Extension of the PESIM simulation tool

Milan Urbášek[*]
urbasek@dcse.fee.vutbr.cz

Milan Češka[**]
ceska@dcse.fee.vutbr.cz

Abstract

The paper describes the use of the particular transformations between Petri nets, called reductions, in the analysis process for Petri nets. Reduction methods reduce the graph of the net by agglomeration of place nodes and transition nodes. The reductions transform one net into another, preserving some dynamical properties of Petri nets such as liveness, boundedness or invariants. Then the standard, computer-assisted analysis methods can be applied.

1. Introduction

Petri nets were introduced by C. A. Petri in 1962. Since their introduction, Petri nets have been used to model many different systems and various classes of net have been developed. The standard class is the class of P/T Petri nets, but there are classes of coloured, timed and probabilistic net introduced by many authors as an extensions or restrictions of the standard class. The growing importance of Petri nets is related to their usefulness in modelling concurrent systems, distributed processing, communicating systems, all of whose dynamical behaviour can be easily expressed as movement of tokens in Petri nets.

2. Analysis methods

Petri nets can model a real system and provide an abstraction of the behaviour of the system without the encumbrance of irrelevant system detail. Then an analysis of the behaviour of the Petri net model will hopefully lead to important insights into the behaviour of the modelled system. Several techniques have been developed for the analysis of Petri nets. The basic method for investigating of the properties of Petri nets is using the state graph or coverability tree. The methods begin by constructing the state graph, which represents the dynamical behaviour of the Petri net as a transition system, in general infinite. The powerful and elegant analytical techniques for state graphs of Petri nets are based on linear algebra. These methods include the investigation of invariants, which reflect particular aspects of the repetitive behaviour of the net. Also included are methods based on net reduction which are described later in detail.

3. Simulation tools

The analysis methods can be applied algorithmically and a lot of systems deal with a analysis and simulation of Petri nets. The very important feature of Petri nets is the possibility of graphical expression of the model. This allows easier and intuitive comprehension and the

[*] Department of Computer Science and Engineering, FEECS, Technical University (VUT) Brno, Božetěchova 2, 612 66 Brno – Královo Pole

[**] Doc., RNDr., CSc., Department of Computer Science and Engineering, FEECS, Technical University (VUT) Brno, Božetěchova 2, 612 66 Brno – Královo Pole

graphic outline is, of course, more suitable for the presentation. This is nice feature, but in general, the development of the graphical editor and suitable graphical user interface is not a trivial problem. One of the implemented systems for simulation and analysis oriented on Petri nets is PESIM system. The PESIM system was developed at the Department of Computer Science and Engineering, FEECS, TU Brno. It is expected the integration of the reduction methods to this tool and enriching the analysis power of PESIM system.

4. Development of the PESIM system

In 1990-92, at the Department of Computer Science and Engineering, FEECS, Technical University of Brno the tool for working with Petri nets, called PESIM, was developed. The system is implemented in the environment of the operating system Windows 3.1.

The tool was designed in the C language using the object extension of the language. The present version was compiled to executable code using Borland C++, version 2.0.

System has got an integrated editor for designing the Petri nets. In its present implementation the system deals with initialized Petri nets, and the main methods for investigating such nets are implemented. The system allows one to work with two main classes of Petri nets: P/T Petri nets or stochastic Petri nets. It also deals with several restrictions and extensions of P/T Petri net class, such as marked graphs, free-choice nets, P/T Petri nets with inhibitor arcs and with capacity function of places.

After designing a Petri net using editor, simulation and analysis is possible. Simulation is interactive, allowing one to choose which of the enabled transitions fires and then follow the evolving of the behavior.

The analysis features allows investigating of the main properties of the net. There are implemented methods based on the reachability tree and invariants to study liveness, boundedness, safeness and conservativeness of the Petri net. For the stochastic nets, the analysis of the probabilistic features of the system is possible. The detailed description of the PESIM system can be found in [3] and [4].

5. Reduction methods

Although the construction of the state graph is undeniably an efficient method for finding the properties of small-sized Petri nets, it can be a lengthy process for a Petri nets which have a large number of reachable markings or states. The use of linear algebra also becomes difficult when the size of the Petri net increases.

From this reason, it can be advantageous before starting the analysis of the net to reduce the size of Petri net in order to obtain a net with a smaller number of places and transitions. Of course the properties of the initial net have to be preserved. In such case, the standard, computer-assisted methods for analysis of Petri nets can be applied more easily.

Reduction methods enable a Petri net to be transformed into another simpler Petri net, while maintaining certain properties of the initial net. The simplified net reflects the behaviour of the original net and preserves some properties, but is not equivalent to the original net. If the original net had an interpretation in a system modelled by the net, this interpretation need not be applicable to the simplified net.

We need to appreciate, however, that any given reduction method is not universally useful: it is not possible to obtain small net and preserve all the properties of initial Petri net. This means that reduction methods are divided into subclasses according to the properties which are preserved by the reductions. We shall present two main sublasses. The first is made up of methods preserving liveness and boundedness. The other class considered here contains reductions preserving invariants. We present reductions mentioned in [2].

5.1. Reduction methods preserving liveness and boundedness

The properties appearing in this section have been proved by G. Berthelot (See [6]). The reductions are described for ordinary Petri nets, this means for Petri nets which have the weight of all the arcs equal to one. The use for such nets is usually sufficient. They can be applied to other classes of Petri nets when some adaptation is made.

The reduction methods described in the literature for preserving of liveness are:

- R_1 – substitution of a place
- R_2 – suppressing an implicit place
- R_3 – suppressing a neutral transition
- R_4 – suppressing one of two identical transitions.

All these methods reduce the graph of the net, omitting some irrelevant information about the net. The main goal is to make the Petri net graph as small as possible and preserve the liveness and boundedness features as mentioned before. The applicability of every method depends on the certain set of conditions to be satisfied before the reduction may start.

The methods are used iteratively, so after applying one reduction, another one may be applied on the result of previous one. When no reduction is possible to be used the reduction process ends.

Figure 1 shows using the reduction on the a bit complicated net. The performing of reduction rules is illustrated by arrows with labels of reduction, which is performed.

We can see that the reduced net is *bounded*, *safe*, *live*, *deadlock-free* and with a *home-state*. Thus, all these properties are valid for the original net.

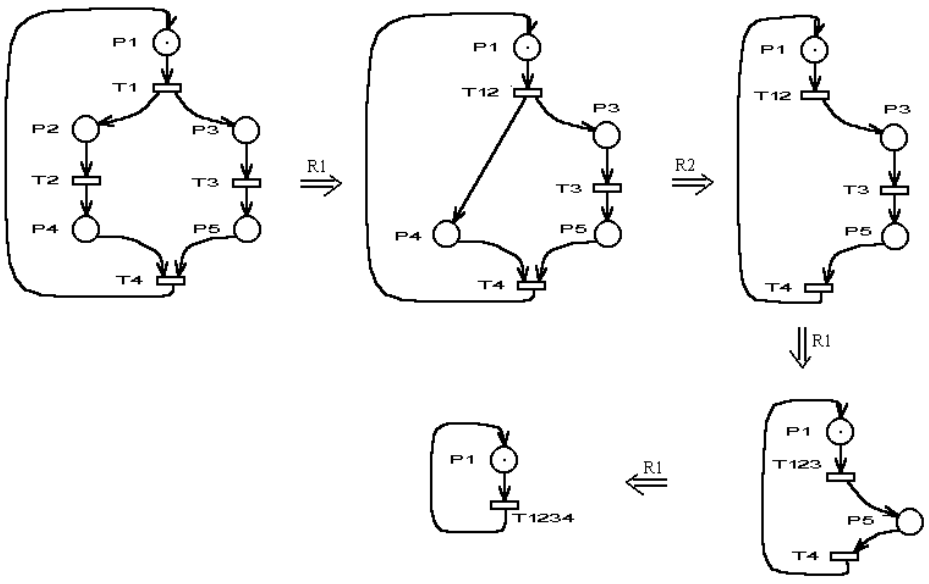

Figure 1 - Reduction example (liveness & boundedness)

Certain configuration such as the presence of a neutral transition or of identical transitions or even of an implicit place, are rarely encountered in a well designed Petri nets. They may, however, appear after the reduction process has begun and must be removed iteratively.

5.2 Reduction methods preserving invariants

There are two main methods for reducing, preserving invariants. There are applied to:

- R_a – pure transitions (without any self-loops) in the net,
- R_b – self-loop transitions.

What the application of the reductions looks like is shown in the following, where reductions preserving invariants are applied.

As shown, reductions are repeated, until no further reduction can be applied. The result is a simplified net, whose properties can be investigated using any of standard methods.

In general the result of two reduction steps depends on the order in which the steps are applied, but every time the preserved properties are the same for both orders of application.

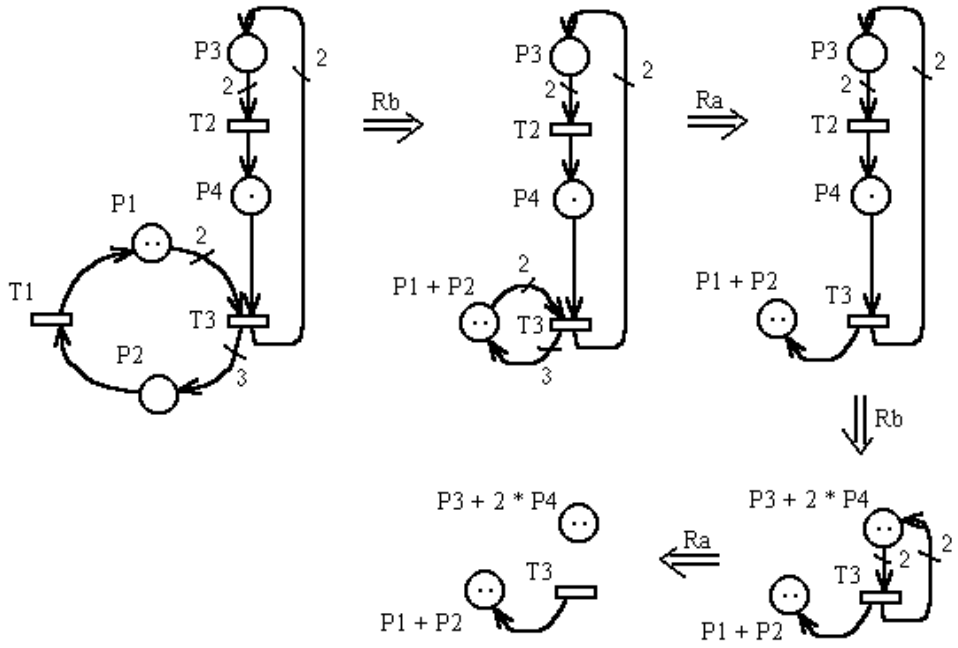

Figure 2 – Reduction example (invariants)

The reductions R_a and R_b preserve P-invariants and are useful for calculating these. The T-invariants of the Petri net could be obtained as P-invariants of the dual net, which has a set of places as a set of transitions and vice versa and all arcs have an opposite direction.

5.3 Algorithm for searching invariants

Reductions preserving invariants have an interesting property. Their application yields the invariants without using any other additional method. Their use is straightforward when the Petri net is described with using a change matrix and a algorithm for such use is available. The algorithm is described on the next page.

Algorithm 1 - Searching for invariants

Step 1. Let A be the dimension n unit matrix (n = number of places of investigated Petri net) and $B = \underline{N}$ be the change matrix of investigated net. Construct matrix $[A \mid B]$.

Step 2. For each index j of transition T_j.

Step 2.1. Add to matrix $[A \mid B]$ as many rows i as there are linear combinations of two rows, with positive integer coefficients, such that element (i, j) is zero.

Step 2.2. Eliminate from matrix $[A \mid B]$ the rows i whose element (i, j) is not zero.

Step 3. The P-invariants correspond to the non-zero rows of A.

A small example of the algorithm for searching invariants

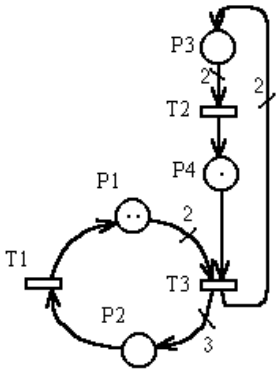

Figure 3 – Petri net and its change matrix

				T_1	T_2	T_3	
1	0	0	0	1	0	-2	P_1
0	1	0	0	1	0	+3	P_2
0	0	1	0	0	-2	+2	P_3
0	0	0	1	0	1	-1	P_4
1	1	0	0	0	0	1	P_1+P_2
0	0	1	2	0	0	0	P_3+2P_4

Step 1. The matrix of four rows (P_1 to P_4) and seven columns is constructed.

Step 2.

Step 2.1 for T_1: the row $P_1 + P_2$ is added which is the sum of the first two rows.
Step 2.2 for T_1: rows P_1 and P_2 are suppressed.
Step 2.1 for T_2: the row $P_3 + 2*P_4$ is added (weight 1 for P_3 and weight 2 for P_4).
Step 2.2 for T_2: rows P_3 and P_4 are suppressed.
Step 2.1 for T_3: no rows are added.
Step 2.2 for T_3: row $P_1 + P_2$ is suppressed.

Step 3. Only row $P_3 + 2*P_4$ remains, which indicates that there is a P-invariant $M_i(P_3) + 2 * M_i(P_4)$.

In this example just one invariant exists for the Petri net. In the case of bigger nets, many invariants can be obtained. It is necessary to get to know, that the result depends on the reduction order. Sometimes it is possible to obtain even non-basic components of invariants, even some invariants can be obtained several times.

If we would like to use results of this algorithm later in the analysis process, it is necessary to check the consistency of obtained invariants to get only relevant ones, without any duplicity.

6. Prototyped implementation of reduction methods

As a part of the project the prototype of simple programs that used the algorithms for net reductions has been implemented. This prototype implementation is determined for carrying out the implementation and verifying function of algorithms in order to find any difficulties or any other problems. In the future it is intended this module of analysis be integrated into the PESIM tool.

The prototype does not use the graphical interface, so it was necessary to choose another form of the Petri net representation suitable for input and saving. For such representation the simple language was proposed that allows to define all necessary features for the class of P/T Petri nets.

For an internal representation of a net in a memory the arrays were used, similar ones to these used in PESIM system. The transfer of the algorithms to the PESIM system will be then easier in the future.

The input module of all programs is a simple compiler of the input language, which checks the language syntax and the semantics, performing the translation of the language to the internal data structures used by the other modules. The next step of the analysis is the application of reductions and then the presentation of results.

7. Conclusion

Reduction methods provide a means of reducing large Petri nets to a point where computer-assisted analysis methods developed for smaller Petri nets can be applied. Several types of reductions can be used for investigating basic properties of Petri nets – liveness & boundedness and invariants. Using of reduction methods precedes the application of standard methods. Standard methods are applied to simplified nets, so the analysis results are obtainable more easily and quickly.

Application of reduction methods is algorithmic, and all methods can be programmed. This allows inclusion of the reduction methods into tools for Petri net simulation and analysis, as e.g. in the PESIM system, developed in the Department of Computer Science and Engineering, Technical University of Brno.

Acknowledgements

This work was supported by GAČR under the contract No. 102/98/0552 „Research and Applications of Heterogenous Models“.

References

[1] Češka M.: Petriho sítě - Základy teorie a nástroje pro jejich aplikaci, Akademické nakladatelství CRM, Brno 1994.

[2] David R., Alla H.: *Petri Nets and Grafcet*, Cambridge Press, 1992.

[3] Skácel M.: *Implementace prostředků pro práci s Petriho sítěmi pod MS Windows 3.0*, diploma project, DCSE FEECS, TU Brno, 1992.

[4] Karásek P.: *Podsystém pro práci se stochastickými Petriho sítěmi*, diploma project, DCSE FEECS, TU Brno.

[5] Češka M., Skácel M.: *The Petri Net Tool PESIM and Its Application in VLSI and ASIC Design*, In: Proc. of The 7th Microcomputer School VLSI and ASIC Design, Wydawnictwo Prac Naukowych FORMAT, Wroclaw 1995, pp. 185-194

[6] Berthelot G.: *Transformation et analyse de réseaux de Petri: application aux protocoles*, These d'Etat, Nice University, 1983.