High Level Branching Processes for High Level Petri Nets

Fleischhack, H. and Pelz, E.

Technical Report TR-2002-18

Laboratory of Algorithms, Complexity and Logic
University of Paris XII, Val-de-Marne
61, avenue du Général de Gaulle
F-94010 CRÉTEIL Cedex – FRANCE
Tel: +33 (0)1 45 17 16 47
Fax: +33 (0)1 45 17 66 01
High Level Branching Processes for High Level Petri Nets

Hans Fleischhack
Fachbereich Informatik
CVO Universität Oldenburg
D-26111 Oldenburg
hans.fleischhack@informatik.uni-oldenburg.de

Elisabeth Pelz
LACL, Faculté de Sciences
Université Paris XII-Val de Marne
F-94010 Créteil Cedex
pelz@univ-paris12.fr

Abstract
Complex parallel or distributed systems, including real time ones, can be modeled in a compact way by means of high level Petri nets. Among the most efficient verification techniques for Petri nets are partial order model-checking methods (à la McMillan), based on the concept of branching processes. Up to date, these are only applicable to low level Petri nets. A high level net first has to be unfolded into a low level one. This unfolding process often results in an unnecessary space explosion. The paper aims at proposing a high level version of such branching processes which can be defined directly from the high level net model. These are shown to be equivalent to the low level ones. They will become the basic structures on which new really efficient model-checking algorithms can be defined.

0 INTRODUCTION
Since 1994, a group around Esparza is developing model-checking techniques for the verification of concurrent systems $\Sigma$ based on the concept of net-unfolding, a well-known partial order semantics [E94,?]. The method uses a complete finite prefix $\beta_H(\Sigma_U)$ (à la McMillan, [M92, M95]) of the maximal branching process of a system $\Sigma$, initially given by a finite low level (II) Petri net $\Sigma_U$. There is an associated logic, Esparza’s logic $L$ in which the properties to be checked can be easily formulated. These partial order methods often avoid the state explosion problem, which is well known for classical (sequential) model checking [QS81, CES86].

For the specification of ‘real world’ problems which may require the use of large—possibly infinite—data types, high level (hl) Petri nets ($\Sigma_H$) have been introduced as a compact representation. Model checking of these systems unfortunately requires preliminary unfolding $\Sigma_H \rightarrow \Sigma_U$, generating possibly exponentially larger or even infinite nets.

In [FP00], we achieved a first improvement of this approach by calculating the finite (low level) prefix $\beta_U$ of the maximal branching process directly from the the high level model $\Sigma_H$, i.e. without preliminary unfolding.

Now we propose to improve substantially by defining the notion of high level maximal branching process using a symbolic representation of data for it, and its finite prefix $\beta_H$. It can be calculated by first transforming the original high level net into an equivalent high level net with respect to some well established equivalence relations [KR96] and then doing some symbolic unfolding.

Motivating - and convincing - examples will illustrate our approach, where for high level nets with large or infinite low level branching processes we obtain very small high level ones.

These high level branching processes will now constitute the basic structures for a partial order high level model checking method, where Esparza’s logic $L$ need to be extended slightly by atomic formulas of the form $v \in A$, where $v$ denotes a variable and $A$ a type. The definition of this method is the goal of a European research collaboration.

1 A CLASS OF HIGH LEVEL PETRI NETS
The results shown in this paper apply to all high level net classes fulfilling the following conditions:

(i) The types of the places are arbitrary subsets (possibly infinite) of some given enumerated set of values VAL.
(ii) Arc inscriptions are (possibly infinite) multisets over VAL or variables VAR.
(iii) The firing conditions of transition are arbitrary sets of predicates built over the sets VAL, VAR and a set of fitting operators.

This applies, for instance, to colored Petri nets [J92]. For reasons of simplicity, in this paper the class of compositional colored nets, called M-nets, is chosen as net model [BFP95,?]. For these a good equivalence relation, Klausel-Riemann equivalence, is properly defined and proved consistent with respect to the unfolded 1-safe nets, Petri Boxes [BDH92].

Two M-nets $N_1$ and $N_2$ are called KR-equivalent, noted $N_1 \equiv_{KR} N_2$, if their associated low level nets,
obtained by unfolding, are duplication equivalent, i.e. if the two low level nets only differ by the presence of places or transitions having the same connectivity and the same labels as others [KR96, KRG96]. There, a system of transformation rules on M-nets (called here KR-rules) is defined which is proved to be complete with respect to the KR-equivalence.

M-nets are able to represent inside the same model most kind of paradigms we can meet in the area of parallel, real time or distributed computing, such as synchronous and asynchronous communication, the control and as well the data part of programs, timing constraints, preemption, abstraction, mobility, ... [BFP98, ?]. An appropriate tool, the famous PEP tool [GB96,?] has been developed, which allows to design, model, simulate and verify M-nets and their corresponding low level nets, Petri Boxes.

In the Appendix, the notion of M-nets is defined and illustrated.

2 MODEL-CHECKING AND LOW LEVEL BRANCHING PROCESSES

A process of a net traces one possible, truly concurrent execution of the net by a (cycle-free) occurrence net preserving locally the causality and independence relations. The maximal branching process regroups all processes in a single (cycle-free) occurrence net, i.e. all choices are now presented. This concept is often called 'unfolding', too, as it 'unfolds' all possible executions [ENG91]. A concurrent system (think of it as a tuple of communicating transition systems) is to its 'unfolding' what a single transition system is to its unwinding into a tree. Branching processes usually are infinite, and so they cannot be directly stored in a computer. However, K. McMillan observed in [McM93] that it is possible to construct a finite initial part of the branching process containing as much information as the unfolding itself. Later Esparza and al. improved the algorithm, and we call such an initial part the complete finite prefix [ERV95]. The set of reachable states of a complete finite prefix coincides with the set of reachable states of the system. The construction of the structure is stopped at some cut-off events, whose precise definition is not important for the purpose this paper. Finite complete prefixes can be generated, stored in a computer, and used to check behavioural properties [E94].

Model checking of high level net systems unfortunately requires preliminary unfolding $\Sigma_{hl} \rightarrow \Sigma_{ill}$ into a one safe Place/Transition system, generating possibly exponentially larger or even infinite nets, before computation of the complete finite prefix can be started. To resume the two steps:

- net unfolding $U : \Sigma_{hl} \rightarrow \Sigma_{ill}$
- and Esparza’s construction $\beta$ of (the finite complete prefix of the) branching process on the net $\Sigma_{hl}$.

An other, more efficient way to calculate the branching process of an M-net $\Sigma_{hl}$ was proposed in [FP00]. It consists in defining directly the desired low level branching process $\beta_{hl}$. We have proved the coherence with the traditional two steps algorithm:

Theorem 1.

$$\beta_{hl}(\Sigma_{hl}) = \beta(U(\Sigma_{hl}))$$

The definition given here is the second one, that by the algorithm $\beta_{hl}$ which proceeds by induction (as already the traditional one $\beta$). In this paper, that notion is improved by a parallel treatment at the level of cuts depending of the enabled transitions for each legal binding.

First let us define cuts: A cut of a branching process is a maximal anti-chain of places which has no pair of conflicting transitions in its prefix; it presents the marking reached by the net after having executed the events of its prefix.

Construction of the maximal low level branching process $\beta_{hl}(\Sigma_{hl})$:

**Begin of induction:** unfolding of the initial marking $M_0$ of $\Sigma_{hl}$:
For each $s \in S$ and each $v \in type(s)$: generate new places $\{(s, v, i)|1 \leq i \leq M_0(s, v)\}$.

**Induction step:** for each already reached cut $M_t$, for each $t \in T$ and each legal binding $\sigma$ enabling $t$ at $M_t$, add one transition $t_\sigma$ to the occurrence net. For each place $s \in t^*$ and each $v \in type(s)$ we create a new place $p_v$, then we add post-places $s - v$ and arcs $(t_\sigma, s_v)$ if $s_v$ appeared in the label of arc $\sigma(t, s)$. We are only interested in the complete finite prefix of the branching process whose cut-off events are calculated with respect to exactly the same criteria as in Esparza’s algorithm [ERV95].

3 COMPLEXITY OF THE BRANCHING PROCESS CONSTRUCTION

In order to understand why it is usually too inefficient to check a property on the complete finite (low
level) prefix of the branching process, we will evaluate the size of it. In particular, we will see that locally, around a transition, it can already explode. Let

- \( \Sigma_{hl} \) an M-net;
- \( t \) a transition of \( \Sigma_{hl} \);
- \( \varphi = (y_1, y_2, \ldots, y_n) \), the variables in the area of \( t \) seen as a finite vector: \( \varphi = \text{var}(t) \);
- \( \beta(t) \) the complete finite prefix of the branching process;
- \( \gamma(t) \) the firing condition of \( t \) given by the set of predicates \( \phi(y_1, \ldots, y_n) \) and
- for all \( i, \ 1 \leq i \leq n, \ A_i = \text{type}(y_i), \) the type of the variable \( y_i \).

In the construction \( \beta(t) \) and for a given cut or marking \( M \) of \( \Sigma_{hl} \), \( t \) gives rise to as many copies as there are bindings of \( (y_1, \ldots, y_n) \) satisfying the condition \( \phi \) for this marking. More precisely, let

\[
\tilde{S}_t = \{ \tilde{\varphi} = (a_1, \ldots, a_n) \in A_1 \times \ldots \times A_n \mid \forall i \ a_i = \sigma(y_i) \ \text{and} \ \{ \sigma(\varphi) = \phi(\varphi) \} \ \text{satisfied by} \ M \}.
\]

Then we obtain the following result:

**Proposition 1.** In \( \beta(t) \) there are \( |S_t| \) copies of \( t \). Moreover, the local data around these copies of \( t \) are represented by \( |A_1 \times \ldots \times A_n| \) places.

Parts which are inherited from the high level features, as the set \( S_t \) or the number of places generated around the new copies of transitions, are candidate for reduction. More precisely, we like to regroup parts of \( S_t \) and sets of places, and to replace them by symbolic (high level) representations.

High level branching processes will have, as already the low level ones, occurrence nets as underlying net structure. This signifies in particular, that they have to be unmarked and without multiple arcs. But they are allowed to have symbolic arcs (labeled by one variable), symbolic places (with sets being their type) and symbolic transitions (having adjacent symbolic arcs).

**4 MOTIVATING EXAMPLES**

We need to determine, if we are able to associate 'directly' a high level branching process to a given M-net. For this we need to know which parts can be treated symbolically and how to recognize these parts. Are they immediately recognizable or only after some preprocessing? If some preprocessing is necessary, it should be a net-transformation which yields an equivalent net, possibly a KR-equivalent one: the goal is an high level construction (provably) coherent with the low level one.

The Petri nets studied in this section represent a sample of nets to provide us with arguments for the choice of a feasible definition of high level branching processes. Also, they will illustrate the construction of some intermediate high level nets with some 'good properties' and show how these properties have to look like. Without loss of generality, each M-net considered here will be finite and cycle free, therefore its maximal low level branching process will be identical to its complete finite prefix. By this way, all obtained low level branching processes will have a small (tree) height and we do not need to treat the cut-off events, whose definition would, in each case, be transposed from low level to high level without change.

Thus we can concentrate on the problem of the (tree) width of the branching processes. The goal is to unfold as few as possible at a fixed level, and treat symbolically all what is 'assimilable'.

**First Example**

Our first M-net illustrates a common situation where the firing condition of a transition \( t \) implies an equivalence relation on natural numbers: \( t \) can be fired if the values assigned to \( x \) and \( y \) are such that their sum is even.

![Figure 1. The M-net \( \Sigma_1 \)](attachment:image)

By distinguishing the odd and even values of \( x \) and \( y \), we can split \( P \) (respectively \( P' \)) as well as \( t \) (using the transformation rules of [KR96,?]), and obtain immediately the following KR-equivalent M-net \( \Sigma_1 \). Now the firing conditions of the copies of \( t \) are trivially 'true'. This signifies that independently of the chosen bindings for \( x \), \( y \) and \( z \), both copies of \( t \) are always firable. These copies could therefore be treated symbolically.
For this example the high level branching process can be chosen to be isomorphic to the intermediate net $\Sigma_1'$ by two reasons: a) the initial net is without cycles and so cannot be unfolded further in depth, and b) $\Sigma_1'$ only contains transitions which can be treated symbolically, which signifies that these transitions will not be unfolded in width. We may conclude a first rule:

**Observation**

If $t$ is a transition with firing condition 'true', then $t$ can be treated symbolically.

**Second Example**

In the next example we will consider some combinatorial choice. Let be $N_{100} = \{0, 1, \cdots, 99\}$ the hundred first natural numbers. Transition $t$ chooses 50 numbers from $N_{100}$ and puts them into $P$, $t'$ chooses one single element from $P$ and puts it into $Q$. Let us remark that the firing conditions of $t$ and $t'$ are already trivially true.

To become able to treat $t$ symbolically, first the multiple arc has to be unfolded. Thus we are first looking for an equivalent M-net where this arc is unfolded into 50 simple arcs.

**Third Example**
Let us consider now the M-net \( \Sigma_3 \) which is finite and cycle-free. Thus its low level branching process and its finite complete prefix are still identical. This branching process contains 100 copies of \( t : t_1, \ldots, t_{99} \) (one for each value of \( w \)), 100 × 100 copies of \( P \), 100 of \( P' \) and \( 99 \times 100 = 9900 \) copies of \( t' \) and as many copies of the place \( Q \). Thus \( \beta_{hl}(\Sigma_3) \) is enormous.

How to regroup intelligently? We could distinguish the bindings of \( w \), that with value 0 and all the others. Thus an intermediate M-net \( \Sigma'_3 \) can be obtained by splitting \( t \) into two transitions, one for 0 and one for non zero values of \( w \). Thus \( P' \) also splits into two places, and \( (N_{100} \backslash \{0\}) \) becomes the type of \( P'_{\neq 0} \), singleton \( \{0\} \) the other one.

The transitions \( t_0, t_{\neq 0}, t'_{\neq 0} \) all have a 'true' condition. But the transition \( t'_0 \) has the condition 'false'. Such a transition is called 'dead' and can be removed.

Both M-nets of figure 5 are KR-equivalent (the splitting could be done by the KR-transformation rules).

Now, the construction of the high level branching process can be tried. \( P \) needs to be unfolded twice, because of its multiple ingoing arcs, in hundred places \( P_j \) with type \( \{j\}, (j \leq 99) \); idem for \( Q \). Transition \( t_{\neq 0} \) could be treated symbolically, therefore \( P'_{\neq 0} \) becomes a symbolic place (not unfolded). The result is the occurrence net of figure 6.

Compared with \( \beta_{hl}(\Sigma_3) \), the high level occurrence net \( \beta_{hl}(\Sigma'_3) \) is very small: it needs only two copies of \( t \) (instead of 100), 100 copies of \( t' \) (instead of 9900), 2 × 100 places corresponding to the former place \( P \) (instead of 10000), 2 copies of \( P' \) (instead of 100) and 100 copies of \( Q \) (instead of 9900).

**5 EQUIVALENT HIGH LEVEL NETS**

From the previous examples we can generalize that we first should transform the initial M-net into an equivalent one \( \Sigma' \) with good properties from which the construction of a high level branching process becomes easy. We will make precise now which properties are desired.

**Preprocessing for multiple arcs**

We have seen the obstacle of multiple arcs: a place cannot be treated symbolically if it has an ingoing or outgoing arc labeled by a non singleton set where variables appear. The reason is that here we do not have
to consider an arbitrary instance of a variable, but an arbitrary combination of instances of a lot of variables, like for net $\Sigma_1$. In opposition to this, there is no problem if a multiple arc only contains constants, like for net $\Sigma_2$ (all natural numbers from 0 to 99). In this case, the unfolding into simple arcs can be postponed to the construction of the branching process. The desired property therefore is:

**Property 1**: If $\Sigma'$ has a multiple arc then its label consists of a multi-set of values from $Val$.

The transformation needed here (to ensure property 1) is a partial unfolding: arc $t$ with multiplicity $n$ from or to a place $p$ labeled by variables will be replaced by $n$ simple arcs from or to $n$ copies of $p$ each one having the same inscription as $p$, which now may become symbolic places.

At the moment, this partial unfolding cannot be done within the KR-rule system because that system was defined for M-nets without multiple arcs. Some consistent extension of KR-equivalence and KR-rules to a larger class of M-nets (that with multiple arcs) is possible without problem. Some rules have to be added to unfold multiple arcs, and the other side round, to fold arcs together. In the extended KR-equivalence two nets will be equivalent if their total unfoldings yield the same low level nets.

In the sequel we suppose that all M-nets considered are without multiple arcs having several variables. i.e., we suppose that an adequate preprocessing has been done on the M-nets.

**Partitions**

The second requirement follows from the observation made for example 1:

**Property 2**: All transitions in the net have condition 'true'. Such transitions are called trivial.

In a net fulfilling property 2, places around transitions can be treated as symbolic places and do not need to be unfolded. Example $\Sigma_2$ also illustrates this: every instance for the unique variable $x$ (or $y$) will satisfy the now trivial firing condition.

KR-equivalence will be the adequate equivalence relation to relate the initial net to that where a transition is split into several copies with firing condition 'true'. Let us explain how to transform an M-net accordingly:

Let $\Sigma_{hl}$ be an M-net without multiple arcs inscribed with variables. And let $t_1, t_2, ..., t_m$ be an enumeration of the transitions of $\Sigma_{hl}$ compatible with accessibility. I.e., $t_i$ is enabled at the initial marking $M_0$ and $t_{i+1}$ is enabled at the marking reached after having fired the sequence $t_1, t_2, ..., t_i$. Following this enumeration, for each transition $t$ we perform a local transformation, creating as many copies of $t$ as necessary to have only firing conditions 'true' at all copies. (No transformation applies if the condition is initially 'true'). Such a splitting is just the result of the application of the KR-rule system. This transformation is based on the following partition result:

**Theorem 2.** Let $t$ be a transition of $\Sigma_{hl}$ firable from a marking $M$ and $\Sigma_i$ be the set of vectors of values making the firing condition of $t$ true under $M$, as defined in Section 3.
If there exists a natural number $k_t$, and a partition of $\Sigma_i$ into $k_t$ cartesian products of the shape

$$(A_1^n \times ... \times A^n_{k_t}) \cup (A_1^{n_1} \times ... \times A_{k_t}^{n_{k_t}})$$

and if $\Sigma_{hl}(t)$ is obtained by $\Sigma_{hl}$ by splitting $t$ in $k_t$ copies $t_1, t_2, ..., t_{k_t}$ by applying the KR-rules, then

$$\Sigma_{hl} \equiv_{KR} \Sigma_{hl}(t).$$

Moreover, each copy $t_i$ has firing condition 'true'.

**Consequence**

We only need at most $n \times k_t$ places around the $k_t$ new transitions $t_i$.

The proof of this theorem follows directly from the application of the KR-rule system and the completeness proof in [KR96].

We deduce the following inductive algorithm from this result:

**Begin of induction:** Let $\Sigma_0 = \Sigma_{hl}$.

**Induction hypothesis:** Let $\Sigma_i$ be the M-net obtained after the local transformation around $t_i$.

**Induction step:** Now let $t = t_{i+1}$ be the next transition to treat and M the marking of $\Sigma_i$ reached after the firing of the sequence $t_1, t_2, ..., t_i$. Let $k$ be the minimal $k_t$ for which exists a partition of $\Sigma_i$ in a cartesian product of $k_t$ parts as specified in the theorem above. i.e., we distribute the different vectors of values for which the firing condition of $t$ is satisfied in a good (and minimal) partition. Then we apply the KR-transformation rules to split $t$ in $k$ corresponding copies which has as side effect a splitting of the adjacent places in at most $n \times k$ places. We set: $\Sigma_{i+1} = \Sigma_i(t)$. 
Finally, $\Sigma'_{hl} = \Sigma_{ml}$ is the desired intermediate M-net with 'good properties'.

6 HIGH LEVEL BRANCHING PROCESSES

Here we start with some intermediate M-net $\Sigma'_{hl}$ fulfilling properties 1 and 2. A high level branching process can be associated to it as follows:

We proceed inductively as in the construction of $\beta_l(\Sigma'_{hl})$ from section with one exception: bindings are replaced by substitutions $\sigma$ verifying $\sigma(x) = x$ for variables $x$ from arcs (necessarily from simple arcs because of property 1). We will call the method symbolic unfolding.

If $s$ is a place initially marked without multiple outgoing arcs and with at least one outgoing arc labeled by a variable it is not unfolded.

If transition $t$ is considered at cut $M$, an adjacent multiple arc is always unfolded because the goal is an occurrence net. As its label necessarily contains only values, this splits the places on the place on the other end of the arc, too. Moreover if $t$ is enabled by only one substitution $\sigma$ at cut $M$, it is not split, but just one $t_\sigma$ is added in the branching process. Consequently, the variable arcs around $t$ are not unfolded, nor are the places at their respective other ends. These places, transitions and arcs are now symbolic.

As before, we are only interested in the complete finite prefix of the now high level branching process whose cut-off events are calculated with respect to exactly the same criteria as in Esparza's algorithm [ERV95].

At the end of the construction we have an M-net, called $\beta_l(\Sigma'_{hl})$, without any multiple arcs. Thus the high level branching process and the low level one can be proved KR-equivalent:

**Theorem 3.**

\[ \beta_l(\Sigma'_{hl}) \equiv_{KR} \beta_l(\Sigma_{hl}) \]

7 WORST CASE COMPLEXITY

The reduction rate on the given examples is spectacular. But, the method based on partitions defined here, does not always bring an optimization. In the case where the minimal $k_i$ for theorem 2 is the maximal one, i.e., $k_i = |S_i|$, each element of the partition is a single vector of values. This is a possible consequence for some firing conditions which do not allow regroupment of values.

Consider a net with only one transition $t$ between two places with type $N$, label $x$ on the incoming arc, label $y$ on the outgoing one, and the firing condition of $t$ requiring incrementation, i.e., $t$ having the inscription $\theta \{ x+1 = y \}$, we encounter this worst case. Then the high level branching process is just the low level one. Cases like this appear during modeling. For instance, for causal time M-nets with a clock net [KP00,P02], the clock is continuously incremented.

8 CONCLUSION

We have shown in this paper how we can associate a high level branching process to a high level net in three steps:

(i) Preprocessing to unfold multiple arcs with variables.
(ii) Partitioning to obtain firing condition 'true'.
(iii) Symbolic unfolding.

In a lot of cases, this permits to obtain very small structures. In a recent master thesis it has been studied how the partitions of $S_i$ can be calculated depending on the particular predicates in the firing condition of $t$ [L02]. Now, slightly modified model-checking techniques need to be developed, which are able to check properties on these high level structures. A European collaboration involving the Universities of Newcastle upon Tyne, Oldenburg, UL Brussels and Paris 12 aims to attack this goal. To avoid the quoted worst case, we will also try to improve the equivalence relation $\equiv_{KR}$. In [KK02] our Newcastle’s partners propose some first ideas on this problem. Another notion to be exploited could be the concept of 'region', as for timed automata. In a second time, the high level model checking features should be integrated in the tools PEP and MARIA [G97,?].

*References*


