A Local Cut-off Criterion for Unfoldings of Safe Petri Nets

Status Report of a Doctoral Thesis

Niels Lohmann
Supervisor: Wolfgang Reisig

Humboldt–Universität zu Berlin
Institut für Informatik
nlohmann@informatik.hu-berlin.de

Abstract. Finite prefixes of branching processes are a compact way to represent the state space of finite safe systems. McMillan in [1] formulated a *cut-off criterion* stating under which conditions the (usually infinite) maximal branching process can be truncated without loosing reachable markings. However, this criterion depends on the reachable markings itself and thus suffers the state explosion problem. In this work, we propose local cut-off criteria that base on partial markings.

1 Introduction

Computer-aided verification and especially model checking based on the transition system of a distributed system usually suffers the state explosion problem: original independent—and thus unordered—actions are ordered arbitrarily, thus yielding an exponential number of interleavings and intermediate states. Many approaches exists to avoid or at least ease the state explosion (see [2] for a survey).

One widely accepted approach is a finite prefix of a branching process as introduced by McMillan in [1]. McMillan defines a criterion how the (usually infinite) maximal branching process of a finite 1-safe system Σ can be truncated such that the remainder (an *unfolding* of Σ) still represents all reachable markings of Σ. The resulting unfoldings are usually much smaller than the state space of Σ.

Several works enhance this cut-off criterion to allow for unfoldings of other net classes, e. g. time Petri nets [3], unbounded nets [4] or nets with read arcs [5]. Other works focus on model checking using unfoldings [6–8] or on efficient algorithms to generate finite prefixes [9]. Moreover, the original cut-off criterion itself has been generalized and improved to generate smaller unfoldings [10, 11].

However, the main idea of how to truncate the maximal branching process while preserving all reachable markings has not changed: intuitively, the branching process can be truncated whenever a *configuration* describes a marking represented by a configuration processed before. As markings depend on the whole net they can be considered as a *global* cut-off criterion. That is, during the construction of the unfolding each marking (or configuration) reached has to be stored to find future cut-off points. This has a enormous drawback on the efficiency of any unfolding algorithm as the number of reachable markings of course suffers from the state explosion.

2 Goals of the Doctoral Thesis

In my doctoral thesis I study *local* cut-off criteria of unfoldings. Instead of using configurations or markings as cut-off criterion, as a first goal, I am going to define cut-off criteria that only take the prefix of a *single* condition into account to decide whether or not that condition has to be part of the finite prefix.

The second goal is the improvement of the local cut-off criteria to generate unfoldings of minimal size. While several approaches (e. g. [12, 13]) propose a subsequent minimization, the local criteria may allow for "on-the-fly" minimization.

The third goal is a prototypic implementation to validate the theoretical results. The local criteria may be implemented more efficiently and—due the locality—also distributedly. Furthermore, the unfolding algorithm may be combined with known reduction techniques such as exploiting net symmetries [14, 15] or "forgetting" information that is not any longer relevant for further unfolding [16].

3 Current Results

In this section we present the first results of the thesis: first we sum up basic definitions used in the rest of this paper. Then in Sect. 3.2 we consider a restricted class of systems, namely finite safe systems without join transitions. For those systems a simple local cut-off criterion can be defined. Finally in Sect. 3.3 we study arbitrary finite safe systems, and introduce an enhanced cut-off criterion.

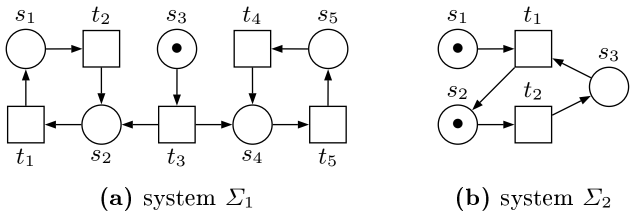
3.1 Preliminaries

The reader is expected to be familiar with Petri nets [17], branching processes [18] and finite prefixes of the latter as introduced by McMillan [1]. We use the standard notations $N = (S, T, F)$ for nets, $\Sigma = (S, T, F, m_0)$ for systems and $\beta = (B, E, F, h)$ for branching processes, where $h : B \cup E \to S \cup T$ is a net homomorphism. Furthermore $\upharpoonleft x$ denotes the prefix of a node $x \in B \cup E$ and $Min(\beta)$ is the set of conditions $b \in B$ with $\upharpoonleft b = \emptyset$.

We structurally fix a class of transitions:

Definition 1 (join transition). *Let $N = (S, T, F)$ be a net. A transition $t \in T$ is a* join transition *iff $|{}^\bullet t| > 1$. Denote the set of join transitions with T_{join}.*

Figure 1 shows two finite 1-safe systems: while Σ_1 (Fig. 1(a)) has no join transitions, Σ_2 (Fig. 1(b)) contains a join transition: t_1. We use these systems in the following subsections to demonstrate the cut-off criteria.



(a) system Σ_1 **(b)** system Σ_2

Fig. 1. Two finite safe systems: Σ_1 without join transitions (a) and Σ_2 with join transition t_1 (b).

3.2 Systems without join transitions

In this subsection we only consider finite safe systems without join transitions, i.e. systems that fulfill ${}^\bullet t = 1$ for all transitions $t \in T$.[1] This constraint has an impact on the structure of the resulting branching processes:

Lemma 1 (branching process is a forrest). *Let $\beta = (B, E, F, h)$ be a branching process of a finite safe system without join transitions. Then β is a forest*[2] *having the elements of $Min(\beta)$ as roots.*[3]

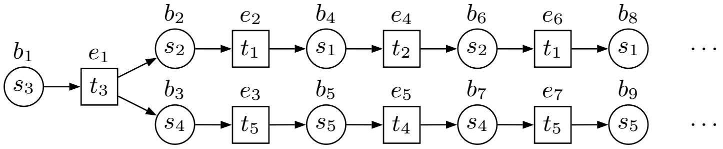


Fig. 2. Branching process of system Σ_1. The resulting graph is an infinite tree rooted by condition b_1.

A prefix of the maximal branching process of Σ_1 is depicted in Fig. 2. As the transitions t_1, t_2, t_4 and t_5 are live, the maximal branching process is infinite. We can truncate the branching process at the cut $\{b_6, b_7\}$. This cut represents the marking $\{s_2, s_4\}$ which was reached before from the initial marking by firing t_3. In fact, McMillan's cut-off criterion marks out the events e_4 and e_5 as cut-off events, both having e_1 as base event.

Instead of using configurations or markings to truncate the branching process, we can examine the *partial* markings of the two infinite "strings" beginning at event e_1 separately:

[1] The requirement $|{}^\bullet t| \leq 1$ is not sufficient as $|{}^\bullet t| = 0$ would make all places in $t^\bullet$ unbounded.

[2] A forest is a graph in which any two vertices are connected by at most one path. An equivalent definition is that a forest is a disjoint union of trees.

[3] Esparza in [19] introduces a "virtual event" $\bot$ with $\bot^\bullet = Min(\beta)$. This event would make the described forest be a tree having $\bot$ as root.

Definition 2 (cut-off condition, base condition). *Let $\beta = (B, E, F, h)$ be a branching process of a finite safe system without join transitions. Let $B_{\text{cutoff}} \subseteq B$ be the set of* cut-off conditions *given by:*

$$B_{\text{cutoff}} = \{b \in B \mid \exists b' \in B : b' \prec b \wedge h(b) = h(b')\}.$$

Given a cut-off conditions b, we have a uniquely defined base condition *$b'' \in B \setminus B_{\text{cutoff}}$ with $b'' \prec b$ and $h(b) = b(b'')$.*

From Def. 2 and Lemma 1 we conclude the following lemma:

Lemma 2 (concurrency and cut-off conditions). *Let b be a cut-off condition, b' its base condition and b^* an arbitrary condition. Then b co b^* iff b' co b^*.*

Proof. Follows from the fact that the underlying system does not contain join transitions and therefore the branching process is a forest. □

We now use the cut-off criterion defined in Def. 2 to define a finite prefix of the maximal branching process:

Definition 3 (cc-unfolding). *Define β_{cc} to be the branching process having $E_{\text{cc}} = \{e \mid {}^\bullet e \not\subseteq B_{\text{cutoff}}\}$ as events and $B_{\text{cc}} = Min(\beta) \cup E_{\text{cc}}{}^\bullet$ as conditions. β_{cc} is called* cc-unfolding.[4]

Using the local cut-off criterion we can truncate the maximal branching process of Σ_1 at two cut-off conditions, namely b_6 and b_7. The resulting cc-unfolding is depicted in Fig. 3.

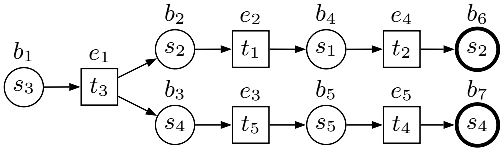


Fig. 3. The cc-unfolding of system Σ_1. b_6 and b_7 are cut-off conditions (depicted bold) having b_2 and b_3 as base condition, resp.

This small example shows the effect of the local cut-off criterion: instead of storing all reachable markings ($\{s_3\}$,$\{s_2, s_4\}$,$\{s_2, s_5\}$,$\{s_1, s_4\}$,$\{s_1, s_5\}$), only partial markings ($\{s_1\}$,$\{s_2\}$,$\{s_3\}$,$\{s_4\}$,$\{s_5\}$) are stored during the construction of the cc-unfolding. However, the resulting finite prefix still represents all reachable markings of the system:

Proposition 1 (completeness of $\boldsymbol{\beta_{\text{cc}}}$). *Let Σ be a finite safe system without join transitions and m be a reachable marking of Σ. Let β_{cc} be the cc-unfolding of Σ. Then there exists a cut C of β_{cc} such that $h(C) = m$.*

[4] "cc" stands for cut-off conditions.

Proof. Assume there does not exist such a cut in β_{cc}. As m is a reachable marking of Σ, there exists a cut C' of the maximal branching process β of Σ with $h(C') = m$.

As C' is no cut of β_{cc}, $C' \setminus B_{cc} = \{b_1, \dots, b_n\} \neq \emptyset$. By Def. 3 we have $\{b_1, \dots, b_n\} \subseteq B_{\text{cutoff}}$, and by Def. 2 there exist $b'_i \in B_{cc}$ with $b'_i \prec b_i$ and $h(b'_i) = h(b_i)$ for all $1 \leq i \leq n$.

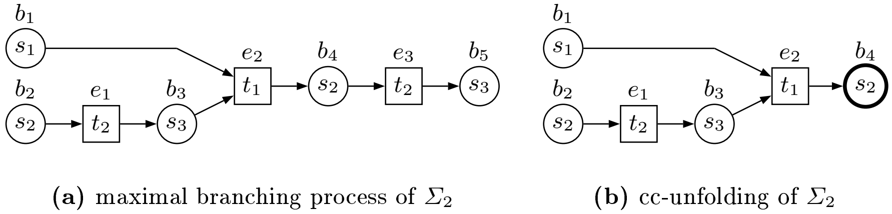
Let $C'' = C' \setminus \{b_1, \dots, b_n\}$, i. e. $C'' \subseteq B_{cc}$. As C' is a cut and $b_1 \in C'$ we have b_1 *co* b_i for all $b_i \in C'$. The results of Lemma 2 state that $(C' \setminus \{b_1\}) \cup \{b'_1\}$ is a cut with $h((C' \setminus \{b_1\}) \cup \{b'_1\}) = h(C')$. Repeating this argument we can construct a cut $C'' \cup \{b'_1, \dots, b'_n\}$ with $h(C'' \cup \{b'_1, \dots, b'_n\}) = h(C')$ and $C'' \subseteq B_{cc}$ which contradicts the assumption that no such cut exists. □

Considering a finite safe system, the resulting cc-unfolding is isomorphic to the unfolding defined by McMillan:

Proposition 2 (cut-off conditions and cut-off events [1]). *Let Σ be a finite safe system without join transitions, β_{cc} the cc-unfolding of Σ and $b \in B_{cc} \cap B_{\text{cutoff}}$ a cut-off condition. Then the event e with $e^\bullet = \{b\}$ is a cut-off event as defined in* [1].

3.3 Systems with join transitions

The results of Sect. 3.2 base on the observation that the branching process of finite safe systems without join transitions are forests. However, this does not hold for systems with join transitions. As a result, the cut-off criterion does not preserve completeness of the cc-unfoldings:



(a) maximal branching process of Σ_2 **(b)** cc-unfolding of Σ_2

Fig. 4. The maximal branching process of Σ_2 (a) and its cc-unfolding using the cut-off criterion of Sect. 3.2 (b). In the latter, b_4 is a cut-off condition (depicted bold) having b_2 as base condition.

In the finite safe system Σ_2 (c. f. Fig. 1(b)) the marking $\{s_3\}$ is reachable by firing the transition sequence $t_2 t_1 t_2$. This marking is represented by the cut $\{b_5\}$ of the maximal branching process of Σ_2, c. f. Fig. 4(a). However, this marking is not represented by a cut in the cc-unfolding (c. f. Fig.4(b)) using the cut-off criterion of Def. 2.

This example shows that it is incorrect to truncate the branching process when a cut-off condition is identified. To avoid this premature truncation, the cut-off criterion has to be adapted to the new structure of the branching processes.

As a first approach, we add labels to the conditions of the branching process. These labels consist of the set of join transitions occurring in the configuration (i. e. in the prefix) of each condition.

Definition 4 (labeling function). *Let $\beta = (B, E, F, h)$ be a branching process and $l : B \rightarrow \mathcal{P}(T_{join})$ a function labeling each condition $b \in B$ as follows: $l(b) = h(\downarrow b) \cap T_{join}$.*

Figure 5 shows the labeled maximal branching process of system Σ_2. The labels are depicted below the conditions.

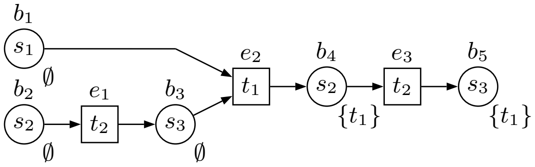


Fig. 5. Labeled maximal branching process of system Σ_2.

We now use the labels to define an enhanced cut-off criterion:

Definition 5 (new cut-off criterion). *Let $\beta = (B, E, F, h)$ be a labeled branching process of a finite safe system. Let $B_{cutoff} \subseteq B$ be the set of cut-off conditions given by: $B_{\text{cutoff}} = \{b \in B \mid \exists b' \in B : b' \prec b \wedge h(b) = h(b') \wedge l(b) = l(b')\}$.*

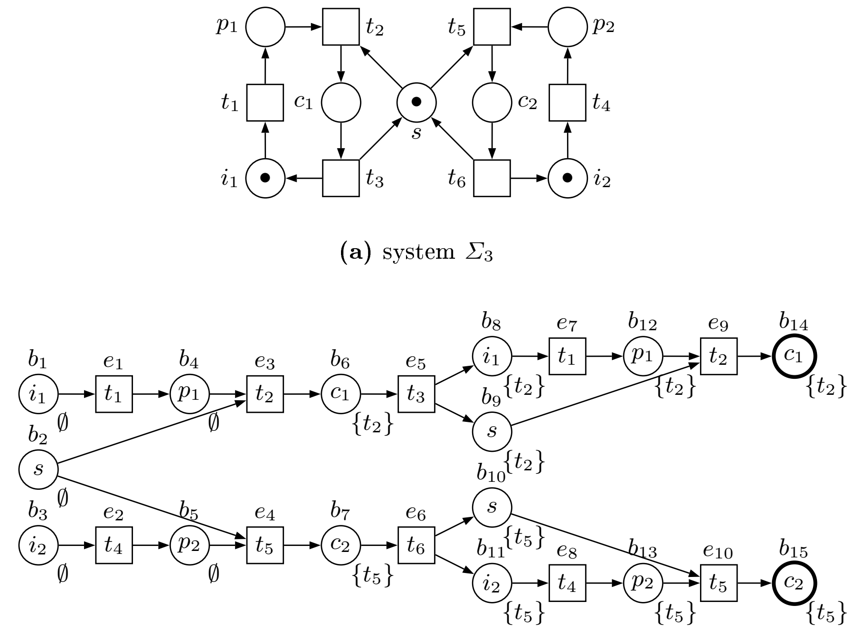
Using this new cut-off criterion, we can canonically define the finite prefix of the labeled branching process similarly to Definition 3. For system Σ_2 (c. f. Fig. 1(b)) this finite prefix corresponds to the maximal labeled branching process (c. f. Fig. 5).

Unfortunately, a proof for completeness cannot be given at this moment. However, the author did not encounter any counter-example yet.

As an example, we consider another system, Σ_3 (c. f. Fig. 6(a)). Its cc-unfolding (c. f. Fig. 6(b)) contains two cut-off conditions (depicted bold). The resulting finite prefix is complete (i. e., it represents all reachable markings of Σ_3). However, the cc-unfolding is not minimal as it contains more conditions and events than necessary. Using McMillan's cut-off criterion, the events e_5 and e_6 would have been detected as cut-off events.

Nevertheless, even without generating the minimal prefix, the local cut-off criterion still has a lower memory requirement: instead of storing eight reachable markings covering 20 places[5], the cc-unfolding only stores 15 conditions, each covering one place.

[5] As an example, the markings $\{i_1, i_1, s\}$ and $\{i_1, p_1, s\}$ together cover six places.



(a) system Σ_3

(b) cc-unfolding of Σ_3

Fig. 6. The system Σ_3 (a) and its cc-unfolding using the new cut-off criterion (b).

4 Open Tasks

The work presented so far is at its very beginning. Only a part of the first goal is achieved: formal proofs are only given for small class of systems, namely those without join conditions. For arbitrary finite safe systems, an intuitive cut-off criterion together with a completeness proof have to be elaborated. Furthermore, the enhanced cut-off criterion (c. f. Def. 5) yields to unfoldings which are in most cases larger than McMillan's unfoldings. An improvement of the local cut-off criterion similar to [10] could result in smaller cc-unfoldings.

References

1. McMillan, K.L.: Using Unfoldings to Avoid the State Explosion Problem in the Verification of Asynchronous Circuits. In von Bochmann, G., Probst, D.K., eds.: CAV. Volume 663 of Lecture Notes in Computer Science., Springer-Verlag (1992) 164–177
2. Valmari, A.: The State Explosion Problem. In Reisig, W., Rozenberg, G., eds.: Petri Nets. Volume 1491 of Lecture Notes in Computer Science., Springer-Verlag (1996) 429–528
3. Semenov, A., Yakovlev, A.: Verification of Asynchronous Circuits using Time Petri-Net Unfolding. In: DAC, ACM Press (1996) 59–62

4. Desel, J., Juhás, G., Neumair, C.: Finite Unfoldings of Unbounded Petri Nets. In Cortadella, J., Reisig, W., eds.: ICATPN. Volume 3099 of Lecture Notes in Computer Science., Springer-Verlag (2004) 157–176
5. Vogler, W., Semenov, A.L., Yakovlev, A.: Unfolding and Finite Prefix for Nets with Read Arcs. In Sangiorgi, D., de Simone, R., eds.: CONCUR. Volume 1466 of Lecture Notes in Computer Science., Springer-Verlag (1998) 501–516
6. Esparza, J.: Model Checking Using Net Unfoldings. In Gaudel, M.C., Jouannaud, J.P., eds.: TAPSOFT. Volume 668 of Lecture Notes in Computer Science., Springer-Verlag (1993) 613–628
7. McMillan, K.L.: A Technique of State Space Search Based on Unfolding. Formal Methods in System Design **6**(1) (1995) 45–65
8. Esparza, J., Heljanko, K.: Implementing LTL Model Checking with Net Unfoldings. In Dwyer, M.B., ed.: SPIN. Volume 2057 of Lecture Notes in Computer Science., Springer-Verlag (2001) 37–56
9. Khomenko, V., Koutny, M.: Towards an Efficient Algorithm for Unfolding Petri Nets. In Larsen, K.G., Nielsen, M., eds.: CONCUR. Volume 2154 of Lecture Notes in Computer Science Volume., Springer-Verlag (2001) 366–380
10. Esparza, J., Römer, S., Vogler, W.: An Improvement of McMillan's Unfolding Algorithm. In Margaria, T., Steffen, B., eds.: TACAS. Volume 1055 of Lecture Notes in Computer Science., Springer-Verlag (1996) 87–106
11. Khomenko, V., Koutny, M., Vogler, W.: Canonical Prefixes of Petri Net Unfoldings. In Brinksma, E., Larsen, K.G., eds.: CAV. Volume 2404 of Lecture Notes in Computer Science., Springer-Verlag (2002) 582–595
12. Heljanko, K.: Minimizing Finite Complete Prefixes. In Burkhard, H.D., Czaja, L., Nguyen, H.S., Starke, P., eds.: CS&P, University of Warsaw (1999) 83–95
13. Khomenko, V., Kondratyev, A., Koutny, M., Vogler, W.: Merged Processes - A New Condensed Representation of Petri Net Behaviour. In Abadi, M., de Alfaro, L., eds.: CONCUR. Volume 3653 of Lecture Notes in Computer Science., Springer-Verlag (2005) 338
14. Schmidt, K.: How to Calculate Symmetries of Petri Nets. Acta Informatica **36**(7) (2000) 545–590
15. Schmidt, K.: Integrating Low Level Symmetries into Reachability Analysis. In Graf, S., Schwartzbach, M.I., eds.: TACAS. Volume 1785 of Lecture Notes in Computer Science., Springer-Verlag (2000) 315–330
16. Christensen, S., Kristensen, L.M., Mailund, T.: A Sweep-Line Method for State Space Exploration. In Margaria, T., Yi, W., eds.: TACAS. Volume 2031 of Lecture Notes in Computer Science., Springer-Verlag (2001) 450–464
17. Reisig, W.: Petri nets: an introduction. Springer-Verlag (1985)
18. Engelfriet, J.: Branching Processes of Petri Nets. Acta Informatica **28**(6) (1991) 575–591
19. Esparza, J.: Model Checking Using Net Unfoldings. Hildesheimer Informatik-Berichte 14/92, Universität Hildesheim (1992)