Improving Reachability Analysis for Partially Symmetric High Level Petri Nets

Soheib Baarir, Jean-Michel Ilié, Alexandre Duret-Lutz
(Souheib.Baarir,Jean-Michel.Ilie, Alexandre.Duret-Lutz@lip6.fr)
LIP6, UMR CNRS 7606, Université PARIS VI, 8 rue du Capitaine Scott,75015 PARIS, FRANCE

Introduction

The coloured Petri nets formalism is an expressive model extending the representation of concurrency by Petri nets with a data management via coloured domains and functions. However this expressiveness leads in practice to huge state graphs considerably restricting their use for the reachability and the model checking problems.

Therefore, a recurrent research topic is the building of a reduced graph equivalent to the original one w.r.t. some set of properties. Among the proposed approaches, the symmetry based method builds a symbolic reachability graph (SRG) where a node corresponds to a set of states leading to an equivalent behaviour up to some "admissible" colour permutation. In order to be applicable, such a method must detect the admissible permutations by a syntactical examination of the net. This requirement has motivated the introduction of the well-formed nets model which is expressively equivalent to the coloured Petri net model but with a restricted syntax allowing the automatic computation of the SRG (Chiola *et al.* 1993). On this reduced graph, one solves the reachability problem.

However, the above approach suffers from a major limitation. It is well-known that without process identities, many distributed problems do not have solutions. Indeed in distributed algorithms, identity comparisons break deadlock situations. Modelling such algorithms produces nets whose behaviour is symmetric with the exception of a small set of transitions. Symmetry-based methods are not able to efficiently handle partial symmetries since they require a symmetry upon the whole model.

Here, we present the design and evaluation of a method for partially symmetric systems expressed with well-formed nets. It concentrates on the reachability problem and refines the concepts presented in (Haddad *et al.* 2000) and (Capra *et al.* 1999).

Description of the DSRG method

Our approach may be summarized as follows: (1) The asymmetric system is modelled as the synchronized product of a symmetrical model SYM, which represents the potential behaviour of the system but does not abstract some situations which are actually prohibited, and a control automaton CTRL that restricts the former potential behaviour up to obtain the real one; (2) a compact structure called Dynamical SRG (DSRG) is built, in particular by means of original symbolic operations like the symbolic refinement, grouping and inclusion test.

The modelling stage

In the standard WN modelling approach, the control policy is strongly coupled with the nominal description of the system. This often leads to complex nets hard to read and analyze.

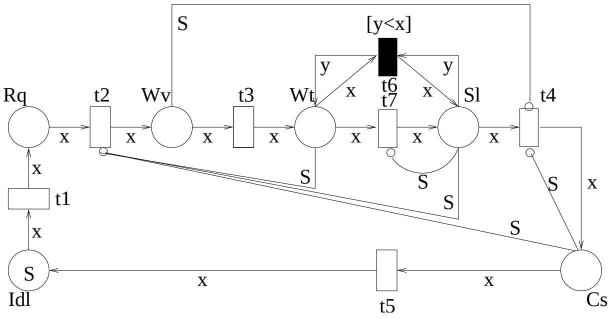


Figure 1: Modelling of a distributed algorithm with priorities for a critical section

In Figure 1, a distributed critical section algorithm is modelled in WN. The critical section accesses correspond to the firings of the t_4 transition, however, to bypass possible conflict situations, a control policy is modelled by several elements : the Sl place, the t_6 transition (immediate then privileged) and the t_7 transition. Moreover, the selection of a colour in the Sl place is based on the identities of colours due to guard $[x < y]$ attached to the t_6 transition (This guard ensures that the highest colour will be put in Sl, among the candidates). Beyond the modelling difficulty, the analysis problem comes from the fact that in WN, the former guard does not allow any colour permutation, so none set of markings could be considered as symmetric!

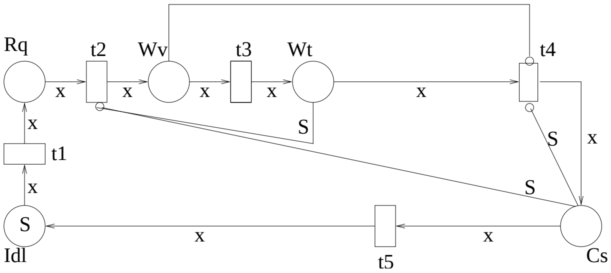


Figure 2: A symmetrical WN (SYM)for the DCS algorithm

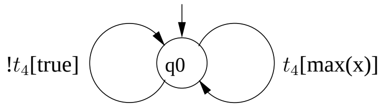


Figure 3: A control automaton (CTRL)

By decoupling the control policy and representing it by means of an *external component*, the WN modelling is reduced to the nominal behaviour of the system. As we can see in Figure 2, the net is less complex than the one of Figure 1 and there is no more distinction between the colours (We say that the *net is symmetric*). The aim of the external component is to obtain the real behaviour of the system, once composed with the symmetrical WN. In this paper, we propose to use an event-based automaton, namely control automaton, to control the firings of the transitions by means of synchronization operations. Each arc of the automaton is labelled by a boolean expression made of atomic propositions being predicates controlling the WN actions. In Figure 3, the arc labelled by $t_4[max(x)]$ means that among a given set of possible events for the t_4 transition, only the event that corresponds to the highest value for the variable x is allowed.

The symbolic representation

Let us recall that in standard WN, each colour class is partition in subsets called static-subclasses, such that the colours within each subset can be permuted.

The symbolic reachability graph (SRG) lies on a compact representation for a set of equivalent ordinary markings, called a symbolic marking (noted $\widehat{m}$). To specify a symbolic marking, each static-subclass is divided into dynamic subclasses (further, the function that associates to each dynamic subclass the corresponding static subclass is denoted d). A dynamic subclass is only specified by its size (cardinality), thus each consistent choice of colours for the dynamical subclasses leads to an ordinary marking. The colours implicitly represented by a dynamic subclass are assumed to be *in the same state*. Consequently, the marking of the places in the net is defined w.r.t. to dynamical subclasses instead of colours.

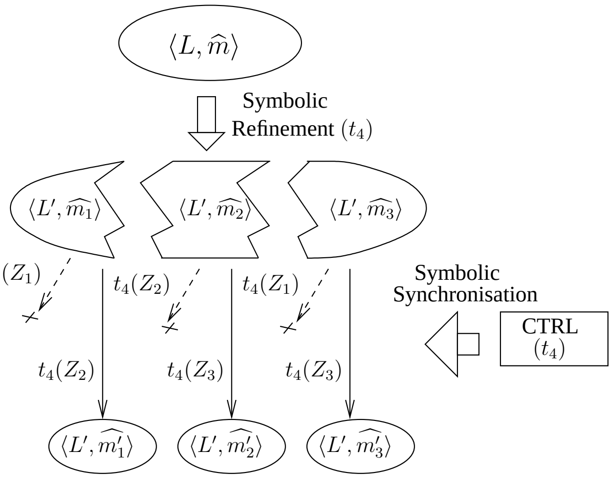
For instance, assume that the WN of Figure 2 allows to reach the following three markings: $idle(\langle 1\rangle), Wt(\langle 2\rangle + \langle 3\rangle)$ or $idle(\langle 2\rangle), Wt(\langle 1\rangle+\langle 3\rangle)$ or $idle(\langle 3\rangle), Wt(\langle 1\rangle+\langle 2\rangle)$. Each one corresponds to the situation where one process is in its idle state and the two others are waiting, attempting to access the critical section. If we consider that all the colours of the net can always be permutable, then there is no need to partition the colour class of the net (the WN is symmetric) and these three markings can be represented compactly by the symbolic marking : $idle(\langle Z_1\rangle), Wt(\langle Z_2\rangle)$ where Z_1 and Z_2 are dynamic subclasses s.t. $d(Z_1) = d(Z_2) = \{1, 2, 3\}$, and $|Z_1| = 1$, $|Z_2| = 2$ meaning that there is one process in its idle state and two waiting processes.

In our context, we reuse the notion of symbolic marking but the partition of colour classes is no more defined statically. Actually, the colour permutations that are declared available must accord with both structures, the WN and the control automaton. Our aim is to reevaluate them at each synchronization operation in order to obtain for each symbolic representation the roughest colour partition, thus enhancing the possibility of marking symmetries. A symbolic marking is now a pair $\langle L, \widehat{m}\rangle$ where L represents a *local partition* of colours used to build the symbolic representation $\widehat{m}$. We will see in the next section how colour partitions are evaluated in practise.

Computing the symbolic successors

One must define *a symbolic synchronization operation* between a symbolic firing of the (SYM) WN of Figure 2 and an arc of the CTRL automaton.

The first stage is to define a common set of symmetries between the symbolic marking and the predicate associated to the considered arc in the automaton. The symmetry of a predicate corresponds to the existence of symmetric atomic propositions within it. In our case, this is represented by a partition of colours. For instance, predicate "$x > 2$ and $x \leq 4$" over 5 colours, produces a colour partition $\{\{1, 2\}, \{3, 4\}, \{5\}\}$, predicate "$x < y$" as well as predicate "$max(x)$" produces a complete splitting in elementary parts (the function max, returns the event with the highest priority among the enabled ones). The predicates "$x = y$" and "$x \neq y$" produce no splitting at all. Once the common set of symmetries is computed, the colour partition is sufficiently refined to allow a symbolic satisfaction test.



$\langle L, \widehat{m}\rangle$
Symbolic
Refinement (t_4)
$\langle L', \widehat{m_1}\rangle$
$\langle L', \widehat{m_2}\rangle$
$\langle L', \widehat{m_3}\rangle$
$t_4(Z_1)$
$t_4(Z_2)$
$t_4(Z_1)$
Symbolic
Synchronisation
CTRL
(t_4)
$t_4(Z_2)$
$t_4(Z_3)$
$t_4(Z_3)$
$\langle L', \widehat{m'_1}\rangle$
$\langle L', \widehat{m'_2}\rangle$
$\langle L', \widehat{m'_3}\rangle$

Figure 4: Symbolic refinement and firing w.r.t. an arc of CTRL

According to a symbolic marking $\langle L, \widehat{m}\rangle$ and the events of a transition enabled from it, a symbolic synchronization operation against an arc labelled by a predicate P of colour partition L_P, is performed in the following three stages :

- A common colour partition L' is computed by intersecting L and L_P;
- this requires to *refine* the symbolic marking representation, since some of the symmetries could now be prohibited; a family $\{\langle L', \widehat{m_1}\rangle, \langle L', \widehat{m_2}\rangle, \ldots\}$ of symbolic markings would be obtained.
- from each $\langle L', \widehat{m_i}\rangle$, the set of enabled transitions (events) is computed, and only those that satisfy the predicate P are

kept. Hence, a set of valid symbolic successors $SUCC$ is obtained.

Assume that the t_4 transition is a potential candidate for a firing. In order to test the satisfaction of the predicate $max(x)$ yielded by the control automaton (CTRL), a decomposition of L in $L' = \{\{1\}, \{2\}, \{3\}\}$ is performed.

With respect to L', the symbolic marking $\langle L, \widehat{m}\rangle$ is refined in the following three symbolic markings $\langle L', \widehat{m_1}\rangle, \langle L', \widehat{m_2}\rangle, \langle L', \widehat{m_3}\rangle$.

In fact, all these markings have the same general form $Rq(\langle Z_1\rangle)$, $Wt(\langle Z_2 + Z_3\rangle)$ but their dynamic subclasses are attached differently to the colour partition :
$d(Z_1) = \{1\}$, $d(Z_2) = \{2\}$, $d(Z_3) = \{3\}$ or
$d(Z_1) = \{2\}$, $d(Z_2) = \{1\}$, $d(Z_3) = \{3\}$ or
$d(Z_1) = \{3\}$, $d(Z_2) = \{1\}$, $d(Z_3) = \{2\}$.

For each of these symbolic markings, a valid successor is obtained by firing the transition, which corresponds each time to take the highest colour in the Wt place : $\langle L', \widehat{m'_1}\rangle$ = $Rq(\langle Z_3\rangle)$, $Wt(\langle Z_1\rangle)$, $Cs(\langle Z_2\rangle)$.
$\langle L', \widehat{m'_2}\rangle$ = $Rq(\langle Z_1\rangle)$, $Wt(\langle Z_2\rangle)$, $Cs(\langle Z_3\rangle)$.
$\langle L', \widehat{m'_3}\rangle$ = $Rq(\langle Z_2\rangle)$, $Wt(\langle Z_1\rangle)$, $Cs(\langle Z_3\rangle)$.

So, the building of the Dynamic Symbolic Reachability Graph (DSRG) looks like a standard algorithm for a reachability analysis but uses symbolic markings and symbolic operations which act on symbolic markings directly.

Reduction of the symbolic structure

In order to make the symbolic structure compact, we introduce two new operations : the *grouping* and the *inclusion* of symbolic markings.

In particular, each set $SUCC$ of symbolic successors can be reduced by grouping several elements in a single pair $\langle L, \widehat{m}\rangle$. Moreover, there may be some elements in $SUCC$ that have already been visited, then can be discarded. Since each symbolic marking are now associated with a different colour partition, the equality test between symbolic representations must be replaced by a more complex inclusion operation.

Below, we formalize the definitions of the grouping and the inclusion operation test. Consider that $[\langle L, \widehat{m}\rangle]$ represents the set of ordinary markings represented by $\langle L, \widehat{m}\rangle$.

Definition 0.1 (grouping) *the symbolic marking* $\langle L_{\widehat{m}}, \widehat{m}\rangle$ *is a valid grouping for the set of symbolic markings* $\{\langle L, \widehat{m_1}\rangle, \ldots, \langle L, \widehat{m_n}\rangle\}$ *iff*
$\cup_{i=1\ldots n} [\langle L, \widehat{m_i}\rangle] = [\langle L_{\widehat{m}}, \widehat{m}\rangle]$.
Note that the refinement and the grouping operations are dual operations.

Definition 0.2 (inclusion) *The symbolic marking* $\langle L_{\widehat{m}}, \widehat{m}\rangle$ *is said to be included in the symbolic marking* $\langle L_{\widehat{m\prime}}, \widehat{m\prime}\rangle$ *iff* $[\langle L_{\widehat{m}}, \widehat{m}\rangle] \subseteq [\langle L_{\widehat{m\prime}}, \widehat{m\prime}\rangle]$.

In our approach, the symbolic inclusion does not bring out new difficulty since it can be brought back to an equality test by refining the symbolic markings to be compared on the same colour partition. The two resulting sets of symbolic markings can thus be compared using a standard symbolic equality test.
The symbolic grouping requires more effort, since the equivalence classes of markings that can be gathered must yield an equivalence class with a symbolic representation. Fortunately our algorithm works on the symbolic representation directly to save computation time.

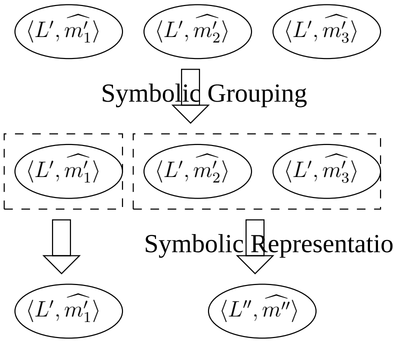

$\langle L', \widehat{m'_1}\rangle$
$\langle L', \widehat{m'_2}\rangle$
$\langle L', \widehat{m'_3}\rangle$
Symbolic Grouping
$\langle L', \widehat{m'_1}\rangle$
$\langle L', \widehat{m'_2}\rangle$
$\langle L', \widehat{m'_3}\rangle$
Symbolic Representation
$\langle L', \widehat{m'_1}\rangle$
$\langle L'', \widehat{m''}\rangle$

Figure 5: An example of symbolic grouping

For instance, continuing our sample, one can note that the three symbolic markings can be candidates for a grouping since they all have the same form. However, only $\langle L', \widehat{m'_2}\rangle$ and $\langle L', \widehat{m'_3}\rangle$, can be replaced by a unique symbolic representation $\langle L_{\widehat{m\prime\prime}}, \widehat{m\prime\prime}\rangle$ with $L_{\widehat{m\prime\prime}} = \{\{1, 2\}, \{3\}\}$ and $\widehat{m\prime\prime} = Rq(\langle Z_1\rangle)$, $Wt(\langle Z_2\rangle)$, $Cs(\langle Z_3\rangle)$ and $d(Z_1) = d(Z_2) = \{1, 2\}$ and $d(Z_3) = \{3\}$.

Evaluation

We have implemented our symbolic methods by reusing the core implementation of the GreatSPN software proposed for qualitative analyses and performance evaluations. GreatSPN is a well-known software which computes the SRG of well formed nets, including the management of the symbolic marking representation (Chiola and Gaeta 1995).

The DSRG module implements the main algorithm using the DySy module, a *dynamic manager of symmetries*, and the Aut module which manages the control automaton and performs the symbolic satisfaction test. The GreatSPN core is reused to realize standard WN operations (e.g. the symbolic firing).

We measure the time spent as well as the memory consumed, in comparison with the SRG and ESRG methods. The memory consumption is measured in number of nodes.
It is worth noting that we used a 2 Ghz Intel Pentium IV machine, with 775 Mbytes memory size and working on Linux 9.1 Operating System.

Table 1 shows how the DSRG and SRG behave, similarly Table 2 compares the DSRG against the ESRG. The model considered for the DSRG is the symmetrical model of Figure 2 synchronized with the CTRL automaton of Figure 3, while the model for the SRG and the ESRG is the asymmetric WN of Figure 1. For this last model, the SRG is the RG due to the partition of the colours of class C in elementary static subclasses.
We note on both tables, that there is an exponential gain in

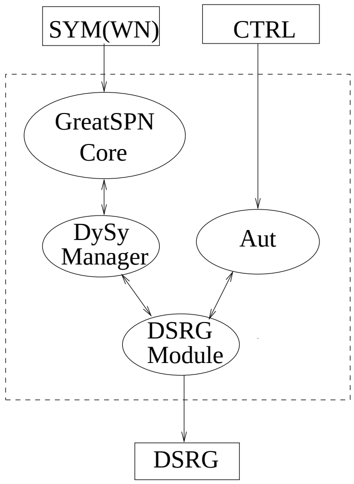


Figure 6: Architecture of the system

time. Also there is an exponential gain regarding the memory consumption.

Table 1: DSRG vs SRG(RG).

#P.	SRG(RG)		DSRG		Ratio	
	T.	#N.	T.	#N.	T.	S.
3	0	139	0	28	0	5
5	2	2709	0	96	0	29
7	41	50159	3	253	14	199
9	1147	911017	45	559	26	1630
11	—	16378179	1830	1090	–	15026

Table 2: DSRG vs ESRG.

#P.	ESRG		DSRG		Ratio	
	T.	#N.	T.	#N.	T.	S.
3	0	54	0	28	0	2
5	1	441	0	96	0	5
7	25	4918	3	253	5	20
9	939	57211	45	559	21	102
11	54074	639056	1830	1090	30	586

References

Capra, L., C. Dutheillet, G. Franceschinis and J.M. Ilie (1999). Towards Performance Analysis with partially Symmetrical SWN. In: *Proc of MASCOTS'99, 7th Int Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. IEEE Computer Society Press. University of Maryland, College Park MD, USA.

Chiola, G. and R. Gaeta (1995). Efficient Simulation of Parallel Architectures Exploiting Symmetric Well-formed Petri Net Models. In: *Sixth International Workshop on Petri nets and Performance Models*. IEEE Computer Society Press. Durham, NC, USA.

Chiola, G., C. Dutheillet, G. Franceschinis and S. Haddad (1993). Stochastic well-formed coloured nets for symmetric modelling applications. *IEEE Transactions on Computers* **42**(11), 1343–1360.

Haddad, S., J.M. Ilié and K. Ajami (2000). A model checking method for partially symmetric systems. In: *Proceedings of FORTE/PSTV'00*. Kluwer Academic Publishers. Pisa, Italy. pp. 121–136.