Coreflective Concurrent Semantics for Single-Pushout Graph Grammars

Paolo Baldan\textsuperscript{1}, Andrea Corradini\textsuperscript{2}, Ugo Montanari\textsuperscript{2}, and Leila Ribeiro\textsuperscript{3}

\textsuperscript{1} Dipartimento di Informatica, Università Ca’ Foscari di Venezia, Italia
\textit{baldan@dsi.unive.it}

\textsuperscript{2} Dipartimento di Informatica, Università di Pisa, Italia
\{\textit{andrea,ugo}\}@\textit{di.unipi.it}

\textsuperscript{3} Instituto de Informática, Universidade Federal do Rio Grande do Sul, Brazil
\textit{leila@inf.ufrgs.br}

Abstract. The problem of extending to graph grammars the unfolding semantics originally developed by Winskel for (safe) Petri nets has been faced several times along the years, both for the single-pushout and double-pushout approaches, but only partial results were obtained. In this paper we fully extend Winskel’s approach to single-pushout grammars providing them with a categorical concurrent semantics expressed as a coreflection between the category of graph grammars and the category of prime algebraic domains.

Introduction

It belongs to the folklore that Graph Grammars [25] generalise Petri nets, in that they allow for a more structured representation of system states, modelled in terms of graphs rather than (multi)sets, and for a more general kind of state transformation, modelling also preservation of parts of the state, besides deletion and creation.

During the last years, a rich theory of concurrency for the algebraic approaches to graph transformation has been developed, including the generalisation of various classical Petri net concurrency models, like Goltz-Reisig process semantics [13] and Winskel’s unfolding semantics [27].

Recall that, building on [22], the seminal work [27] gives the concurrent semantics of (safe) nets by means of a chain of coreflections leading from the category of safe Petri nets to the category of prime algebraic domains.

\[
\text{Safe Nets} \xrightarrow{\perp_{\text{U}}} \text{Occurrence Nets} \xleftarrow{N_{\text{E}}} \text{Prime Event Structures} \xrightarrow{p_{\sim}} \text{Domains}
\]

The first step unfolds any (safe) net into an occurrence net, i.e., a branching acyclic net making explicit causality and conflict (nondeterministic choice point) between events in the net. The second step produces a prime event structure.

\* Research supported by the FET-GC Project IST-2001-32747 AGILE and by the MIUR Project COFIN 2001013518 CoMETA.
(PES) abstracting away the state and recording only the events and the relationships between events. Finally, the last step maps any PES into the corresponding prime algebraic domain of configurations.

Some important steps have been taken in the direction of developing an analogous semantical framework for algebraic graph grammars, but a definitive answer has not been provided yet. More precisely, a number of constructions have been defined for algebraic, double-pushout (DPO) graph grammars \([12,9]\) by the first three authors (see, e.g., \([1]\)), as summarised by the following diagram:

\[
\text{DPO Graph} \xleftrightarrow{\perp_{\text{ug}}} \text{Occurrence Grammars} \xrightarrow{\varepsilon_{\text{g}}} \text{Inhibitor Event Structures} \xleftarrow{\perp_{\text{L_i}}} \text{Domains}
\]

Even if at this level of abstraction it is not possible to see the relevant differences in the technical treatment of DPO grammars w.r.t. the much simpler case of Petri nets, still it is worth pointing at the evident differences between this chain of categories and the corresponding one for nets. Firstly, the category of PES’s is replaced by that of inhibitor event structures (IES’s), which, assuming conditional or-causality as a basic relation between events, are able to capture both the asymmetric conflicts between events arising from the capability of preserving part of the state and the inhibiting effects related to the presence of the application conditions for rules. The category of domains can be viewed as a coreflective subcategory of IES’s (as shown by the last step of the chain) and thus one can also recover a semantics for DPO grammars in terms of domains and PES’s. Secondly, the functor from the category of occurrence grammars to the category of IES’s does not admit a left adjoint establishing a coreflection between IES’s and occurrence grammars, and thus the whole semantic transformation is not expressed as a coreflection.

In this paper we concentrate on the single-pushout (SPO) approach \([18,11]\) to graph transformation. One of the main differences with respect to the DPO approach lies in the fact that there are no conditions on rule application, i.e., whenever a match is found the corresponding rule can always be applied. For SPO grammars an unfolding construction has been proposed in \([24]\), corresponding to the first step in the above chains of coreflections.

Building on the results briefly summarised above, we provide a coreflective unfolding semantics for SPO graph grammars, defined through the following chain of coreflections:

\[
\text{SPO Graph} \xleftrightarrow{\perp_{u_s}} \text{Occurrence Grammars} \xrightarrow{\varepsilon_{s}} \text{Asymmetric Event Structures} \xleftarrow{\perp_{L_a}} \text{Domains}
\]

In particular, this construction differs from and improves that for DPO graph grammars, discussed above, because of the following facts:

- Due to the absence of application conditions for rules, a less powerful and more manageable kind of event structures called asymmetric event structures (introduced to deal with contextual nets in \([4]\), can be used to represent the dependency structure of SPO graph grammars.