Electronic Notes in Theoretical Computer Science 50 No. 3 (2001) – Proc. GT-VMT 2001
URL: http://www.elsevier.nl/locate/entcs/volume50.html 7 pages

Modeling Petri Nets by Local Action Systems [1]

Nico Verlinden [2]

Department of Mathematics and Computer Science
Universitaire Instelling Antwerpen
Antwerp, Belgium

Dirk Janssens [3]

Department of Mathematics and Computer Science
Universitaire Instelling Antwerpen
Antwerp, Belgium

1 Introduction

Among the formal models for describing concurrent systems, Petri Nets are certainly among the most popular ones. They have an attractive visual presentation, and a rich theory has been developed for them [8,9]. Over the years a wide variety of variants have been described such as Place/Transition Nets [2], Elementary Nets [10], Colored Nets [6], Inhibitor Nets [7], etc. On the other hand, the theory of graph rewriting systems is another important model of computation where the configurations of a system are graphs, and hence suited for a visual representation. Several types of graph rewriting systems have been proposed, included Algebraic Graph Grammars [3] and Local Action Systems [5]. The theory of the latter is based on the notion of a process, a description of a system run where the nodes are the nodes of configurations and where the causal relation is represented by a partial order on these nodes.

The aim of the paper is to explore the encoding of Petri Nets by Local Action Systems rewriting discrete graphs, and in particular the effect of choosing different ways to encode the markings of a Petri Net. It is shown that one gets systems with the same sequential behavior as the net, but with different concurrent behaviors. It turns out that two particular choices (token and place representation) correspond to well-known classes of Petri Nets. Subsequently, an application is considered in which the concurrent behavior of a net

[1] Partially supported by the EC TMR Network GETGRATS (General Theory of Graph Transformation Systems) and Esprit Working Group APPLIGRAPH through Universitaire Instelling Antwerpen.
[2] Email: Nico.Verlinden@ua.ac.be
[3] Email: Dirk.Janssens@ua.ac.be

changes over time, and it is shown that this situation can be handled using the encoding presented.

Thus, while it is not surprising that Petri Nets can be modeled by Local Action Systems, there is evidence that the latter offer sufficient flexibility to serve as a unifying framework for some of the numerous variants of Petri Nets. Moreover, Local Action Systems in their own right offer a powerful language for describing concurrent systems.

2 Local Action Systems (LAS)

First a few basic ideas of LAS graph rewriting are recalled. Local Action Systems are a type of graph rewriting systems based on node rewriting and embedding, introduced in [5]. Only the very restricted case where discrete graphs are rewritten is considered in this paper.

In a LAS, it is assumed that the set of node labels forms a commutative monoid. A production π consists of

- a labeled, discrete graph l,
- a labeled, discrete graph r,
- for each pair (x, y), where x is a node of l and y is a node of r, an action $a_{x,y}$ on node labels.

In LAS locality of the actions means that both the input and the output of an action correspond to a node. An action affects a graph only locally.

To apply π to a graph g, one has (1) to find a subgraph l' of g that matches l, (2) replace the nodes of l' by new nodes, one for each node of r, and (3) compute the label of the new nodes. If the new node y' corresponds to node y of r, then its label is the sum of the label of y in r and all the labels $a_{xy}(lab_g(x'))$, where x is a node of l, x' is the node of g that corresponds to x, and $lab_g(x')$ is its label.

It is assumed that the set of node labels is equipped with a partial order $\leq$. A graph l matches a graph l' if there is a bijective function f from the set of nodes of l into the set of nodes of l' such that for each node x of l, $f(lab_l(x)) \leq lab_{l'}(x)$. Hence the notion of matching depends on the choice of $\leq$.

A Local Action System consists of a graph and a set of productions.

3 Encoding of Petri Nets

Marked Place/Transition Nets with arc-weights, shortly Marked Nets, are the first type of Petri Nets considered; it is well-known that various variants of Petri Nets can be modeled by Marked Nets. Fig. 1 depicts a Marked Net. In this abstract we restrict attention to the case where all weights are 1.

A state of a Marked Net is a marking, while in a graph rewriting system

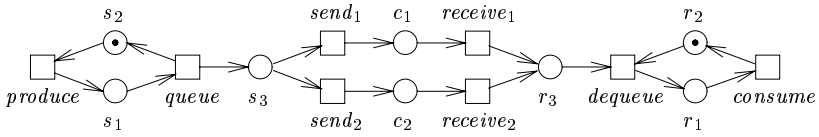

Fig. 1. A Marked Place/Transition Net.

a state is a graph. If one wants to model Marked Nets by a graph rewriting system, a graph representation is needed for each marking. The following three types of graph representations are considered.

- Each node of a graph represents one token of a marking.
- Each node of a graph represents a place.
- Each node of a graph represents a group of places.

Fig. 2 shows three different graph representations for the marking which assigns 1 token in place s_2, 2 tokens in place s_3, and 1 token in place r_2. Formally,

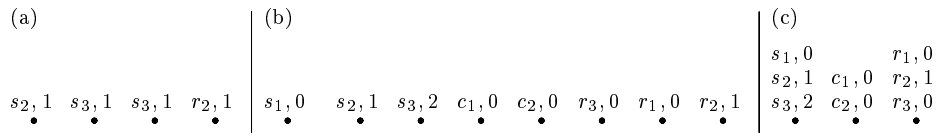

Fig. 2. Different graph represention for a marking.

the node labels are elements of $\Delta^v = [S \rightharpoonup \mathbb{Z}]$, i.e. the monoid of partial functions from the set of places S into $\mathbb{Z}$, where the domain of $f + g$ for 2 partial functions f and g is the union of the domains of f and g. In Fig. 2 (c) the set of places is divided into the groups $\{s_1, s_2, s_3\}$, $\{c_1, c_2\}$, and $\{r_1, r_2, r_3\}$. E.g., the label of the left most node in Fig. 2 (c) denotes the partial function with domain the places $\{s_1, s_2, s_3\}$, and mapping s_1 to 0, s_2 to 1, and s_3 to 2.

In this paper it is shown that different graph representations give rise to Local Action Systems such that their sequential behaviors are the same but their concurrent behaviors differ. In this way one gets different concurrent views of the same Marked Net by Local Action Systems.

All systems considered in this paper, Petri Nets as well as Local Action Systems, have a process semantics, and each process determines a set of sequentializations. Since in all cases the systems are either Petri Nets or LAS encodings of a Petri Net, these sequentializations can be viewed as occurrence sequences in the sense of [2]. Thus one may use the following notion of equivalence to compare systems, and in particular their concurrent behavior.

Definition 3.1

- *Let P and P' be processes. P and P' are equivalent if they determine the same set of occurrence sequences.*
- *Let N and N' be systems. N and N' are process-equivalent if, for each process P of N, there exists an equivalent process P' of N', and conversely, for each process P' of N', there exists an equivalent process P of N.*

The way these different representations of a marking lead to different Local Action Systems is now considered in more detail.

3.1 Token Representation

In the first representation each node of a graph exactly represents one token of a marking. This representation is perhaps the most straightforward and has already been studied, e.g. in [1]. A transition is modeled by a production such that the left-hand side is a graph representation of the tokens that are consumed by the transition, and the right-hand side is a graph representation of the tokens that are produced by the transition. Fig. 3 depicts the productions of three transitions of the Marked Net in Fig. 1. The dashed lines

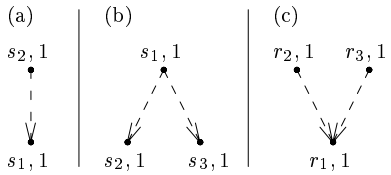

Fig. 3. Productions modeling the transitions (a) *produce*, (b) *queue*, and (c) *dequeue*.

depicts the causality relation. All local actions are $\mathbf{0}$, i.e. the function on Δ^v mapping every node label into the partial function in Δ^v with empty domain, and hence they are not depicted. It turns out that the Local Action System obtained in this way has the same sequential and concurrent behavior as the Marked Net it encodes. (see [4]).

Theorem 3.2 *Let N be a Marked Net and let N' be its (token-)encoding. Then N and N' are process-equivalent.*

3.2 Place Representation

In the second representation each place of a Marked Net corresponds to one node of a graph. Each transition changes the amount of tokens in several places and hence a production modeling a transition t has to rewrite each node that corresponds to either an input or an output place of t. In a first approach, equality is used for the partial order that determines the matching. Since the number of tokens in each place may vary, a transition is modeled by an infinite set of productions. Fig. 4 depicts a few productions modeling the transition *queue* of the Marked Net in Fig. 1. E.g, the production in Fig. 4(c)

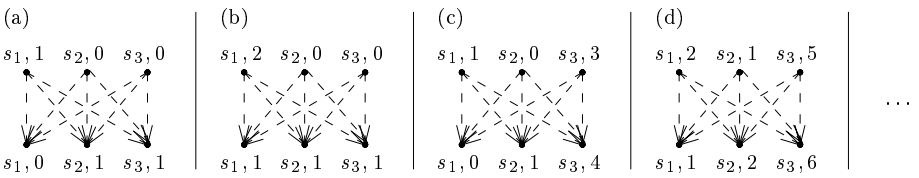

Fig. 4. An infinite number of productions modeling the transition *queue*.

is applicable to a graph modeling 1 token in s_1, no tokens in s_2, and 3 tokens in s_3. As a result the token of s_1 is taken away, and 1 token is added to both s_2 and s_3 yielding 1 token in s_2 and 4 tokens in s_3. Obviously, the Local Action System obtained in this way has an infinite set of productions. By introducing nonzero local actions and using another partial order to define the matching, one obtains a finite version of it. The infinite set of productions modeling the transition *queue* can be replaced by one production which is depicted in Fig. 5 (a). A dashed line of the causality relation is replaced by a solid line if the corresponding local action is not **0**. In the figure there are 3 solid lines labeled by the local action **1**, denoting the identity. As a consequence this production can be applied to any graph which models at least one token in s_1 (one chooses for the validity the relation $\leq$ on the set of partial functions from the places into $\mathbb{Z}$). As a result of the combination of the local actions with the right-hand side, one token of s_1 is taken away, and one token is added to both s_2 and s_3. One has the following theorem.

Theorem 3.3 *Let N be a Marked Net. Let N' be its infinite place-encoding, and let N'' be its finite place-encoding. Then N' and N'' are process-equivalent.*

With this encoding two productions can be applied concurrently only if they do not rewrite the same node, i.e. only if the places of the transitions they model do not overlap. Hence the places of the Marked Net can be seen as resources that can be accessed only by one transition at a time. This situation also occurs with Elementary Nets and hence contact-free Elementary Nets can be modeled in this way.

Theorem 3.4 *Let N be a contact-free Elementary Net system and let N' be its (finite or infinite) place-encoding. Then N and N' are process-equivalent.*

3.3 Representation of groups of places

To illustrate further the flexibility of the LAS mechanism a situation is considered where the concurrent behavior of the Marked Net of Fig. 1 is further restricted: it is assumed that the places are grouped together and that each group of places can be accessed by only 1 transition at a time. This situation arises when one has a certain number of memory blocks managed by processors and one assigns each memory object (place) to a memory block (processor). In this way a transition does not lock a place, but a whole block. In a situation where s_1, s_2 belong to the same group, but s_3 does not, the transition *queue* can be encoded by the production of Fig. 5 (b).

One may go one step further and model possible migrations of a place from one group to another. Then one may use for each place p a migration production (see Fig. 5 (c)). In the figure the local actions $\tilde{\boldsymbol{p}}$ and $\overline{\boldsymbol{p}}$ denote the restrictions of partial functions to $\{p\}$ and the complement of p, respectively. By applying such a production one changes the grouping of places (and hence the concurrent behavior).

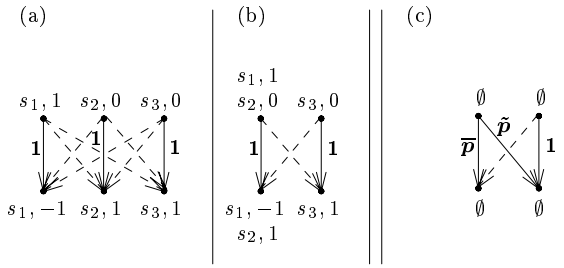

Fig. 5. Different grouping of places in (a) and (b), and migration of a place in (c).

Conclusion and Future Work

It is shown that Local Action Systems offer a flexible framework for modeling different kinds of Petri Nets and for modeling Petri Nets with different views of concurrency. Apart from considering other and possibly more complex variants of Petri Nets (e.g., Inhibitor Nets) it may be interesting to develop a theory of processes on a higher level, abstracting away from the concrete details of systems like Petri Nets or Local Action Systems. In this way one may get a framework in which the sequential and concurrent behavior of a large class of systems, including Petri Nets and various types of graph rewriting systems, can be investigated.

References

[1] A. Corradini, Concurrent Computing: From Petri Nets to Graph Grammars, In *Proceedings of the Joint COMPUGRAPH/SEMAGRAPH Workshop on Graph Rewriting and Computation (SEGRAGRA'95)*, Electronic Notes in Theoretical Computer Science, **Vol. 2**, Elsevier, 1995.

[2] J. Desel and W. Reisig, Place/Transition Petri Nets, In [9], 122-173.

[3] H. Ehrig, M. Korff, and M. Löwe, Tutorial Introduction to the Algebraic Approach of Graph Grammars based on Double and Single Pushouts. In *Proceedings of the 4th International Workshop on Graph-Grammars and Their Application to Computer Science*, Lecture Notes in Computer Science, **Vol. 532**, Springer-Verlag, Berlin, 1991, 24-37.

[4] U. Goltz and W. Reisig, Processes of Place/Transition - Nets, In *Automata, Languages and Programming*, Lecture Notes in Computer Science, **Vol. 154**, Springer-Verlag, Berlin, 1983, 264-277.

[5] D. Janssens, Actor Grammars and Local Actions, In *Handbook of Graph Grammars and Computing by Graph Transformation*, **Vol. 3**, World Scientific, 1999, 57-106.

[6] K. Jensen, An Introduction to the Theoretical Aspects of Coloured Petri nets, In *A Decade of Concurrency*, Lecture Notes in Computer Science, **Vol. 803**, Springer-Verlag, Berlin, 1994, 230-272.

[7] J. Kleijn and M. Koutny, Process Semantics of P/T-Nets with Inhibitor Arcs. In *Application and Theory of Petri Nets*, Lecture Notes in Computer Science, **Vol. 1825**, Springer-Verlag, Berlin, 2000, 261-281.

[8] W. Reisig, Petri Nets, *EATCS Monographs on Theoretical Computer Science*, **Vol. 4**, Springer-Verlag, Berlin, 1985.

[9] W. Reisig and G. Rozenberg, (eds.). Lectures on Petri Nets I: Basic Models. In *Advances in Petri Nets*, Lecture Notes in Computer Science, **Vol. 1491**, Springer-Verlag, Berlin, 1998.

[10] G. Rozenberg and J. Engelfriet, Elementary Net Systems, In [9], 12-121.