion of a cutting context, which is central to this paper. Section 5 proves the correctness of the proposed unfolding algorithm. Section 6 gives an overview of related work in the area of unfolding prefixes and argues in some detail that the newly proposed theory and unfolding algorithm generalise previous approaches; as a first application, we show that in our setting also depth-first search can be applied, which is somewhat in contrast to earlier results. In Section 7, we demonstrate how the canonicity result of [22] can be restored by restricting the selection strategy used by the new unfolding algorithm. Section 8 demonstrates the usefulness of the developed theory and unfolding algorithm on a number of applications: In particular, the disentanglement of the selection strategy from the cut-off condition allows one to use some extra information (in the form of a heuristic function) to guide the unfolding algorithm towards a target configuration, to prune 'dead ends' (configurations that cannot be extended to a target configuration), and to find an optimal target configuration (according to various measures of cost). Section 9 concludes the paper and gives some possible directions for future work.

2. Basic notions

In this section, we first present basic definitions concerning Petri nets, and then recall (see also [9,12]) notions related to net unfoldings.

2.1. Petri nets

A *net* is a triple $N \overset{\mathrm{df}}{=} (P, T, F)$ such that P and T are disjoint sets of respectively *places* and *transitions*, and $F \subseteq (P \times T) \cup (T \times P)$ is a *flow relation*. A *marking* of N is a multiset M of places, i.e. $M : P \to \mathbb{N} = \{0, 1, 2, \ldots\}$. We adopt the standard