

Contents lists available at [ScienceDirect](#)

Information Sciences

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

An extension of the taxonomy of persistent and nonviolent steps



Maciej Koutny^{a,*}, Łukasz Mikulski^{a,b}, Marta Pietkiewicz-Koutny^a

^a School of Computing Science, Newcastle University, Newcastle upon Tyne, NE1 7RU, United Kingdom

^b Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, Chopina 12/18, Toruń, Poland

ARTICLE INFO

Article history:

Received 13 November 2015

Revised 28 January 2017

Accepted 31 January 2017

Available online 31 January 2017

Keywords:

Persistence

Nonviolence

Step semantics

Single-server execution semantics

Petri net

Taxonomy

Behaviour

Structure

ABSTRACT

The design and analysis of concurrent computing systems is often concerned with fundamental behavioural properties involving system activities, e.g., boundedness, liveness, and persistence. This paper is about the latter property and a complementary property of non-violence. Persistence means that an enabled activity cannot be disabled, whereas nonviolence means that executing an activity does not disable any other enabled activity.

Since its introduction in the 1970s, persistence has been investigated assuming that each system activity is a single atomic action, but in the design of Globally Asynchronous Locally Synchronous (GALS) systems one also needs to allow activities represented by steps, each step being a set of simultaneously executed atomic actions. Dealing with step based execution semantics creates a wealth of new fundamental problems and questions. In particular, there are different ways in which the standard notion of persistence (and nonviolence) could be lifted to the level of steps.

We provide a rich classification of different types of step based persistence and nonviolence. We first do this for a general model of (step) transition systems. After that, we focus on Petri nets, and introduce a taxonomy of persistent and nonviolent steps and markings. We also characterise key structural properties of persistence and nonviolence, linking these behavioural notions with the presence of self-loops in Petri nets.

© 2017 The Authors. Published by Elsevier Inc.
This is an open access article under the CC BY license.
(<http://creativecommons.org/licenses/by/4.0/>)

1. Introduction

The concept of concurrency in today's computing is prevalent and can be found in, e.g., hardware systems, programming languages, global computer networks, operating systems, and models of biochemical reactions in living cells [9,12,21,24,28]. As the complexity of concurrent systems grows rapidly, it is increasingly challenging to guarantee their correctness. A key issue is to provide appropriate formal models and abstractions for capturing their behavioural properties. The main model used in this paper is Petri nets [25,26] as they provide a framework in which both state-based and action-based system information is represented explicitly, and also because they had been extensively used in the past to treat persistence and nonviolence. A part of the technical development will be formulated in terms of more general transition systems [17] which provide a semantical link between Petri nets and other concurrency models, e.g., process calculi.

* Corresponding author.

E-mail address: maciej.koutny@ncl.ac.uk (M. Koutny).

To model (ongoing) behaviour of computing systems one can use sequences of executed actions, but such a sequential view of system behaviour may not be optimal for dealing with concurrent systems. For example, to describe the result of action refinement a suitable information about concurrency or independence between actions may be needed, as in the traces of [11], resulting in a variant of causal partial orders semantics. We use a semantical model which lies in-between the sequential and causal partial order ones. A key idea is that the behaviours of concurrent systems are represented by sequences of groups (or *steps*) of simultaneously executed activities [27]. We follow the basic variant of the step sequence approach—often referred to as the single-server semantics [15]—where steps are sets of executed actions. This fits well, e.g., the operational behaviour of asynchronous circuits [13,23] and reaction systems [12]. Moreover, in the context of Petri nets with time, the single-server semantics means that a single clock is assigned to an action and, as argued in [15], such an approach has been used most widely for modelling real-time concurrent computing systems. The alternative multiple-server and infinite-server semantics, both outside the scope of this paper, define steps as multisets of actions.

Persistence and nonviolence. The design and analysis of concurrent systems is very often concerned with fundamental behavioural properties involving system activities, including *persistence* as well as a complementary property of *nonviolence* [2,4]. A concurrent system is persistent [4,5,20] if no enabled activity can be prevented from being executed by any other activity. This is often a highly desirable property, in particular, in hardware systems [31]. A complementary nonviolence property means that executing an activity cannot disable any other enabled activity. In other words, persistence means that an enabled activity can be delayed but not disabled, whereas nonviolence means that executing an activity can delay other enabled activities, but cannot disable them.

Persistence is one of crucial properties in the design of asynchronous hardware [9], and in the design of arbiter-free synchronisations [19]. Other areas where concepts related to persistence turned out to play a key role are, e.g., continuous Petri nets, introduced as approximations for coping with the state explosion problem in the verification of discrete (standard) Petri nets [16,22], and performance evaluation of discrete event systems [15]. There are several structurally defined and extensively applied persistent sub-classes of Petri nets, including choice-free nets [14,30], marked graphs [8], and MTS nets [5,16,22] demonstrated that persistence (together with other properties) guarantees separability which means that a Petri net can be seen and verified as a composition of finitely many concurrently operating independent copies of much simpler net [6,7] presented a way in which persistent nets can be synthesised and re-engineered from finite transition systems.

In the past work on different notions related to persistence, the underlying assumption was that it is a property of a sequential execution semantics. Recently, in [13] we argued that such a notion of persistence is restricted and in dealing with the design of GALS systems one also needs to consider activities represented by sets of simultaneously executed transitions. Moving into the realm of step based execution semantics creates a wealth of new fundamental problems and intriguing questions, some of which have been addressed in [13,18]. In particular, there are different ways in which the standard notion of persistence could be lifted from the level of sequential semantics to the level of step semantics. Moreover, one may consider steps which are persistent and cannot be disabled by other steps, as well as steps which are nonviolent [2,4] and cannot disable other steps. The decidability status of various notions of step based persistence and nonviolence has been investigated in [1].

Step persistence in asynchronous circuits. The original motivation for the work presented here came from a challenge to deal with the design of asynchronous circuits [10]. One of the main issues in asynchronous circuit design is the handling of hazards which can be interpreted as a violation of persistence [31] in models such as transition systems [17] or Petri nets [9]. Although synchronous circuits are hazard-free, the inherent adaptiveness of the asynchronous circuits means that they can cope much better with chip sizes scaling to deep sub-micron level. An attractive compromise turns out to be mixed synchronous-asynchronous GALS design [29], where a digital system is divided into hazard-free synchronous islands communicating asynchronously using potentially hazardous handshake mechanism.

To model circuit behaviour exhibiting parallelism between actions in the same clock cycle and sequential order between groups of actions in adjacent clock cycles, one can use step semantics. This, in turn calls for an adequate capture of a persistence in the context of the step semantics. In [13], we proposed such a notion and applied it in a procedure for turning reachability graphs of sequentially persistent safe nets into step persistent transition systems from which correct GALS circuits can be derived.

About this paper. We aim to classify different types of persistent and nonviolent steps taking PT-nets (Place/Transition nets) [26] to be the system model, revising and extending the results of [18]. In particular, we define three *distinct* concepts of step persistence and nonviolence. We also introduce and investigate persistence and nonviolence with respect to the markings of PT-nets. Another aim is to investigate behavioural and structural properties pertaining to persistence and nonviolence both for the general PT-nets and safe PT-nets.

The paper is organised as follows. In the next section, we summarise basic notions and notations. In Section 3, we present definitions of persistent and nonviolent steps in the setting provided by transition systems. Section 4 introduces various types of persistent and nonviolent steps of transitions in PT-nets, and Section 5 provides their taxonomy. The following section extends the discussion of persistence and nonviolence to markings of PT-nets. Section 7 investigates persistent and nonviolent steps of transitions in PT-nets, and Section 8 focuses specifically on safe PT-nets.

Download English Version:

<https://daneshyari.com/en/article/4944591>

Download Persian Version:

<https://daneshyari.com/article/4944591>

[Daneshyari.com](https://daneshyari.com)