

Goal-Driven Unfolding of Petri Nets —

Appendices

Thomas Chatain¹ and Loïc Paulevé²

¹ LSV, ENS Cachan – INRIA – CNRS, France

² CNRS, LRI UMR 8623, Univ. Paris-Sud – CNRS, France

A Proof of Theorem 11

Let E be a minimal configuration of \mathcal{N} to the goal. We want to prove that $E \subseteq E_{\text{gd}}$. Given the definition of E_{gd} and given that E is causally closed, this is equivalent to proving that $E \cap E_{\text{Ignored}} = \emptyset$: trivially, if some $e \in E$ is also in E_{Ignored} , then $\lceil e \rceil \cap E_{\text{Ignored}}$ contains at least e , so it is nonempty, and by definition $e \notin E_{\text{gd}}$; conversely, if some $e \in E$ is not in E_{gd} , this is because $\lceil e \rceil \cap E_{\text{Ignored}} \neq \emptyset$, and since E is causally closed, $\lceil e \rceil \subseteq E$, which implies that $E \cap E_{\text{Ignored}} \neq \emptyset$.

Then, it remains to show that $E \cap E_{\text{Ignored}} = \emptyset$. Let $e = \langle C, t \rangle \in E$; we have to show that, for every $e' \in \lfloor e \rfloor$, $t \notin \text{Useless}(e')$. If $e' \notin E'$, then, by definition, $\text{Useless}(e') \stackrel{\text{def}}{=} \bigcup_{e'' \in \lfloor e' \rfloor} \text{Useless}(e'')$, which means that e' has a causal predecessor e'' which also satisfies $t \notin \text{Useless}(e'')$.

Select now an e' which is minimal w.r.t. causality. This eliminates the previous case, so we have $e' \in E'$ and $t \in \bigcup_{e'' \in \lfloor e' \rfloor} \text{Useless}(e'')$ and $t \notin \text{useless-trs}\left(\mathcal{N}, g, \text{Mark}(\lceil e' \rceil), \bigcup_{e'' \in \lfloor e' \rfloor} \text{Useless}(e'')\right)$. Assuming that useless-trs is a reduction procedure satisfying Definition 9, this implies that no minimal firing sequence from $\text{Mark}(\lceil e' \rceil)$ to the goal uses t , which contradicts the fact that E is a minimal configuration to the goal: Indeed, since $e' \in \lfloor e \rfloor$, there exists a linearization¹ $e_1, \dots, e_{|E|}$ of E in which the events in $\lceil e' \rceil$ occur before the others, i.e. $\{e_1, \dots, e_{|\lceil e' \rceil|}\} = \lceil e' \rceil$, $e_{|\lceil e' \rceil|} = e'$ and $\{e_{|\lceil e' \rceil|+1}, \dots, e_{|E|}\} = E \setminus \lceil e' \rceil$; then $t_{|\lceil e' \rceil|+1} \dots t_{|E|}$ (with $t_i \stackrel{\text{def}}{=} h(e_i)$ the transition corresponding to e_i) is a firing sequence from $\text{Mark}(\lceil e' \rceil)$ to the goal and it uses t . If it is not minimal, then because it has a feasible cycling permutation σ , then $t_1, \dots, t_{|\lceil e' \rceil|} \cdot \sigma$ is a feasible cycling permutation from M_0 to the goal, which contradicts the fact that E is a minimal configuration to the goal. ◀

B Proof of Lemma 12

Because the set of markings is finite and because $\text{Alt}(\mathcal{E})$ is also finite (computed on a finite prefix), procedure $\text{PUTATIVE-GD-PREFIX}(\mathcal{N}, \Delta)$ always terminates; moreover all the iterations in procedure $\text{POST-}\Delta(\Delta, \mathcal{P})$ are over finite sets. Finally, we prove that procedure $\text{GD-PREFIX}(\mathcal{N})$ terminates, i.e., after a finite number of iterations, $\text{POST-}\Delta(\Delta, \mathcal{P}) = \Delta$. First, by construction, $\forall c \in \Delta, c \in \Delta'$ and $\Delta'(c) \subseteq \Delta(c)$, with $\Delta' = \text{POST-}\Delta(\Delta, \mathcal{P})$. Then, remark that, due to the cut-off treatment, any event of any putative prefix has a bounded number of event ancestors (causal past): the number of reachable markings. Finally, because the branching up to a given depth is finite, only a finite number of events can be considered in any iteration of the putative prefix; hence the number of events registered in Δ is finite.

¹ A linearization of E is a total ordering of $e_1, \dots, e_{|E|}$ of the events in E such that for $e_i < e_j \implies i < j$.



licensed under Creative Commons License CC-BY

Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Therefore, due to the monotonicity of Δ modifications, the iterative procedure necessarily converges towards a unique finite prefix in a finite number of steps. \blacktriangleleft

C Proof of Theorem 13

We first show that for every configuration \mathcal{E} that can be extended to a minimal configuration to the goal, there exists a configuration in the goal-driven prefix which contains no cut-off event and reaches $Mark(\mathcal{E})$. The principle is the one used for completeness of classical finite prefixes defined using adequate orders: if \mathcal{E} contains no cut-off event, it is in the goal-driven prefix, since the construction of the *Useless* is more permissive in the goal-driven prefix (with the use of $Alt()$) than in the goal-driven unfolding. Now, if \mathcal{E} contains a cut-off event e (w.r.t. an event e' such that $\lceil e' \rceil \triangleleft \lceil e \rceil$ and $Mark(\lceil e' \rceil) = Mark(\lceil e \rceil)$), then \mathcal{E} can be decomposed as $\lceil e \rceil \uplus D$ and e' has an extension D' isomorphic to D . Then $\lceil e' \rceil \uplus D'$ is smaller than \mathcal{E} w.r.t. \triangleleft and reaches the same marking. This operation can be iterated if needed; it terminates because \triangleleft is well founded, and gives a configuration \mathcal{E}' without cut-offs which reaches the same marking as \mathcal{E} . If \mathcal{E} can be extended with an event f , then so can \mathcal{E}' with an event f' corresponding to the same transition $h(f') = h(f)$. The event f' is in the prefix but may be a cut-off.

It remains to make sure that the transitions of \mathcal{E} (plus $h(f)$) are not considered useless. For this, focus on $\lceil e \rceil \uplus D$ mapped to $\lceil e' \rceil \uplus D'$. For every event $d \in D$, let d' be the corresponding event in D' . We have $\lceil d \rceil \cup \lceil e \rceil \in Alt(\lceil d' \rceil)$ because the causal past of d' uses at least one condition from the cut of $\lceil e' \rceil$. This ensures that the transitions fired in D after $\lceil d \rceil \cup \lceil e \rceil$ are taken into account in the computation of the transitions allowed after d' (if d' is itself in the prefix, otherwise apply this inductively), ensuring that in the end $\mathcal{E}' \cup \{f'\}$ is in the goal-driven prefix. \blacktriangleleft

D Goal-oriented reduction of automata networks

This appendix reproduces the main definitions of the goal-oriented reduction of automata networks introduced in [1]. The interested reader should refer to this later reference for further details.

The reduction is defined on *asynchronous automata networks* and all minimal traces from the initial state to a state where an automaton g is in state \top , specifying the *goal*. The reduction relies on the static analysis of dependencies of local paths within individual automata, in order to gather the necessary conditions related to the other automata.

We first give the formal definition of asynchronous automata networks, where any transition modifies the state of one and only one automaton.

D.1 Automata networks

An Automata Network (AN) is a set of finite state machines where the local transitions can be conditioned with the state of other automata in the network.

► **Definition 1** (Automata Network (Σ, S, T)). An *Automata Network* (AN) is defined by a tuple (Σ, S, T) where

- Σ is the finite set of automata identifiers;
- For each $a \in \Sigma$, $S(a) = \{a_i, \dots, a_j\}$ is the finite set of local states of automaton a ;
 $S \triangleq \prod_{a \in \Sigma} S(a)$ is the finite set of global states;
- $\mathbf{LS} \triangleq \bigcup_{a \in \Sigma} S(a)$ denotes the set of all the local states.

- $T = \{a \mapsto T_a \mid a \in \Sigma\}$, where $\forall a \in \Sigma, T_a \subseteq S(a) \times 2^{\mathbf{LS} \setminus S(a)} \times S(a)$ with $(a_i, \ell, a_j) \in T_a \Rightarrow a_i \neq a_j$ and $\forall b \in \Sigma, |\ell \cap S(b)| \leq 1$, is the mapping from automata to their finite set of local transitions.

We note $a_i \xrightarrow{\ell} a_j \in T \stackrel{\Delta}{\Leftrightarrow} (a_i, \ell, a_j) \in T(a)$. Given $t = a_i \xrightarrow{\ell} a_j \in T$, $\text{orig}(t) \stackrel{\Delta}{=} a_i$, $\text{dest}(t) \stackrel{\Delta}{=} a_j$, $\text{enab}(t) \stackrel{\Delta}{=} \ell$, $\bullet t \stackrel{\Delta}{=} \{a_i\} \cup \ell$, and $t^\bullet \stackrel{\Delta}{=} \{a_j\} \cup \ell$.

At any time, each automaton is in one and only one local state, forming the global state of the network. Assuming an arbitrary ordering between automata identifiers, the set of global states of the network is referred to as S as a shortcut for $\prod_{a \in \Sigma} S(a)$. Given a global state $s \in S$, $s(a)$ is the local state of automaton a in s , i.e., the a -th coordinate of s . Moreover we write $a_i \in s \stackrel{\Delta}{\Leftrightarrow} s(a) = a_i$; and for any $ls \in 2^{\mathbf{LS}}$, $ls \subseteq s \stackrel{\Delta}{\Leftrightarrow} \forall a_i \in ls, s(a) = a_i$.

For this appendix, we consider the asynchronous application of local transitions. Given a state $s \in S$, a transition $t = a_i \xrightarrow{\ell} a_j \in T$ can be applied only if $\bullet t \subseteq s$. The application of it results in the state $s \cdot t$ where the local state of a has been replaced from a_i to a_j . Note that [1] allows more general semantics.

► **Definition 2 (Trace).** Given an AN (Σ, S, T) and a state $s \in S$, a *trace* π is a sequence of transitions T such that $\forall i \in [1; |\pi|], \bullet \pi^i \subseteq (s \cdot \pi^1 \dots \pi^{i-1})$.

The pre-condition $\bullet \pi$ and the post-condition π^\bullet are defined as follows: for all $n \in [1; |\pi|]$, for all $a_i \in \bullet \pi^n$, $a_i \in \bullet \pi \stackrel{\Delta}{\Leftrightarrow} \forall m \in [1; n-1], S(a) \cap \bullet \pi^m = \emptyset$; similarly, for all $n \in [1; |\pi|]$, for all $a_j \in \pi^n \bullet$, $a_j \in \pi^\bullet \stackrel{\Delta}{\Leftrightarrow} \forall m \in [n+1; m], S(a) \cap \pi^m \bullet = \emptyset$. If π is empty, $\bullet \pi = \pi^\bullet = \emptyset$.

The set of transitions composing a trace π is noted $\text{tr}(\pi) \stackrel{\Delta}{=} \{\pi^n \mid 1 \leq n \leq |\pi|\}$.

Given an automata network (Σ, S, T) and a state $s \in S$, the local state $g_\top \in \mathbf{LS}$ is *reachable* from s if and only if either $g_\top \in s$ or there exists a trace π with $\bullet \pi \subseteq s$ and $g_\top \in \pi^\bullet$. We consider a trace π for g_\top reachability from s is *minimal* if and only if there exists no sub-trace reaching g_\top .

► **Definition 3 (Minimal trace for local state reachability).** A trace π is *minimal* w.r.t. g_\top reachability from s if and only if there is no trace ϖ from s , $\varpi \neq \pi$, $|\varpi| < |\pi|$, $g_\top \in \varpi^\bullet$, such that there exists an injection $\phi : [1; |\varpi|] \rightarrow [1; |\pi|]$ with $\forall i, j \in [1; |\varpi|], i < j \Leftrightarrow \phi(i) < \phi(j)$ and $\varpi^i = \pi^{\phi(i)}$.

An automata network (Σ, S, T) can be straightforwardly encoded as a safe Petri net having groups of mutually exclusive places acting as the automata, and where each transition $t \in T$ of the AN is encoded as a Petri net transition with incoming arcs from $\text{orig}(t)$ and $\text{enab}(t)$, out-going arcs to $\text{dest}(t)$ and $\text{enab}(t)$.

D.2 Local Causality

Locally reasoning within one automaton a , the reachability of one of its local state a_j from some global state s with $s(a) = a_i$ can be described by a (local) *objective*, that we note $a_i \rightsquigarrow a_j$ (Def. 4).

► **Definition 4 (Objective).** Given an automata network (Σ, S, T) , an *objective* is a pair of local states $a_i, a_j \in S(a)$ of a same automaton $a \in \Sigma$ and is denoted $a_i \rightsquigarrow a_j$. The set of all objectives is referred to as $\mathbf{Obj} \stackrel{\Delta}{=} \{a_i \rightsquigarrow a_j \mid (a_i, a_j) \in S(a) \times S(a), a \in \Sigma\}$.

Given an objective $a_i \rightsquigarrow a_j \in \mathbf{Obj}$, $\text{local-paths}(a_i \rightsquigarrow a_j)$ is the set of local acyclic paths of transitions $T(a)$ within automaton a from a_i to a_j (Def. 5).

► **Definition 5** (local-paths). Given $a_i \rightsquigarrow a_j \in \mathbf{Obj}$, if $i = j$, $\text{local-paths}(a_i \rightsquigarrow a_i) \triangleq \{\varepsilon\}$; if $i \neq j$, a sequence η of transitions in $T(a)$ is in $\text{local-paths}(a_i \rightsquigarrow a_j)$ if and only if $|\eta| \geq 1$, $\text{orig}(\eta^1) = a_i$, $\text{dest}(\eta^{|\eta|}) = a_j$, $\forall n \in [1; |\eta| - 1]$, $\text{dest}(\eta^n) = \text{orig}(\eta^{n+1})$, and $\forall n, m \in [1; |\eta|], n > m \Rightarrow \text{dest}(\eta^n) \neq \text{orig}(\eta^m)$.

As stated by Property 1, any trace reaching a_j from a state containing a_i uses all the transitions of at least one local acyclic path in $\text{local-paths}(a_i \rightsquigarrow a_j)$.

► **Property 1.** For any trace π , for any $a \in \Sigma$, $a_i, a_j \in S(a)$, $1 \leq n \leq m \leq |\pi|$ where $a_i \in \bullet\pi^n$ and $a_j \in \pi^{m\bullet}$, there exists a local acyclic path $\eta \in \text{local-paths}(a_i \rightsquigarrow a_j)$ that is a sub-sequence of $\pi^{n..m}$, i.e., there is an injection $\phi : [1; |\eta|] \rightarrow [n; m]$ with $\forall u, v \in [1; |\eta|], u < v \Leftrightarrow \phi(u) < \phi(v)$ and $\eta^u \in \pi^{\phi(u)}$.

A local path is not necessarily a trace, as transitions may be conditioned by the state of other automata that may need to be reached beforehand. A local acyclic path being of length at most $|S(a)|$ with unique transitions, the number of local acyclic paths is polynomial in the number of transitions $T(a)$ and exponential in the number of local states in a .

D.3 Necessary condition for local reachability

Given an objective $a_i \rightsquigarrow a_j \in \mathbf{Obj}$, we give in Proposition 1 a definition of a predicate $\text{valid}_s(a_i \rightsquigarrow a_j)$ which is true if there exists a trace π from s such that $\exists m, n \in [1; |\pi|]$ with $m \leq n$, $a_i \in \bullet\pi^m$, and $a_j \in \pi^{n\bullet}$.

It is a simplified version of a necessary condition for reachability demonstrated in [2]. Essentially, the set of valid objectives Ω is built as follows: initially, it contains all the objectives of the form $a_i \rightsquigarrow a_i$ (that are always valid); then an objective $a_i \rightsquigarrow a_j$ is added to Ω only if there exists a local acyclic path $\eta \in \text{local-paths}(a_i \rightsquigarrow a_j)$ where all the objectives from the initial state s to the enabling conditions of the transitions are already in Ω : if $b_k \in \text{enab}(\eta^n)$ for some $n \in [1; |\eta|]$, then the objective $b_0 \rightsquigarrow b_k$ is already in the set, assuming $s(b) = b_0$.

► **Proposition 1.** For all objective $P \in \mathbf{Obj}$, $\text{valid}_s(P) \Leftrightarrow P \in \Omega$ where Ω is the least fixed point of the monotonic function $F : 2^{\mathbf{Obj}} \rightarrow 2^{\mathbf{Obj}}$ with

$$F(\Omega) \triangleq \{a_i \rightsquigarrow a_j \in \mathbf{Obj} \mid \exists \eta \in \text{local-paths}(a_i \rightsquigarrow a_j) : \\ \forall n \in [1; |\eta|], \forall b_k \in \text{enab}(\eta^n), s(b) \rightsquigarrow b_k \in \Omega\} .$$

D.4 Goal-oriented reduction procedure

The reduction procedure (Def. 6) consists in the set \mathcal{B} of objectives whose local acyclic paths may contribute to a minimal trace for the goal reachability. Given an objective, only the local paths where all the enabling conditions lead to valid objectives are considered (local-paths_s). The local transitions corresponding to the objectives in \mathcal{B} are noted $\text{tr}(\mathcal{B})$.

► **Definition 6** (Goal-oriented reduction). Given an AN (Σ, S, T) , an initial state s where, without loss of generality, $\forall a \in \Sigma, s(a) = a_0$, and a local state g_\top with $g \in \Sigma$ and $g_\top \in S(g)$, $\mathcal{B} \subseteq \mathbf{Obj}$ is the smallest set which satisfies the following conditions:

1. $g_0 \rightsquigarrow g_\top \in \mathcal{B}$
2. $b_j \xrightarrow{\ell} b_k \in \text{tr}(\mathcal{B}) \Rightarrow \forall a_i \in \ell, a_0 \rightsquigarrow a_i \in \mathcal{B}$
3. $b_j \xrightarrow{\ell} b_k \in \text{tr}(\mathcal{B}) \wedge b_\star \rightsquigarrow b_i \in \mathcal{B} \Rightarrow b_k \rightsquigarrow b_i \in \mathcal{B}$

with $\text{tr}(\mathcal{B}) \triangleq \bigcup_{P \in \mathcal{B}} \text{tr}(\text{local-paths}_s(P))$, where, $\forall P \in \mathbf{Obj}$,

$\text{local-paths}_s(P) \triangleq \{\eta \in \text{local-paths}(P) \mid \forall n \in [1; |\eta|], \forall b_k \in \text{enab}(\eta^n), \mathbf{valid}_s(b_0 \rightsquigarrow b_k)\}$.

► **Theorem 7.** *For each minimal trace π reaching g_\top from s , $\text{tr}(\pi) \subseteq \text{tr}(\mathcal{B})$.*

One can then define $\text{useless-trs}(\mathcal{N}, g_\top) \triangleq T \setminus \text{tr}(\mathcal{B})$ where \mathcal{N} is the Petri net corresponding to the AN (Σ, S, T) with initial state s .

References

- 1 Loïc Paulevé. Goal-Oriented Reduction of Automata Networks. In *CMSB 2016 - 14th conference on Computational Methods for Systems Biology*, volume 9859 of *Lecture Notes in Bioinformatics*. Springer, 2016. doi:10.1007/978-3-319-45177-0_16.
- 2 Loïc Paulevé, Morgan Magnin, and Olivier Roux. Static analysis of biological regulatory networks dynamics using abstract interpretation. *Mathematical Structures in Computer Science*, 22(04):651–685, 2012. doi:10.1017/S0960129511000739.