Axioms for Contextual Net Processes*

F. Gadducci¹ and U. Montanari²

¹ Technical University of Berlin, Fach. 13 Informatik, gfabio@cs.tu-berlin.de.
² University of Pisa, Dipartimento di Informatica, ugo@di.unipi.it.

Abstract. In the classical theory of Petri nets, a process is an operational description of the behaviour of a net, which takes into account the causal links between transitions in a sequence of firing steps. In the categorical framework developed in [19, 11], processes of a P/T net are modeled as arrows of a suitable monoidal category: In this paper we lay the basis of a similar characterization for contextual P/T nets, that is, P/T nets extended with read arcs, which allows a transition to check for the presence of a token in a place, without consuming it.

1 Introduction

Petri nets [24] are probably the best studied and most used model for concurrent systems: Their range of applications covers a wide spectrum, from their use as a specification tool to their analysis as a suitable semantical domain. A recent extension to the classical model concerns a class of nets where transitions are able to check for the presence of a token in a place without actually consuming it. While the possibility of sensing for both presence and absence of a token yields very expressive nets equipped also with inhibitory arcs [14, 7, 4, 5], in the paper we focus our attention to nets extended with read arcs only, generically referred to as contextual nets, which have a richer theory and refer to several well-tailored applications. In fact, important constructions on ordinary nets can be extended to nets with read arcs, like those concerning non-sequential processes [22, 28] and event structures [1]. Moreover, these nets naturally model read-write access to shared memory, where readers are allowed to progress in parallel, with applications to transaction serializability in databases [25, 10], concurrent constraint programming [21, 3], asynchronous systems [27] and process algebras [20].

* Research partly supported by the EC TMR Network GETGRATS (General Theory of Graph Transformation Systems) through the Technical University of Berlin and the University of Pisa; by the Office of Naval Information Research Contracts N00014-95-C-0225 and N00014-96-C-0114; by the National Science Foundation Grant CCR-9633363; by the U.S. Army Contract DABT63-96-C-0096 (DARPA); and by the Information Technology Promotion Agency, Japan, as part of the Industrial Science and Technology Frontier Program "New Models for Software Architecture" sponsored by NEDO (New Energy and Industrial Technology Development Organization). Research carried out in part while the second author was on leave at Computer Science Laboratory, SRI International, Menlo Park, USA, and visiting scholar at Stanford University.
The operational behaviour of Petri nets can be described either via firing sequences, or via non-sequential processes [13]. Even if tightly related, only the latter option fully exploits the ability of nets to describe concurrency. Processes are acyclic, deterministic safe nets whose transitions are occurrences of the transitions of the original net. A process thus defines a partial ordering of transition occurrences, and captures the abstract notion of concurrent computation, in the sense that all the firing sequences corresponding to linearizations of the partial ordering are considered equivalent. Most semantic and logic notions specifying the concurrent behavior of nets are based on the notion of process [23, 2].

Processes play an important role in the “Petri Nets are Monoids” approach to net theory [19, 11, 26]. In this approach, a net \( N \) is analogous to a signature \( \Sigma \), and the symmetric monoidal category \( \mathcal{P}(\mathcal{N}) \) associated to \( N \) is analogous to the cartesian category \( \mathcal{L}(\Sigma) \) of terms and substitutions freely generated by \( \Sigma \). As the (tuples of) terms in \( T_\Sigma(X) \) are the arrows of \( \mathcal{L}(\Sigma) \), the processes\(^1\) of \( N \) are the arrows of \( \mathcal{P}(\mathcal{N}) \). The construction of \( \mathcal{P}(\mathcal{N}) \) provides a concise, algebraic description of the concurrent operational semantics of P/T nets. Since \( \mathcal{P}(\mathcal{N}) \) can be finitely axiomatized [26], this construction provides a finite axiomatization of non-sequential processes. Moreover, the well-understood setting of monoidal categories allows for an easy comparison with related models, like linear logic [17].

The aim of this paper is to extend the above categorical approach to P/T nets with read arcs. Our results should enable a fully algebraic description and analysis of these nets and, as a consequence, of the concurrency paradigm based on shared memory they represent. To the best of our knowledge, the problem has been tackled only in [18]. The solution proposed there associates to a CP/T net \( N \) a monoidal category \( \mathcal{P}'(\mathcal{N}) \), where the monoid of objects is not freely generated from the class of places. Our solution is instead “more in line”, so to say, with the approach, since the only axioms are on arrows: The results for processes of ordinary P/T nets can then be lifted to contextual processes.

The technical development presented in the paper is based on equipping a symmetric strict monoidal category with a transformation consisting of certain arrows, called duplicators. They are reminiscent of arrows of the same name which are obtained in cartesian categories as pairings of two instances of an identity. However they do not form a natural transformation as duplicators in cartesian categories. Besides duplicators, gs-monoidal categories [8] are equipped with dischargers and they differ from cartesian categories just for missing the naturality axioms on duplicators and dischargers. The arrows of the gs-monoidal category freely generated by a signature \( \Sigma \) represent the term graphs [8] on \( \Sigma' \). Symmetric strict monoidal categories equipped with duplicators (without naturality) are called share categories; match-share, if both them and their opposite are share categories. Those are the categories where processes of contextual nets live: The main result of the paper states that the category of processes of a contextual P/T net \( N \) is characterized as an inductively generated subcategory of the match-share category \( \mathcal{CP}(\mathcal{N}) \) associated via a free construction to \( N \).

\(^1\) Actually, a slightly extended version of the processes presented in [13], called concatenable processes, is needed to guarantee the uniqueness of sequential composition.