Journal of Logical and Algebraic Methods in Programming 84 (2015) 853–883

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

A coalgebraic semantics for causality in Petri nets ☆

Roberto Bruni [a], Ugo Montanari [a], Matteo Sammartino [b,*]

[a] University of Pisa, Computer Science Department, Largo Bruno Pontecorvo 3, 56127 Pisa, Italy
[b] Radboud University, Institute for Computing and Information Sciences, Faculty of Science, Heyendaalseweg 135, 6525AJ Nijmegen, The Netherlands

ARTICLE INFO

Article history:
Received 1 December 2014
Received in revised form 7 May 2015
Accepted 10 July 2015
Available online 17 July 2015

Keywords:
Petri nets
Causal case graphs
Behavior structures
Presheaves
Coalgebras
HD-automata

ABSTRACT

In this paper we revisit some pioneering efforts to equip Petri nets with compact operational models for expressing causality. The models we propose have a bisimilarity relation and a minimal representative for each equivalence class, and they can be fully explained as coalgebras on a presheaf category on an index category of partial orders. First, we provide a set-theoretic model in the form of a *causal case graph*, that is a labeled transition system where states and transitions represent markings and firings of the net, respectively, and are equipped with causal information. Most importantly, each state has a poset representing causal dependencies among past events. Our first result shows the correspondence with behavior structure semantics as proposed by Trakhtenbrot and Rabinovich. Causal case graphs may be infinitely-branching and have infinitely many states, but we show how they can be refined to get an equivalent finitely-branching model. In it, states only keep the most recent causes for each token, are up to isomorphism, and are equipped with a symmetry, i.e., a group of poset isomorphisms. Symmetries are essential for the existence of a minimal, often finite-state, model. This first part requires no knowledge of category theory. The next step is constructing a coalgebraic model. We exploit the fact that events can be represented as names, and event generation as name generation. Thus we can apply the Fiore–Turi framework, where the semantics of nominal calculi are modeled as coalgebras over presheaves. We model causal relations as a suitable category of posets with action labels, and generation of new events with causal dependencies as an endofunctor on this category. Presheaves indexed by labeled posets represent the functorial association between states and their causal information. Then we define a well-behaved category of coalgebras. Our coalgebraic model is still infinite-state, but we exploit the equivalence between coalgebras over a class of presheaves and *History Dependent automata* to derive a compact representation, which is equivalent to our set-theoretical compact model. Remarkably, state reduction is automatically performed along the equivalence.

☆ Research supported by the EU Integrated Project 257414 ASCENS, by the Italian MIUR Project CINA (PRIN 2010LHT4KM) and by the NWO Project Practical Coinduction (grant number 612.001.113).

* Corresponding author. Tel.: +31 0243652642.
E-mail addresses: bruni@di.unipi.it (R. Bruni), ugo@di.unipi.it (U. Montanari), m.sammartino@cs.ru.nl (M. Sammartino).
URLs: http://www.di.unipi.it/~bruni (R. Bruni), http://www.di.unipi.it/~ugo (U. Montanari), http://www.cs.ru.nl/M.Sammartino/ (M. Sammartino).

http://dx.doi.org/10.1016/j.jlamp.2015.07.003
2352-2208/

1. Introduction

Petri nets are a well-known graphical and formal notation for representing concurrent computations. An interesting aspect of Petri nets is that they allow for the representation of causal dependencies among actions. This kind of information can be useful for debugging distributed systems or for tracing expected or unwanted causal dependencies, and it is usually not provided by interleaving models.

In order to carry out verification on Petri nets, it is convenient to have an *operational* model, that is a model representing single steps of computation and their observable actions. In Petri nets, steps are typically firings and actions are action labels of transitions. One important class of operational models for Petri nets are *behavior structures* [27]. They are automata where each state is equipped with a partial order over events: events represent different occurrences of actions and the poset describes causal dependencies among such occurrences. Behavior structures come with a notion of behavioral equivalence, which later has been called *history preserving bisimilarity* [14].

Other causal models, such as *event structures* [20], do not come with a built-in operational notion of bisimilarity. Such a notion is essential to compute minimal models, where all states with the same behavior are identified. Open maps [16] can be used to derive *hereditary history preserving bisimulations* (HHPBs), but the existence of minimal representatives is not guaranteed by that theory. Indeed, the general agreement is that HHPB is more suited to capture concurrency, whereas the non-hereditary version deals better with causality. The latter equivalence is coarser, but still causality is informative enough to characterize key security properties, such as non-interference [4]. Moreover, the non-hereditary equivalence has better decidability properties than the hereditary one [14].

The main issue with causal operational models is that they often have infinitely many states, so model checking is unfeasible. This is indeed the case of behavior structures, where posets of states are enlarged at each transition, because a new event for the corresponding action is generated. Even if we minimize w.r.t. bisimilarity, there is no way of throwing away "useless" events or decreasing the size of posets.

In this paper we present an approach to obtain compact, and in many cases finite, operational models for causality in Petri nets. They will be presented in two "flavors": a set-theoretic and a categorical one, based on coalgebras [22,1]. In addition to the theoretical and practical interest of reconducting our problem to unifying and well studied models such as coalgebras, we emphasize that our coalgebraic model is simpler than the set theoretical one. In fact, even if deriving a naive set-theoretic model from a Petri net is not difficult, the technical development required to obtain a compact model is quite involved and requires some ingenuity. Instead, in a categorical setting, this machinery will become remarkably simpler and natural. Actually, in a precise sense, the construction of the compact model will be automatic, thus providing a mathematical justification of the otherwise ad hoc set-theoretic constructions.

1.1. Set-theoretic models

After some preliminaries on Petri nets and the presentation of a running example in Section 2, in Section 3 we model the behavior of a labeled Petri net as a *causal case graph* (CG). Recall that a case graph is a labeled transition graph where states are markings and transitions are steps, representing many firings happening simultaneously. In causal case graphs, transitions are single firings, and causal data are used to encode information about concurrency. More precisely (see Definition 3.3, where CGs are called "concrete" as opposed to "abstract" CGs, introduced later):

- states are of the form $O \rhd c$, where: O is a poset describing causal dependencies among a finite collection of events; c is a marking where each token is decorated with its *causes*, i.e. the set of events that led to its creation (included in O);
- the transition relation is written $\xrightarrow{K \vdash e_a}$, where: K is the set of most recent causes of tokens that enabled the firing; e is a *fresh* event, different from all those occurring in the source state; and a is the action label of the fired transition.

We define a notion of bisimilarity for CGs where causal information plays a key role: only states with the same causal dependencies among past events, namely the same poset, are compared. This fact is crucial for the equivalence with history preserving bisimilarity described in Section 4.

Another important aspect is that transitions draw fresh events from an infinite set of event names. For each firing, we have *infinitely many* transitions in the CG, one for each possible fresh event. In this way we implement *event generation* in the same way name generation is represented, e.g., in nominal calculi. This fact will be crucial for our categorical models.

We, then, derive three consecutive refinements of the CG, described in Table 1, each improving the CG on one aspect:

$\mathrm{CG}_{\mathtt{AC}}$ *(Definition 3.8):* the transition relation becomes *finitely branching*, because we don't distinguish between posets with the same structure. In fact, it is enough to generate one canonical event, instead of all possible ones, for each firing. Consequently, states contain canonical representatives of events and only the action label of the new event is recorded in the transition.

$\mathrm{CG}_{\mathtt{IC}}$ *(Definition 3.17):* removing all but immediate causes, and identifying isomorphic states, may significantly reduce the state space, and even make it finite.

$\mathrm{CG}_{\mathtt{ICS}}$ *(Definition 3.27):* we equip each state with a set of isomorphisms acting as the identity on the state. These isomorphisms must form a *symmetry*, i.e., a group of automorphisms, on the state's poset. Transitions are reduced