Flexible Independence of Net Transformations and Token Firing in the Cospan DPO Approach

Kathrin Hoffmann[1], Julia Padberg[1], Hartmut Ehrig[2], Frank Hermann[2]

Hochschule für Angewandte Wissenschaften Hamburg[1]
Technische Universität Berlin[2]

Abstract: The double pushout (DPO) approach for graph transformation has been applied already to Petri nets in order to model rule based transformations of the net structure. Recently, as alternative to the classical DPO approach, the cospan DPO approach has been proposed where rules are cospans instead of spans. Although the cospan DPO approach has been shown to be equivalent to the classical DPO approach, there are several advantages using the cospan DPO approach especially for Petri nets. Roughly spoken, in the classical DPO approach the intermediate net obtained by rule application can be full of holes like Swiss cheese, while in the cospan DPO approach this net includes the source net and the target net. Thus, on the one hand several properties can be formulated in a more intuitive way and on the other hand some aspects can be investigated that have escaped our attention in the classical DPO approach. In this paper we present main results of a line of research[1] concerning the independence of net transformations and token firing. In more detail we apply the cospan DPO approach to Petri nets and give not only sufficient but also necessary conditions for the execution of a transformation step and a firing step leading to the same result.

Keywords: Petri net, rule based transformation, independence, cospan double pushout approach

1. Introduction

In (Hoffmann at all, 2005) the concept of rule based transformations of place/transition (P/T) systems has been introduced that is most useful to model changes of the net structure while the system keeps running. Rule based transformations of P/T systems are inspired by graph transformation systems (Rozenberg at all, 1997), where the basic idea behind net transformations is the stepwise development of P/T systems by given rules. Think of these rules as replacement systems where the left hand side is replaced by the right hand side while preserving a context. In this way not only the follower marking of a P/T system can be computed by token firing but also the structure can be changed by rule application to obtain a new P/T system that is more appropriate with respect to some requirements of the environment. Moreover these activities can be interleaved.

Petri nets that can be changed, have become a significant topic in the recent years, as the adaption of a system to a changing environment gets more and more important. Application areas cover e.g. computer supported cooperative work, multi agent systems, dynamic process mining or mobile networks. Moreover, this approach increases the expressiveness of Petri nets and allows a formal description of dynamic changes.

[1] Formal modeling and analysis of flexible processes in mobile ad-hoc networks, funded by the German Research Council (DFG) (2006-2011), http://tfs.cs.tu-berlin.de/projekte/formalnet/.

While independence conditions for two firing steps of P/T systems are well known, independence of two net transformation steps are closely related to local Church-Rosser properties for graph transformations (Rozenberg et all 1997), that are valid in the case of parallel and sequential independence of rule based transformations. In (Ehrig, 2006) conditions for two transformation steps are given in the framework of high-level replacement systems with applications to net transformations, so that these transformation steps applied to the same P/T system can be executed in arbitrary order, leading to the same result.

But the question arises under which conditions a net transformation step and a firing step are independent. This problem is partially solved in (Ehrig at all, 2007) by analyzing under which sufficient conditions net transformation and token firing are parallel, sequentially and coparallel independent. In more detail, we assume that a given P/T system represents a certain system state. The next evolution step can be obtained not only by token firing but also by rule application. Hence provided that certain conditions are satisfied each of these evolution steps can be postponed after the realization of the other, yielding the same result and, analogously, they can be performed in a different order without changing the result.

In this paper we continue our work by analyzing under which necessary conditions a firing step is independent of a transformation step. In contrast to (Ehrig at all, 2007) we use the cospan double pushout (DPO) approach (Ehrig et all, 2009) instead of the classical DPO approach (Rozenberg, 1997). The advantage is that the notions of independence can be formulated in a more intuitive way.

2. Net Transformations in the Cospan DPO Approach

In the classical DPO approach (Rozenberg, 1997) a rule is given by a span of morphisms and a transformation step via a rule is constructed by two pushouts. The rule specifies the items that are deleted in a first step and the new items added in a second step. From the implementation point of view, it is often more convenient to add the new items first and then delete some of the old items. This idea is adopted in the cospan DPO approach where a rule is given by a cospan of morphisms, while a transformation step via a cospan rule is still defined by two pushouts (Ehrig et all, 2009).

Due to the equivalence of these two approaches each concept and result of the classical DPO approach can be transferred to an adjoint concept and result in the cospan DPO approach, and vice versa. But especially for Petri nets the cospan DPO approach is more appropriate, because the relation between the items added resp. deleted by rule application is precisely stated in the cospan of morphisms. In the classical DPO approach these property preserving transformation have often to be formulated by additional morphisms (Urbasek, 2003).

For place/transition (P/T) systems we use the algebraic notion as in (Meseguer et all, 1990). A P/T net $PN=(P,T,pre,post)$ is given by the set of places P, the set of transitions T, and two mappings $pre,post\colon T\rightarrow P^{\oplus}$, the pre domain and the post domain, where $P^{\oplus}$ is the free commutative monoid over P. A P/T system (PN,M) is given by a P/T net PN with an initial marking $M\in P^{\oplus}$. If a transition is enabled for a marking M the follower marking M' is computed and $(PN,M)\rightarrow(PN,M')$ is called a firing step. In the figure at the end of Section 3 two firing steps are exemplarily depicted: $(PN1,M1) \dashrightarrow (PN1,M1')$ via the transition *t1* and $(PN2,M2) \dashrightarrow (PN2,M2')$ via the transition *t2*.

In order to define rules and net transformations we use simple homomorphisms that are generated over the set of places. These P/T morphisms map places to places and transitions to transitions preserving the pre domain as well as the post domain of a transition. Additionally they require that the initial marking at corresponding places is increasing or even stronger.

$$
\begin{tikzcd}
(L, M_L) \arrow[r, "l"] \arrow[d, "m"'] \arrow[dr, phantom, "(1)"] & (K, M_K) \arrow[d, "k"] \arrow[dr, phantom, "(2)"] & (R, M_R) \arrow[l, "r"'] \arrow[d, "n"] \\
(PN_1, M_1) \arrow[r, "f"'] & (PN_0, M_0) & (PN_2, M_2) \arrow[l, "g"]
\end{tikzcd}
$$

An application of a rule is called a transformation step and describes how an object is actually changed by the rule. A rule $prod=((L,M_L) \rightarrow (K,M_K) \leftarrow (R,M_R))$ in the cospan DPO approach is given by three P/T systems called left hand side, interface and right hand side, respectively, and a cospan of two P/T morphisms *l* and *r*. We additionally need a match morphism $m: (L,M_L) \rightarrow (PN_1,M_1)$ that identifies the relevant parts of the left hand side in a given P/T system. Then a transformation step $(PN_1,M_1) \Rightarrow (PN_2,M_2)$ via a cospan rule *prod* can be constructed in two steps. In a first step we glue together the P/T system (PN_1,M_1) and the interface of the cospan rule along the left hand side leading to an intermediate P/T system. In a second step we delete those elements from the intermediate P/T system which are not preserved by the right hand side of the cospan rule resulting in the new P/T system (PN_2,M_2). Thus a transformation step consists of the pushout diagrams (1) and (2) depicted in the diagram above.

The cospan DPO approach does not allow the treatment of unmatched transitions as well as markings at places which should be deleted. In this case the so called cospan gluing condition forbids the application of rules. Furthermore items which are identified by a non injective match must be preserved by rule applications. Note that a positive check of the cospan gluing condition makes sure that the resulting P/T system is well defined.

The example in Section 3 illustrates two different transformation steps: on the one hand the transformation step $(PN1,M1) \Rightarrow (PN2,M2)$ in the two upper rows and on the other hand the transformation step $(PN1',M1') \Rightarrow (PN2',M2')$ given in the upper and the lower row of the figure. Note, that in both transformation steps the same rule in the upper row is applied.

3. Flexible Independence of Net Transformations and Token Firing

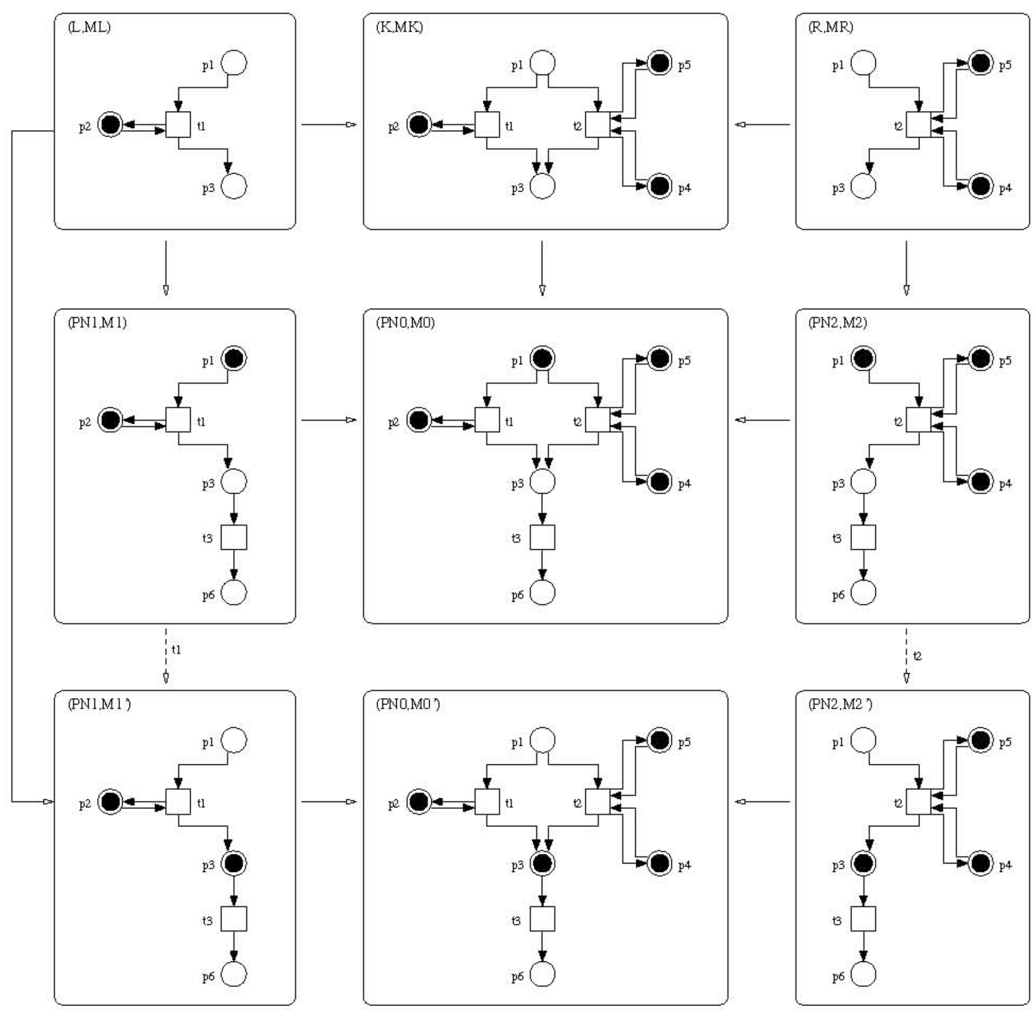
Flexible parallel independence allows the execution of a transformation step and a firing step in arbitrary order leading to the same result. A transformation step $(PN_1,M_1) \Rightarrow (PN_2,M_2)$ via a cospan rule *prod* and a firing step $(PN_1,M_1) \dashrightarrow (PN_1,M_1')$ via a transition $t_1 \in T_1$ are called flexible parallel independent if and only if there is a transition $t_2 \in T_2$ such that (1) the transition t_2 is M_2-enabled, (2) the pre- and post domain of transition t_1 is equal to the pre- and post domain of transition t_2, and (3) the rule *prod* can be applied to the P/T system (PN_1,M_1'). These conditions are not only sufficient but also necessary to ensure each of these steps can be postponed after the realization of the other without changing the result.

Given two flexible independent steps $(PN_1,M_1) \Rightarrow (PN_2,M_2)$ via a cospan rule *prod* and $(PN_1,M_1) \dashrightarrow (PN_1,M_1')$ via a transition $t_1 \in T_1$ then there is a firing step $(PN_2,M_2) \dashrightarrow (PN_2,M_2')$ via a transition $t_2 \in T_2$ as well as a transformation step *$(PN_1,M_1') \Rightarrow (PN_2,M_2')$* via the cospan rule *prod* with the same marking M_2'.

$$\begin{tikzcd}
 & (PN_1, M_1) \arrow[dl, Rightarrow, "(prod,m)"'] \arrow[dr, dashed, "t_1"] & \\
(PN_2, M_2) \arrow[dr, dashed, "t_2"'] & & (PN_1, M_1') \arrow[dl, Rightarrow, "(prod,m')"] \\
 & (PN_2, M_2') &
\end{tikzcd}$$

In the diagram on the left hand side we have required that the upper pair of steps is flexible parallel independent leading to the lower pair of steps. Furthermore we can consider the situations where the left, right or lower pair of steps are given together with suitable notions of sequential resp. coparallel independence such that the right, left and upper pair of steps exist leading to the same result. Detailed notions and proofs can be found in (Hoffmann et all, 2010).

For example the transformation step *(PN1,M1)⇒(PN2,M2)* in the two upper rows in the following figure and the firing step *(PN1,M1)⇢(PN1,M1')* in the lower part of the left column are flexible parallel independent, because the conditions (1) - (3) are satisfied. Thus these two steps can be realized in any order leading to the same result *(PN2,M2')*.



4. Conclusion and Future Work

In (Ehrig at all, 2007) sufficient conditions for parallel independence are introduced based on the classical DPO approach, so that a transformation step and a firing step applied to the same P/T system can be executed in arbitrary order, leading to the same result. In contrast in this paper not only sufficient but also necessary conditions are formulated in the cospan DPO approach leading to the notion of flexible parallel independence. The main difference is that parallel independence requires the preservation of transitions involved in the firing steps, while flexible parallel independence allows the replacement of transitions with equivalent environments. Thus, the notion of parallel independence is a special case of flexible parallel independence.

In future work we will consider further property preserving net transformations in the cospan DPO approach. Of special interest for workflow nets are strong connectivity and liveness. We expect that especially these properties could be formulated in a more intuitive way using the cospan DPO approach. In the example in Section 3, not only the source net *(PN1,M1)* and the target net *(PN2,M2)* but also the intermediate net *(PN0,M0)* are strongly connected in the sense of workflow nets. This is different to the classical DPO approach, where the intermediate net would consist of the transition *t3* and places only and therefore fails to be strongly connected.

References

H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer (2006). *Fundamentals of Algebraic Graph Transformation*. EATCS Monographs in Theoretical Computer Science. Springer, 2006.

H. Ehrig, F. Hermann, U. Prange (2009). *Cospan DPO Approach: An Alternative for DPO Graph Transformations*, Bulletin of the EATCS, 98:139–149. 2009

H. Ehrig, K. Hoffmann, J. Padberg, U. Prange, C. Ermel (2007). *Independence of Net Transformations and Token Firing in Reconfigurable Place/Transition Systems*. Proc. Application and Theory of Petri Nets and Other Models of Concurrency (ATPN), LNCS 4546, pages 104-123. Springer, 2007.

K. Hoffmann, H. Ehrig, and T. Mossakowski (2005). *High-Level Nets with Nets and Rules as Tokens.* Proc. Application and Theory of Petri Nets and Other Models of Concurrency (ATPN), LNCS 3536, pages 268-288. Springer, 2005.

K. Hoffmann, J. Padberg, H. Ehrig, F. Hermann (2010). *Property Preserving Net Transformations in the Cospan DPO Approach*. Technical Report TU Berlin, 2010.

J. Meseguer, U. Montanari (1990). *Petri Nets are Monoids*. Information and Computation, 88(2):105-155, 1990.

M. Urbasek (2003). *Categorical Net Transformations for Petri Net Technology*. Logos Verlag Berlin, 2003.

G. Rozenberg (1997). *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*. World Scientific, 1997.