Goal-Oriented Reduction of Automata Networks

Loïc Paulevé

Goal-Oriented Reduction of Automata Networks

Loïc Paulevé

LRI UMR 8623, Univ. Paris-Sud – CNRS
Université Paris-Saclay, 91405 Orsay, France
loic.pauleve@lri.fr

Abstract

We consider networks of finite-state machines having local transitions conditioned by the current state of other automata. In this paper, we depict a reduction procedure tailored for a given reachability property of the form "from global state s, there exists a sequence of transitions leading to a state where an automaton g is in a local state $\top$". By exploiting a causality analysis of the transitions within the individual automata, the proposed reduction removes local transitions while preserving *all* the minimal traces that satisfy the reachability property. The complexity of the procedure is polynomial in the total number of local transitions, and exponential with the maximal number of local states within an automaton. Applied to automata networks modelling dynamics of biological systems, we observe that the reduction shrinks down significantly the reachable state space, enhancing the tractability of the model-checking of large networks.

1 Introduction

Automata networks model dynamical systems resulting from simple interactions between entities. Each entity is typically represented by an automaton with few internal states which evolve subject to the state of a narrow range of other entities in the network. Richness of emerging dynamics arises from several factors including the topology of the interactions, the presence of feedback loop, and the concurrency of transitions.

Automata networks are notably used to model biological systems, notably signalling networks or gene regulatory networks (e.g., [1, 10, 15, 21, 31, 32, 33, 37]). In such a context, it is crucial to confront the dynamics of a model to actual biological knowledge to validate or refute the model, and infer well-founded hypotheses on the important mechanisms controlling the emerging behaviours.

Part of those properties relate to the reachability problem (PSPACE-complete [7]), for instance the (im)possibility of activation of an entity from a given set of initial states with a potential set of perturbations. Due to the increasing precision of biological knowledge, models of networks become larger and larger and can gather hundreds to thousands of interacting entities making the formal analysis of their dynamics a challenging task.

Facing a model too large for a raw exhaustive analysis, a natural approach is to reduce its dynamics while preserving important properties. Multiple approaches, often complementary, have been explored since decades to address such a challenge in dynamical and concurrent systems [36, 22, 24]. Among those, structural reductions attempt to reduce the model specification while obtaining bisimulation, e.g., [34] on Petri nets; or preserving a subset of dynamical features and properties, e.g., the cone of influence reduction [5] which delimit the variables of the model having an influence of a given property, or Petri net transition agglomerations which preserve liveness and LTL properties [4, 17]. In the scope of rule-based models of biological networks, efficient static analysis methods have been developed to lump numerous global states of the systems based on the fragmentation of interacting components [14]; and to *a posteriori* compress simulated traces to obtain compact witnesses of dynamical properties [12]. Reductions preserving the attractors of dynamics (long-term/steady-state behaviour) have also been studied for chemical reaction networks [25] and Boolean networks [26]. The latter approach applies to formalisms close to automata networks but does not preserve reachability properties.

Contribution We introduce a reduction of automata networks which identifies transitions that do not contribute to a given reachability property and hence can be ignored. The considered automata networks are finite sets of finite-state machines where transitions between their local states are conditioned by the state of other automata in the network. We use a general concurrent semantics where any number of automata can apply one transition within one step. We call a *trace* a sequential interleaved execution of steps.

Our reduction preserves all the minimal traces satisfying reachability properties of the form "from state s there exist successive steps that lead to a state where a given automaton g is in local state $g_\top$". A trace is *minimal* if no step nor transition can be removed from it and resulting in a sub-trace that satisfies the concerned reachability property. The complexity of the procedure is polynomial in the number of local transitions, and exponential in the maximal size of automata. Therefore, the reduction is scalable for networks of multiple automata, where each have a few local states.

The identification of the transitions that are not part of any minimal trace is performed by a static analysis of the causality of transitions within automata. It extends previous static analysis of reachability properties by abstract interpretation [29, 28]. In [29], necessary or sufficient conditions for reachability are derived, but they do not allow to capture all the (minimal) traces towards a reachability goal. In [28], the static analysis extracts local states, referred to as cut-sets, which are necessarily reached prior to a given reachability goal. The results presented here are orthogonal: we identify transitions that are never part of a minimal trace for the given reachability property. It allows us to output a reduced model where all such transitions are removed while preserving all the minimal traces for reachability. Hence, whereas [28] focuses on identifying necessary conditions for reachability, this article focuses on preserving sufficient conditions for reachability.

The effectiveness of our goal-oriented reduction is experimented on actual models of biological networks and show significant shrinkage of the dynamics of the automata networks, enhancing the tractability of a concrete verification. Compared to other model reductions, our goal is somehow close to the cone of influence reduction mentioned above, which identifies variables that do not impact a given property. Here, our approach offers a much more fine-grained analysis in order to identify the sufficient transitions and values of variables that contribute to the property.

Outline Section 2 sets up the definition and semantics of the automata networks considered in this paper, together with the local causality analysis for reachability properties, based on prior work. Section 3 first depicts a necessary condition using local causality analysis for satisfying a reachability property and then introduce the goal-oriented reduction with the proof of minimal traces preservation. Section 4 shows the efficiency of the reduction on a range of biological networks. Finally, section 5 discusses the results and motivates further work.

Notations Integer ranges are noted $[m; n] \triangleq \{m, m+1, \cdots, n\}$. Given a finite set A, $|A|$ is the cardinality of A; 2^A is the power set of A. Given $n \in \mathbb{N}$, $x = (x^i)_{i \in [1;n]}$ is a sequence of elements indexed by $i \in [1; n]$; $|x| = n$; $x^{m..n}$ is the subsequence $(x^i)_{i \in [m;n]}$; $x :: e$ is the sequence x with an additional element e at the end; ε is the empty sequence.

2 Automata Networks and Local Causality

2.1 Automata Networks

We declare an Automata Network (AN) by a finite set of finite-state machines having transitions between their local states conditioned by the state of other automata in the network. An AN is defined by a triple (Σ, S, T) (definition 1) where Σ is the set of automata identifiers; S associates to each automaton a finite set of local states: if $a \in \Sigma$, $S(a)$ refers to the set of local states of a; and T associates to each automaton its local transitions. Each local state is written of the form a_i, where $a \in \Sigma$ is the automaton in which the state belongs to, and i is a unique identifier; therefore given $a_i, a_j \in S(a)$, $a_i = a_j$ if and only if a_i and a_j refer to the same local state of the automaton a. For each automaton $a \in \Sigma$, $T(a)$ refers to the set of transitions of the form $t = a_i \xrightarrow{\ell} a_j$ with $a_i, a_j \in S(a)$, $a_i \neq a_j$, and ℓ the enabling condition of t, formed by a (possibly empty) set of local states of automata different than a and containing at most one local state of each automaton. The *pre-condition* of transition t, noted ${}^\bullet t$, is the set composed of a_i and of the local states in ℓ; the *post-condition*, noted $t^\bullet$ is the set composed of a_j and of the local states in ℓ.

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 $\forall a \in \Sigma, T_a \subseteq S(a) \times 2^{\mathsf{LS}\setminus S(a)} \times S(a)$ 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 $a_i \xrightarrow{\ell} a_j \in T \overset{\Delta}{\Leftrightarrow} (a_i, \ell, a_j) \in T(a)$ and $a_i \rightarrow a_j \in T \overset{\Delta}{\Leftrightarrow} \exists \ell \in 2^{\mathsf{LS}\setminus S(a)}, a_i \xrightarrow{\ell} a_j \in T$. Given $t = a_i \xrightarrow{\ell} a_j \in T$, $\mathrm{orig}(t) \triangleq a_i$, $\mathrm{dest}(t) \triangleq a_j$, $\mathrm{enab}(t) \triangleq \ell$, ${}^\bullet t \triangleq \{a_i\} \cup \ell$, and $t^\bullet \triangleq \{a_j\} \cup \ell$.

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 $a_i \in s \overset{\Delta}{\Leftrightarrow} s(a) = a_i$; and for any $ls \in 2^{\mathsf{LS}}$, $ls \subseteq s \overset{\Delta}{\Leftrightarrow} \forall a_i \in ls, s(a) = a_i$.

In the scope of this paper, we allow, but do not enforce, the parallel application of transitions in different automata. This leads to the definition of a *step* as a set of transitions, with at most one transition per automaton (definition 2). For notational convenience, we allow empty steps. The pre-condition (resp. post-condition) of a step τ, noted ${}^\bullet\tau$ (resp. $\tau^\bullet$), extends the similar notions on transitions: the pre-condition (resp. post-condition) is the union of the pre-conditions (resp. post-conditions) of composing transitions. A step τ is *playable* in a state $s \in S$ if and only if ${}^\bullet\tau \subseteq s$, i.e., all the local states in the pre-conditions of transitions are in s. If τ is playable in s, $s \cdot \tau$ denotes the state after the applications of all the transitions in τ, i.e., where for each transition $a_i \xrightarrow{\ell} a_j \in \tau$, the local state of automaton a has been replaced with a_j.

Definition 2 (Step)**.** Given an AN (Σ, S, T), a *step* τ is a subset of local transitions T such that for each automaton $a \in \Sigma$, there is at most one local transition $T(a)$ in τ $(\forall a \in \Sigma, |(\tau \cap T(a))| \leq 1)$.
We note ${}^\bullet\tau \triangleq \bigcup_{t\in\tau} {}^\bullet t$ and $\tau^\bullet \triangleq \bigcup_{t\in\tau} t^\bullet \setminus \{\mathrm{orig}(t) \mid t \in \tau\}$.
Given a state $s \in S$ where τ is playable (${}^\bullet\tau \subseteq s$), $s \cdot \tau$ denotes the state where $\forall a \in \Sigma$, $(s \cdot \tau)(a) = a_j$ if $\exists a_i \rightarrow a_j \in \tau$, and $(s \cdot \tau)(a) = s(a)$ otherwise.

Remark that $\tau^\bullet \subseteq s \cdot \tau$ and that this definition implicitly rules out steps composed of incompatible transitions, i.e., where different local states of a same automaton are in the pre-condition.

A *trace* (definition 3) is a sequence of successively playable steps from a state $s \in S$. The pre-condition ${}^\bullet\pi$ of a trace π is the set of local states that are required to be in s for applying π (${}^\bullet\pi \subseteq s$); and the post-condition $\pi^\bullet$ is the set of local states that are present in the state after the full application of π ($\pi^\bullet \subseteq s \cdot \pi$).

Definition 3 (Trace)**.** Given an AN (Σ, S, T) and a state $s \in S$, a *trace* π is a sequence of steps such that $\forall i \in [1; |\pi|]$, ${}^\bullet\pi^i \subseteq (s \cdot \pi^1 \cdot \dots \cdot \pi^{i-1})$.
The pre-condition ${}^\bullet\pi$ and the post-condition $\pi^\bullet$ are defined as follows: for all $n \in [1; |\pi|]$, for all $a_i \in {}^\bullet\pi^n$, $a_i \in {}^\bullet\pi \overset{\Delta}{\Leftrightarrow} \forall m \in [1; n-1], S(a) \cap {}^\bullet\pi^m = \emptyset$; similarly, for all $n \in [1; |\pi|]$, for all $a_j \in \pi^{n\bullet}$, $a_j \in \pi^\bullet \overset{\Delta}{\Leftrightarrow} \forall m \in [n+1; m], S(a) \cap \pi^{m\bullet} = \emptyset$. If π is empty, ${}^\bullet\pi = \pi^\bullet = \emptyset$.
The set of transitions composing a trace π is noted $\mathrm{tr}(\pi) \triangleq \bigcup_{n=1}^{|\pi|} \pi^n$.

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 different trace reaching $g_\top$ having each successive step being a subset of a step in π with the same ordering (definition 4). Say differently, a trace is minimal for $g_\top$ reachability if no step or transition can be removed from it without breaking the trace validity or $g_\top$ reachability.

Definition 4 (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, $\varpi \neq \pi$, $|\varpi| \leq |\pi|$, $g_\top \in \varpi^\bullet$, such that there exists an injection $\phi : [1; |\varpi|] \to [1; |\pi|]$ with $\forall i, j \in [1; |\varpi|]$, $i < j \Leftrightarrow \phi(i) < \phi(j)$ and $\varpi^i \subseteq \pi^{\phi(i)}$.

Automata networks as presented can be considered as a class of 1-safe Petri Nets [3] (at most one token per place) having groups of mutually exclusive places, acting as the automata, and where each transition has one and only one incoming and out-going arc and any number of read arcs. The semantics considered in this paper where transitions within different automata can be applied simultaneously echoes with Petri net step-semantics and concurrent/maximally concurrent semantics [20, 30, 19]. In the Boolean network community, such a semantics is referred to as the asynchronous generalized update schedule [2].

2.2 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 $a_i \rightsquigarrow a_j$ (definition 5).

Definition 5 (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 $\mathbf{Obj} \triangleq \{a_i \rightsquigarrow a_j \mid (a_i, a_j) \in S(a) \times S(a), a \in \Sigma\}$.

Given an objective $a_i \rightsquigarrow a_j \in \mathbf{Obj}$, 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 (definition 6).

Definition 6 (local-paths)**.** For each objective $a_i \rightsquigarrow a_j \in \mathbf{Obj}$, if $i = j$, local-paths$(a_i \rightsquigarrow a_i) \triangleq \{\varepsilon\}$; if $i \neq j$, a sequence η of transitions in $T(a)$ is in local-paths$(a_i \rightsquigarrow a_j)$ if and only if $|\eta| \geq 1$, orig$(\eta^1) = a_i$, dest$(\eta^{|\eta|}) = a_j$, $\forall n \in [1; |\eta| - 1]$, dest$(\eta^n) =$ orig(η^{n+1}), and $\forall n, m \in [1; |\eta|], n > m \Rightarrow$ dest$(\eta^n) \neq$ orig(η^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 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 $\eta \in$ local-paths$(a_i \rightsquigarrow a_j)$ that is a sub-sequence of $\pi^{n..m}$, i.e., there is an injection $\phi : [1; |\eta|] \to [n; m]$ 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.

Example 1. Let us consider the automata network (Σ, S, T), graphically represented in figure 1, where:

$$\Sigma = \{a, b, c, d\}$$

$$S(a) = \{a_0, a_1\} \qquad T(a) = \{a_0 \xrightarrow{\{b_0\}} a_1, a_1 \xrightarrow{\emptyset} a_0\}$$

$$S(b) = \{b_0, b_1\} \qquad T(b) = \{b_0 \xrightarrow{\{a_1\}} b_1, b_1 \xrightarrow{\{a_0\}} b_0\}$$

$$S(c) = \{c_0, c_1, c_2\} \qquad T(c) = \{c_0 \xrightarrow{\{a_1\}} c_1, c_1 \xrightarrow{\{b_1\}} c_0, c_1 \xrightarrow{\{b_0\}} c_2, c_0 \xrightarrow{\{d_1\}} c_2\}$$

$$S(d) = \{d_0, d_1\} \qquad T(d) = \emptyset$$

The local paths for the objective $c_0 \rightsquigarrow c_2$ are local-paths$(c_0 \rightsquigarrow c_2) = \{c_0 \xrightarrow{\{a_1\}} c_1 \xrightarrow{\{b_0\}} c_2, c_0 \xrightarrow{\{d_1\}} c_2\}$. From the state $\langle a_0, b_0, c_0, d_0 \rangle$, instances of traces are
$\{a_0 \xrightarrow{\{b_0\}} a_1\} :: \{b_0 \xrightarrow{\{a_1\}} b_1, c_0 \xrightarrow{\{a_1\}} c_1\} :: \{a_1 \xrightarrow{\emptyset} a_0\} :: \{b_1 \xrightarrow{\{a_0\}} b_0\} :: \{c_1 \xrightarrow{\{b_0\}} c_2\}$;
$\{a_0 \xrightarrow{\{b_0\}} a_1\} :: \{c_0 \xrightarrow{\{a_1\}} c_1\} :: \{c_1 \xrightarrow{\{b_0\}} c_2\}$;
the latter only being a minimal trace for c_2 reachability.

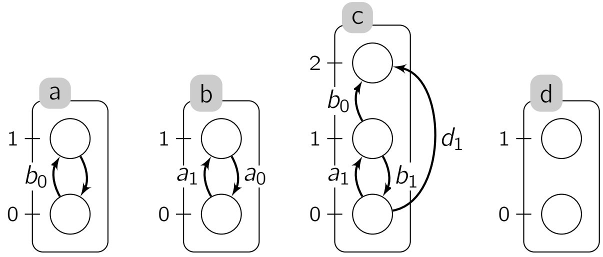

Figure 1: An example of automata network. Automata are represented by labelled boxes, and local states by circles where ticks are their identifier within the automaton – for instance, the local state a_0 is the circle ticked 0 in the box a. A transition is a directed edge between two local states within the same automaton. It can be labelled with a set of local states of other automata. In this example, all the transitions are conditioned by at most one other local state.

3 Goal-Oriented Reduction

Assuming a global AN (Σ, S, T), an initial state $s \in S$ and a reachability goal $g_\top$ where $g \in \Sigma$ and $g_\top \in S(g)$, the goal-oriented reduction identifies a subset of local transitions T that are sufficient for producing all the minimal traces leading to $g_\top$ from s. The reduction procedure takes advantage of the local causality analysis both to fetch the transitions that matter for the reachability goal and to filter out objectives that can be statically proven impossible.

3.1 Necessary condition for local reachability

Given an objective $a_i \rightsquigarrow a_j$ and a global state $s \in S$ where $s(a) = a_i$, prior work have demonstrated necessary conditions for the existence of a trace leading to a_j from s [29, 28]. Those necessary conditions rely on the local causality analysis defined in previous section for extracting necessary steps that have to be performed in order to reach the concerned local state.

Several necessary conditions have been established in [29], taking into account several features captured by the local paths (dependencies, sequentiality, partial order constraints, ...). The complexity of deciding most of these necessary conditions is polynomial in the total number of local transitions and exponential in the maximum number of local states within an automaton.

In this section, we consider a generic reachability over-approximation predicate $\mathbf{valid}_s$ which is false only when applied to an objective that has no trace concretizing it from s: a_j is reachable from s with $s(a) = a_i$ only if $\mathbf{valid}_s(a_i \rightsquigarrow a_j)$.

Definition 7 ($\mathbf{valid}_s$)**.** Given any objective $a_i \rightsquigarrow a_j \in \mathbf{Obj}$, $\mathbf{valid}_s(a_i \rightsquigarrow a_j)$ 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 $a_j \in \pi^{n\bullet}$.

For the sake of self-consistency, we give in proposition 1 an instance implementation of such a predicate. It is a simplified version of a necessary condition for reachability demonstrated in [29]. 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 $P \in \mathbf{Obj}$, $\mathbf{valid}_s(P) \overset{\Delta}{\Leftrightarrow} P \in \Omega$ where Ω is the least fixed point of the monotonic function $\mathrm{F} : 2^{\mathbf{Obj}} \to 2^{\mathbf{Obj}}$ with

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

Applied to the AN of figure 1, if $s = \langle a_0, b_0, c_0, d_0 \rangle$, $\mathbf{valid}_s(c_0 \rightsquigarrow c_2)$ is true because $c_0 \xrightarrow{a_1} c_1 \xrightarrow{b_0} c_2 \in \text{local-paths}(c_0 \rightsquigarrow c_2)$ with $\mathbf{valid}_s(a_0 \rightsquigarrow a_1)$ true and $\mathbf{valid}_s(b_0 \rightsquigarrow b_0)$ true. On the other hand, $\mathbf{valid}_s(d_0 \rightsquigarrow d_1)$ is false.

Note that Proposition 1 is an instance of $\mathbf{valid}_s$ implementation; any other implementation satisfying definition 7 can be used to apply the reduction proposed in this article. In [29], more restrictive over-approximations are proposed.

3.2 Reduction procedure

This section depicts the goal-oriented reduction procedure which aims at identifying transitions that do not take part in any minimal trace from the given initial state to the goal local state $g_\top$. The reduction relies on the local causality analysis to delimit local paths that may be involved in the goal reachability: any local transitions that is not captured by this analysis can be removed from the model without affecting the minimal traces for its occurrence.

The reduction procedure (definition 8) consists in collecting a set $\mathcal{B}$ of objectives whose local acyclic paths may contribute to a minimal trace for the goal reachability. To ease notations, and without loss of generality, we assume that any automaton a is in state a_0 in s. 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 $\mathrm{tr}(\mathcal{B})$.

Initially starting with the main objective $g_0 \rightsquigarrow g_\top$ (definition 8(1)), the procedure iteratively collects objectives that may be involved for the enabling conditions of local paths of already collected objectives. If a transition $b_j \xrightarrow{\ell} b_k$ is in $\mathrm{tr}(\mathcal{B})$, for each $a_i \in \ell$, the objective $a_0 \rightsquigarrow a_i$ is added in $\mathcal{B}$ (definition 8(2)); and for each other objective $b_\star \rightsquigarrow b_i \in \mathcal{B}$, the objective $b_k \rightsquigarrow b_i$ is added in $\mathcal{B}$ (definition 8(3)). Whereas the former criteria references the objectives required for concretizing a local path from the initial state, the later criteria accounts for the possible interleaving and successions of local paths within a same automaton: e.g., $g_\top$ reachability may require to reach b_k and b_i in some (undefined) order, we then consider 4 objectives: $b_0 \rightsquigarrow b_k$, $b_k \rightsquigarrow b_i$, $b_0 \rightsquigarrow b_i$, and $b_i \rightsquigarrow b_k$.

Definition 8 ($\mathcal{B}$)**.** 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. $b_j \xrightarrow{\ell} b_k \in \mathrm{tr}(\mathcal{B}) \Rightarrow \forall a_i \in \ell, a_0 \rightsquigarrow a_i \in \mathcal{B}$
3. $b_j \xrightarrow{\ell} b_k \in \mathrm{tr}(\mathcal{B}) \wedge b_\star \rightsquigarrow b_i \in \mathcal{B} \Rightarrow b_k \rightsquigarrow b_i \in \mathcal{B}$

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

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

$\mathrm{enab}(t)$ being the enabling condition of local transition t (definition 1).

Theorem 1 states that any trace which is minimal for the reachability of $g_\top$ from initial state s is composed only of transitions in $\mathrm{tr}(\mathcal{B})$. The proof is given in appendix A. It results that the AN $(\Sigma, S, \mathrm{tr}(\mathcal{B}))$ contains less transitions but preserves all the minimal traces for the reachability of the goal.

Theorem 1. *For each* minimal *trace* π *reaching* $g_\top$ *from* s, $\mathrm{tr}(\pi) \subseteq \mathrm{tr}(\mathcal{B})$.

Figure 2 shows the results of the reduction on the example AN of figure 1 for the reachability of c_2 from the state where all automata start at 0. Basically, the local path from c_0 to c_2 using d_1 being impossible to concretize (because $\mathbf{valid}_s(d_0 \rightsquigarrow d_1)$ is false), it has been removed, and consequently, so are the transitions involving b_1 as b_1 is not required for c_2 reachability.

Because the number of objectives is polynomial ($|\mathbf{Obj}| = \sum_{a\in\Sigma} |S(a)|^2$), the computation of $\mathcal{B}$ and $\mathrm{tr}(\mathcal{B})$ is very efficient, both from a time and space complexity point of view. The sets $\mathcal{B} \subseteq \mathbf{Obj}$ and $\mathrm{tr}(\mathcal{B}) \subseteq T$ can be built iteratively, from the empty sets: when a new objective $b_\star \rightsquigarrow b_i$ is inserted in $\mathcal{B}$, each transition

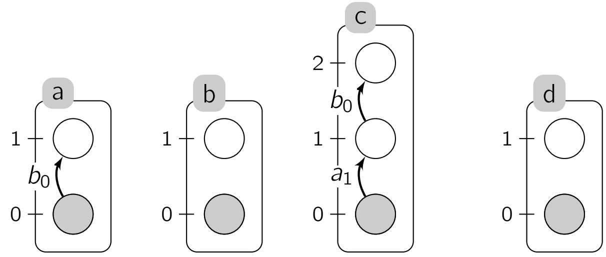

Figure 2: Reduced automata network from figure 1 for the reachability of c_2 from initial state indicated in grey.

in $\mathrm{tr}(\text{local-paths}_s(b_\star \rightsquigarrow b_i))$ is added in $\mathrm{tr}(\mathcal{B})$, if not already in; and for each transition $b_j \to b_k$ currently in $\mathrm{tr}(\mathcal{B})$, the objective $b_k \rightsquigarrow b_i$ is added in $\mathcal{B}$, if not already in. When a new transition $b_j \xrightarrow{\ell} b_k$ is added in $\mathrm{tr}(\mathcal{B})$, for each $a_i \in \ell$, the objective $a_0 \rightsquigarrow a_i$ is added in $\mathcal{B}$, if not already in; and for each objective $b_\star \rightsquigarrow b_i$ currently in $\mathcal{B}$, the objective $b_k \rightsquigarrow b_i$ is added in $\mathcal{B}$, if not already in.

Putting aside the $\mathrm{tr}(\text{local-paths}_s)$ computation, the above steps require a polynomial time and a linear space with respect to the number of transitions and objectives. The computation of $\mathrm{tr}(\text{local-paths}_s(a_i \rightsquigarrow a_j))$ requires a time exponential with the number of local states in automaton a ($|S(a)|$), due to the number of acyclic local paths (section 2.2), but a quadratic space: indeed, each individual local acyclic path does not need to be stored, only its set of local transitions, without conditions. Then, $\mathbf{valid}_s$ is called at most once per objective. We assume that the complexity of $\mathbf{valid}_s$ is polynomial with the number of automata and transitions and exponential with the maximum number of local states within an automaton (it is the case of the one presented in section 3.1)

Overall, the reduction procedure has a polynomial space complexity ($|\mathbf{Obj}| + |T|$) and time complexity polynomial with the total number of automata and local transitions, and exponential with the maximum number k of local states within an automaton ($k = \max_{a \in \Sigma} |S(a)|$). Therefore, assuming $k \ll |\Sigma|$, the goal-oriented reduction offers a very low complexity, especially with regard to a full exploration of the $k^{|\Sigma|}$ states.

4 Experiments

We experimented the goal-oriented reduction on several biological networks and quantify the shrinkage of the reachable state space. Then, we illustrate potential applications with the verification of simple reachability, and of cut sets. In both cases, the reduction drastically increases the tractability of those applications.

4.1 Results on model reduction

We conducted experiments on Automata Networks (ANs) that model dynamics of biological networks. For different initial states, and for different reachability goals, we compared the number of local transitions in the AN specifications ($|T|$), the number of reachable states, and the size of the so-called complete finite prefix of the unfolding of the net [13]. This latter structure is a finite partial order representation of all the possible traces, which is well studied in concurrency theory. It aims at offering a compact representations of the reachable state spaces by exploiting the concurrency between transitions: if t_1 and t_2 are playable in a given state and are not in conflict (notably when ${}^\bullet t_1 \cap {}^\bullet t_2 = \emptyset$), a standard approach would consider 4 global transitions (t_1 then t_2, and t_2 then t_1), whereas a partial order structure would simply declare t_1 and t_2 as concurrent, imposing no ordering between them. Hence, unfoldings drop part of the combinatorial explosion of the state space due to the interleaving of concurrent transitions.

The selected networks are models of signalling pathways and gene regulatory networks: two Boolean models of Epidermal Growth Factor receptors (EGF-r) [32, 33], one Boolean model of tumor cell invasion (Wnt) [10], two Boolean models of T-Cell receptor (TCell-r) [21, 31], one Boolean model of Mitogen-Activated Protein Kinase network (MAPK) [15], one multi-valued model of fate determination in the Vulval Precursor Cells (VPC) in C. elegans [37], one Boolean model of T-Cell differentiation (TCell-d) [1], and one

Boolean models of cell cycle regulation (RBE2F) [11]. The ANs result from automatic translation from the logical network specifications in the above references; for most models using the `logicalmodel` tool [16]. Note that the obtained ANs are bisimilar to the logical networks [6]. For each of these models, we selected initial states and nodes for which the activation will be the reachability goal[1].

Table 1 sums up the results before and after the goal-oriented reduction. The number of reachable states is computed with `its-reach` [23] using a symbolic representation, and the size of the complete finite prefix (number of instances of transitions) is computed with `Mole` [35]. The goal-oriented reduction is performed using `Pint` [27]. In each case, the reduction step took less than 0.1s, thanks to its very low complexity when applied to logical networks.

There is a substantial shrinkage of the dynamics for the reduced models, which can turn out to be drastic for large models. In some cases, the model is too large to compute the state space without reduction. For some large models, the unfolding is too large to be computed, whereas it can provide a very compact representation compared to the state space for large networks exhibiting a high degree of concurrency (e.g., TCell-d, RBE2F). In the case of first profile of TCell-d and EGF-r (104) the reduction removed all the transitions, resulting in an empty model. Such a behaviour can occur when the local causality analysis statically detect that the reachability goal is impossible, i.e., the necessary condition of section 3.1 is not satisfied. On the other hand, a non-empty reduced model does not guarantee the goal reachability. Appendix B show additional results with the reduction made without the filtering $\mathbf{valid}_s$ (section 3.1).

4.2 Example of application: goal reachability

In order to illustrate practical applications of the goal-oriented model reduction, we first systematically applied model-checking for the goal reachability on the initial and reduced model (table 1).

We compared two different softwares: `NuSMV` [8] which combines Binary Decision Diagrams and SAT approaches for synchronous systems, and `its-reach` [23] which implements efficient decision diagram data structures [18]. In both cases, the transition systems specified as input of these tools is an exact encoding of the asynchronous semantics of the automata networks, where steps (definition 2) are always composed of only one transition. For `NuSMV`, the reachability property is specified with CTL [9] ("$\texttt{EF}\ g_\top$", $g_\top$ being the goal local state, and `EF` the *exists eventually* CTL operator). It is worth noting that `NuSMV` implements the *cone of influence* reduction [5] which removes variables not involved in the property. `its-reach` is optimized for checking if a state belongs to the reachable state space, and cannot perform CTL checking.

Experiments show a remarkable gain in tractability for the model-checking of reduced networks. For large cases, we observe that the dynamics can be tractable only after model reduction (e.g., TCell-r (94), RBE2F (370)). `its-reach` is significantly more efficient than `NuSMV` because it is tailored for simple reachability checking, whereas `NuSMV` handles much more general properties.

Because the goal-reduction preserves all the minimal traces for the goal reachability, it preserves the goal reachability: the results of the model-checking is equivalent in the initial and reduced model.

4.3 Example of application: cut set verification

The above application to simple reachability does not requires the preservation of *all* the minimal traces. Here, we apply the goal-oriented reduction to the cut sets for reachability, where the *completeness of minimal traces is crucial*.

Given a goal, a *cut set* is a set of local states such that any trace leading to the goal involves, in some of its transitions, one of these local states. Therefore, disabling all the local states of a cut set should make the reachability of the goal impossible. This disabling could be implemented by the knock-out/in of the corresponding species in the biological system: cut sets predict mutations which should prevent a concerned reachability to occur (e.g., active transcription factor). Such cut sets have been studied in [32, 28] and are close to intervention sets [21] (although those latter are not defined on traces but on pseudo-steady states).

We focus here on verifying if a (predicted) set of local states is, indeed, a cut set for the goal reachability. In the scope of this experiment, we consider cut sets that are disjoint with the initial state. The cut set property can be expressed with CTL: $\{a_1, b_1\}$ is a cut set for $g_\top$ reachability if the model satisfies the CTL

[1] Scripts and models available at http://loicpauleve.name/gored-suppl.zip

				Verification of goal reachability	
Model	$\lvert T \rvert$	# states	$\lvert$unf$\rvert$	NuSMV	its-reach
EGF-r (20)	68	4,200	1,749	0.2s 10Mb	0.17 7Mb
	43	**722**	**336**	**0.1 8Mb**	**0.1s 5Mb**
Wnt (32)	197	7,260,160	KO	30s 48Mb	0.3s 18Mb
	117	**241,060**	**217,850**	**0.9s 32Mb**	**0.5s 17Mb**
TCell-r (40)	90	$\approx 1.2 \cdot 10^{11}$	KO	KO	1.1s 52Mb
	46	**25,092**	**14,071**	**3.8s 36Mb**	**0.6s 15Mb**
MAPK (53) profile 1	173	$\approx 3.8 \cdot 10^{12}$	KO	KO	0.9s 60Mb
	113	$\approx \mathbf{4.5 \cdot 10^{10}}$	**KO**	**KO**	**2s 48Mb**
MAPK (53) profile 2	173	8,126,465	KO	63s 83Mb	0.2s 15Mb
	69	**269,825**	**155,327**	**1.5s 36Mb**	**0.4s 18Mb**
VPC (88)	332	KO	KO	KO	1s 50Mb
	219	$\mathbf{1.8 \cdot 10^{9}}$	**43,302**	**236s 156Mb**	**0.8s 21Mb**
TCell-r (94)	217	KO	KO	KO	KO
	42	**54.921**	**1,017**	**0.4 23Mb**	**0.26s 14Mb**
TCell-d (101) profile 1	384	$\approx 2.7 \cdot 10^{8}$	257	3s 40Mb	0.5s 24Mb
	0	**1**	**1**		
TCell-d (101) profile 2	384	KO	KO	KO	0.5s 23Mb
	161	**75,947,684**	**KO**	**474s 260Mb**	**0.3s 19Mb**
EGF-r (104) profile 1	378	9,437,184	47,425	7s 35Mb	0.6s 23Mb
	0	**1**	**1**		
EGF-r (104) profile 2	378	$\approx 2.7 \cdot 10^{16}$	KO	KO	1.36s 60Mb
	69	**62,914,560**	**KO**	**11s 33Mb**	**0.3s 17Mb**
RBE2F (370)	742	KO	KO	KO	KO
	56	**2,350,494**	**28,856**	**5s 377Mb**	**5s 170Mb**

Table 1: Comparisons before (normal font) and **after** (bold font) the goal-oriented AN reduction. Each model is identified by the system, the number of automata (within parentheses), and a profile specifying the initial state and the reachability goal. $|T|$ is the number of local transitions in the AN specification; "#states" is the number of reachable global states from the initial state; "|unf|" is the size of the complete finite prefix of the unfolding. "KO" indicates an execution running out of time (30 minutes) or memory. When applied to goal reachability, we show the total execution time and memory used by the tools NuSMV and its-reach. Computation times where obtained on an Intel® Core™ i7 3.4GHz CPU with 16GB RAM. *For each case, the reduction procedure took less than 0.1s.*

	Wnt (32)	TCell-r (40)	EGF-r (104)	TCell-d (101)	RBE2F (370)
NuSMV	44s 55Mb	KO	KO	KO	KO
	9.1s 27Mb	**2.4s 34Mb**	**13s 33Mb**	**600s 360Mb**	**6s 29Mb**
its-ctl	105s 2.1Gb	492s 10Gb	KO	KO	KO
	16s 720Mb	**11s 319Mb**	**21s 875Mb**	**KO**	**179s 1.8Gb**

Table 2: Comparisons before (normal font) and **after** (bold font) the goal-oriented AN reduction for CTL model-checking of cut sets.

property `not E [ (not` a_1 `and not` b_1`) U` $g_\top$ `]` (`U` being the *until* operator). The property states that there exists no trace where none of the local state of the cut set is reached prior to the goal. It is therefore required that *all* the minimal traces to the goal reachability are present in the model: if one is missing, a set of local states could be validated as cut set whereas it may not be involved in the missed trace.

Table 2 compares the model-checking of cut sets properties using `NuSMV` and `its-ctl` [23] on a range of the biological networks used in the previous sections. Because the dynamical property is much more complex, `its-reach` cannot be used. The cut sets have been computed beforehand with `Pint`. Because the goal-oriented reduction preserves all the minimal traces to the goal, the results are equivalent in the reduced models. Similarly to the simple reachability, the goal-oriented reduction drastically improves the tractability of large models.

5 Discussion

This paper introduces a new reduction for automata networks parametrized by a reachability property of the form: from a state s there exists a trace which leads to a state where a given automaton g is in state $g_\top$.

The goal-oriented reduction preserves *all* the minimal traces satisfying the reachability property under a general concurrent semantics which allows at each step simultaneous transitions of an arbitrary number of automata. Hence, the minimal trace preservation is ensured for any stricter semantics, ranging from the fully asynchronous to maximally parallel transitions.

Its time complexity is polynomial in the total number of transitions and exponential with the maximal number of local states within an automaton. Therefore, the procedure is extremely scalable when applied on networks between numerous automata, but where each automaton has a few local states.

Applied to logical models of biological networks, the goal-oriented reduction can lead to a drastic shrinkage of the reachable state space with a negligible computational cost. We illustrated its application for the model-checking of simple reachability properties, but also for the validation of cut sets, which requires the completeness of minimal traces in the reduced model. It results that the goal-oriented reduction can increase considerably the scalability of the formal analysis of dynamics of automata networks.

The goal is expressed as a single local state reachability, which also allows to to support sequential reachability properties between (sub)states using an extra automaton. For instance, the property "reach a_1 and b_1, then reach c_1" can be encoded using one extra automaton g, where $g_0 \xrightarrow{\{a_1,b_1\}} g_1$ and $g_1 \xrightarrow{\{c_1\}} g_\top$.

Further work consider performing the reduction on the fly, during the state space exploration, expecting a stronger pruning. Although the complexity of the reduction is low, such approaches would benefit from heuristics to indicate when a new reduction step may be worth to apply.

References

[1] Wassim Abou-Jaoudé, Pedro T. Monteiro, Aurélien Naldi, Maximilien Grandclaudon, Vassili Soumelis, Claudine Chaouiya, and Denis Thieffry. Model checking to assess t-helper cell plasticity. *Frontiers in Bioengineering and Biotechnology*, 2, Jan 2015. doi:10.3389/fbioe.2014.00086.

[2] Julio Aracena, Eric Goles, Andrés Moreira, and Lilian Salinas. On the robustness of update schedules in Boolean networks. *Biosystems*, 97(1):1 – 8, 2009. doi:10.1016/j.biosystems.2009.03.006.

[3] Luca Bernardinello and Fiorella De Cindio. A survey of basic net models and modular net classes. In Grzegorz Rozenberg, editor, *Advances in Petri Nets 1992*, volume 609 of *Lecture Notes in Computer Science*, pages 304–351. Springer Berlin / Heidelberg, 1992. doi:10.1007/3-540-55610-9_177.

[4] Gérard Berthelot. Checking properties of nets using transformations. In G. Rozenberg, editor, *Advances in Petri Nets 1985*, volume 222 of *Lecture Notes in Computer Science*, pages 19–40. Springer Berlin Heidelberg, 1986. doi:10.1007/BFb0016204.

[5] Armin Biere, Edmund Clarke, Richard Raimi, and Yunshan Zhu. Verifying safety properties of a powerpc microprocessor using symbolic model checking without bdds. In *In Proc. 11 th Int. Conf. on Computer Aided Verification*, pages 60–71. Springer-Verlag, 1999. doi:10.1007/3-540-48683-6_8.

[6] Thomas Chatain, Stefan Haar, Loïg Jezequel, Loïc Paulevé, and Stefan Schwoon. Characterization of reachable attractors using Petri net unfoldings. In Pedro Mendes, Joseph Dada, and Kieran Smallbone, editors, *Computational Methods in Systems Biology*, volume 8859 of *Lecture Notes in Computer Science*, pages 129–142. Springer Berlin Heidelberg, 2014. doi:10.1007/978-3-319-12982-2_10.

[7] Allan Cheng, Javier Esparza, and Jens Palsberg. Complexity results for 1-safe nets. *Theor. Comput. Sci.*, 147(1&2):117–136, 1995. URL: http://dx.doi.org/10.1016/0304-3975(94)00231-7, doi:10.1016/0304-3975(94)00231-7.

[8] Alessandro Cimatti, Edmund Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. NuSMV 2: An opensource tool for symbolic model checking. In *Computer Aided Verification*, volume 2404 of *Lecture Notes in Computer Science*, pages 241–268. Springer Berlin / Heidelberg, 2002. doi:10.1007/3-540-45657-0_29.

[9] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs*, pages 52–71, London, UK, 1981. Springer-Verlag.

[10] David P. A. Cohen, Loredana Martignetti, Sylvie Robine, Emmanuel Barillot, Andrei Zinovyev, and Laurence Calzone. Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLoS Comput Biol*, 11(11):e1004571, Nov 2015. doi:10.1371/journal.pcbi.1004571.

[11] Institut Curie/Sysbio. RB/E2F pathway. http://bioinfo-out.curie.fr/projects/rbpathway/.

[12] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, Jonathan Hayman, Jean Krivine, Christopher D. Thompson-Walsh, and Glynn Winskel. Graphs, rewriting and pathway reconstruction for rule-based models. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012*, volume 18 of *LIPIcs*, pages 276–288. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. doi:10.4230/LIPIcs.FSTTCS.2012.276.

[13] Javier Esparza and Keijo Heljanko. *Unfoldings: A Partial-Order Approach to Model Checking (Monographs in Theoretical Computer Science. An EATCS Series)*. Springer Publishing Company, Incorporated, 1 edition, 2008.

[14] Jerome Feret, Heinz Koeppl, and Tatjana Petrov. Stochastic fragments: A framework for the exact reduction of the stochastic semantics of rule-based models. *International Journal of Software and Informatics*, 7(4):527 – 604, 2013.

[15] Luca Grieco, Laurence Calzone, Isabelle Bernard-Pierrot, Franois Radvanyi, Brigitte Kahn-Perlès, and Denis Thieffry. Integrative modelling of the influence of MAPK network on cancer cell fate decision. *PLoS Comput Biol*, 9(10):e1003286, oct 2013. doi:10.1371/journal.pcbi.1003286.

[16] Colomoto group. Logicalmodel. https://github.com/colomoto/logicalmodel.

[17] Serge Haddad and Jean-François Pradat-Peyre. New efficient Petri nets reductions for parallel programs verification. *Parallel Processing Letters*, 16(1):101–116, March 2006. doi:10.1142/S0129626406002502.

[18] Alexandre Hamez, Yann Thierry-Mieg, and Fabrice Kordon. Building efficient model checkers using hierarchical set decision diagrams and automatic saturation. *Fundam. Inf.*, 94(3-4):413–437, 2009. doi:10.3233/FI-2009-137.

[19] Ryszard Janicki, Jetty Kleijn, Maciej Koutny, and Łukasz Mikulski. Step traces. *Acta Informatica*, Jun 2015. doi:10.1007/s00236-015-0244-z.

[20] Ryszard Janicki, Peter E. Lauer, Maciej Koutny, and Raymond Devillers. Concurrent and maximally concurrent evolution of nonsequential systems. *Theoretical Computer Science*, 43(0):213 – 238, 1986. doi:10.1016/0304-3975(86)90177-5.

[21] Steffen Klamt, Julio Saez-Rodriguez, Jonathan Lindquist, Luca Simeoni, and Ernst Gilles. A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinformatics*, 7(1):56, 2006. doi:10.1186/1471-2105-7-56.

[22] Robert P Kurshan. *Computer-aided verification of coordinating processes: the automata-theoretic approach*. Princeton university press, 1994.

[23] LIP6/Move. Its tools. http://ddd.lip6.fr/itstools.php.

[24] Claire Loiseaux, Susanne Graf, Joseph Sifakis, Ahmed Bouajjani, Saddek Bensalem, and David Probst. Property preserving abstractions for the verification of concurrent systems. *Formal methods in system design*, 6(1):11–44, 1995. doi:10.1007/BF01384313.

[25] Guillaume Madelaine, Cédric Lhoussaine, and Joachim Niehren. Attractor Equivalence: An Observational Semantics for Reaction Networks. In *First International Conference on Formal Methods in Macro-Biology*, volume 8738 of *Lecture Notes in Computer Science*, pages 82–101. Springer-Verlag, 2014. doi:10.1007/978-3-319-10398-3_7.

[26] Aurélien Naldi, Elisabeth Remy, Denis Thieffry, and Claudine Chaouiya. Dynamically consistent reduction of logical regulatory graphs. *Theoretical Computer Science*, 412(21):2207 – 2218, 2011. doi:10.1016/j.tcs.2010.10.021.

[27] Loïc Paulevé. PINT - Static analyzer for dynamics of automata networks, http://loicpauleve.name/pint.

[28] Loïc Paulevé, Geoffroy Andrieux, and Heinz Koeppl. Under-approximating cut sets for reachability in large scale automata networks. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification*, volume 8044 of *Lecture Notes in Computer Science*, pages 69–84. Springer Berlin Heidelberg, 2013. doi:10.1007/978-3-642-39799-8_4.

[29] 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.

[30] Lutz Priese and Harro Wimmel. A uniform approach to true-concurrency and interleaving semantics for petri nets. *Theoretical Computer Science*, 206(1–2):219 – 256, 1998. doi:10.1016/S0304-3975(97)00169-2.

[31] Julio Saez-Rodriguez, Luca Simeoni, Jonathan A Lindquist, Rebecca Hemenway, Ursula Bommhardt, Boerge Arndt, Utz-Uwe Haus, Robert Weismantel, Ernst D Gilles, Steffen Klamt, and Burkhart Schraven. A logical model provides insights into t cell receptor signaling. *PLoS Comput Biol*, 3(8):e163, 08 2007. doi:10.1371/journal.pcbi.0030163.

[32] Ozgur Sahin, Holger Frohlich, Christian Lobke, Ulrike Korf, Sara Burmester, Meher Majety, Jens Mattern, Ingo Schupp, Claudine Chaouiya, Denis Thieffry, Annemarie Poustka, Stefan Wiemann, Tim Beissbarth, and Dorit Arlt. Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance. *BMC Systems Biology*, 3(1), 2009. doi:10.1186/1752-0509-3-1.

[33] Regina Samaga, Julio Saez-Rodriguez, Leonidas G. Alexopoulos, Peter K. Sorger, and Steffen Klamt. The logic of egfr/erbb signaling: Theoretical properties and analysis of high-throughput data. *PLoS Comput Biol*, 5(8):e1000438, 08 2009. doi:10.1371/journal.pcbi.1000438.

[34] Philippe Schnoebelen and Natalia Sidorova. Bisimulation and the reduction of Petri nets. In *Application and Theory of Petri Nets 2000*, volume 1825 of *Lecture Notes in Computer Science*, pages 409–423. Springer Berlin Heidelberg, 2000. doi:10.1007/3-540-44988-4_23.

[35] S. Schwoon. Mole. http://www.lsv.ens-cachan.fr/~schwoon/tools/mole/.

[36] Joseph Sifakis. Property preserving homomorphisms of transition systems. In Edmund Clarke and Dexter Kozen, editors, *Logics of Programs*, volume 164 of *Lecture Notes in Computer Science*, pages 458–473. Springer Berlin Heidelberg, 1984. doi:10.1007/3-540-12896-4_381.

[37] N. Weinstein and L. Mendoza. A network model for the specification of vulval precursor cells and cell fusion control in caenorhabditis elegans. *Frontiers in Genetics*, 4(112), 2013. doi:10.3389/fgene.2013.00112.

A Proof of minimal traces preservation

We assume a global AN (Σ, S, T) where $g \in \Sigma$, $g_\top \in S(g)$, and $s \in S$ with $s(g) \neq g_\top$.

From property 1 and definition 7, any trace reaching first a_i and then a_j uses all the transitions of at least one local path in $\text{local-paths}_s(a_i \rightsquigarrow a_j)$.

We first prove with lemma 2 that the last transition of a minimal trace π for $g_\top$ reachability, of the form $\pi^{|\pi|} = \{g_i \to g_\top\}$, is necessarily in $\text{tr}(\mathcal{B})$. Indeed, by definition of $\mathcal{B}$, $g_0 \rightsquigarrow g_\top \in \mathcal{B}$; and by lemma 1, $g_i \to g_\top \notin \text{local-paths}_s(g_0 \rightsquigarrow g_\top)$ implies that reaching g_i requires to reach $g_\top$ beforehand.

Lemma 1. *Given $a_j \to a_i \in T$, if $a_j \to a_i \notin \text{tr}(\text{local-paths}_s(a_0 \rightsquigarrow a_i))$, then for any trace π from s with $a_j \in \pi^{v\bullet}$ and $a_i \in \pi^{w\bullet}$ for some $v, w \in [1; |\pi|]$, there exists $u < v$ with $a_i \in \pi^{u\bullet}$.*

Proof. Let $\eta \in \text{local-paths}_s(a_0 \rightsquigarrow a_j)$ be an acyclic local path such that $\forall n \in [1; |\eta|]$, $a_i \neq \text{dest}(\eta^n)$. The sequence $\eta :: a_j \to a_i$ is then acyclic and, by definition, belongs to $\text{local-paths}_s(a_0 \rightsquigarrow a_i)$, which is a contradiction. $\square$ $\square$

Lemma 2. *If π is a minimal trace for $g_\top$ reachability from state s, then, necessarily, $\pi^{|\pi|} \subseteq \text{tr}(\mathcal{B})$.*

Proof. As π is minimal for $g_\top$ reachability, without loss of generality, we can assume that $\pi^{|\pi|} = \{g_i \to g_\top\}$. By definition, $\text{tr}(\text{local-paths}_s(g_0 \rightsquigarrow g_\top)) \subseteq \text{tr}(\mathcal{B})$. By lemma 1, if $g_i \to g_\top \notin \text{tr}(\text{local-paths}_s(g_0 \rightsquigarrow g_\top))$, then there exists $u < |\pi|$ such that $g_\top \in \pi^{u\bullet}$; hence, π would be non minimal. $\square$ $\square$

The rest of the proof of theorem 1 is derived by contradiction: if a transition of π is not in $\text{tr}(\mathcal{B})$, we can build a sub-trace of π which preserves $g_\top$ reachability, therefore π is not minimal.

Given a transition $a_i \to a_j$ in the q-th step of π that is not in $\text{tr}(\mathcal{B})$, removing $a_i \to a_j$ from π^q would imply to remove any further transition that depend causally on it. Two cases arise from this fact: either all further transitions that depend on a_j must be removed; or $a_i \to a_j$ is part of loop within automaton a, and it is sufficient to remove the loop from π.

Lemma 3 ensures that if $a_z \rightsquigarrow a_k$ is in $\mathcal{B}$ and if a_z occurs before the q-th step and a_k after the q-th step of π, then $a_i \to a_j \notin \text{tr}(\text{local-paths}_s(a_z \rightsquigarrow a_k))$ only if $a_i \to a_j$ is part of a loop, i.e., there are two steps surrounding q where the automaton a is in the same state before their application.

Lemma 3. *Given $a \in \Sigma$ and $u, q, v \in [1; |\pi|]$, $u \leq q < v$, with $a_z \in {}^\bullet\pi^u$, $a_k \in {}^\bullet\pi^v \cup \pi^{v\bullet}$, and $a_i \to a_j \in \pi^q \setminus \text{tr}(\mathcal{B})$, if $a_z \rightsquigarrow a_k \in \mathcal{B}$ then $\exists m, n \in [u; v]$, $m \leq q \leq n$ such that $(\pi^{1..m-1})^\bullet \cap S(a) = (\pi^{1..n})^\bullet \cap S(a)$; and $a_k \in {}^\bullet\pi^v \Rightarrow n < v$.*

Proof. If $a_i \to a_j \notin \text{tr}(\mathcal{B})$ and $a_z \rightsquigarrow a_k \in \text{tr}(\mathcal{B})$, necessarily $a_i \to a_j \notin \text{tr}(\text{local-paths}_s(a_z \rightsquigarrow a_k))$. Therefore $a_i \to a_j$ belongs to a loop of a local path from a_z (at index u in π) to a_k (at index v in π). Hence, $\exists m, n \in [u; v]$ with $m \leq q \leq n$ and $a_h, a_x, a_y \in S(a)$ such that $a_h \to a_x \in \pi^m$ and $a_y \to a_h \in \pi^n$; therefore $(\pi^{1..m-1})^\bullet \cap S(a) = (\pi^{1..n})^\bullet \cap S(a) = a_h$. In the case where $a_k \in {}^\bullet\pi^v$, $a_k \neq a_h$, hence $n < v$. $\square$ $\square$

Intuitively, lemma 3 imposes that π has the following form:

$$\pi = \cdots \; :: \underset{u}{\overset{a_z \in}{\pi^u}} :: \; \cdots \; :: \underset{m}{a_h \to a_x} :: \; \cdots \; :: \underset{q}{\overset{\notin \text{tr}(\mathcal{B})}{\mathbf{a_i} \to \mathbf{a_j}}} :: \; \cdots \; :: \underset{n}{a_y \to a_h} :: \; \cdots \; :: \underset{v}{\overset{a_k \in}{\pi^v}} :: \; \cdots$$

given that $a_z \rightsquigarrow a_k \in \mathcal{B}$.

The idea is then to remove the transitions forming the loop within automaton a. However, transitions in other automata may depend causally on the transitions that compose the local loop in automaton a within steps m and n, following the notations in lemma 3.

Lemma 4 establishes that we can always find m and n such that none of the transitions within these steps with an enabling condition depending on automaton a are in $\mathrm{tr}(\mathcal{B})$. Indeed, if a transition in $\mathrm{tr}(\mathcal{B})$ depends on a local state of a, let us call it a_p, the objectives $a_0 \rightsquigarrow a_p$ and $a_p \rightsquigarrow a_k$ are in $\mathcal{B}$, due to the second and third condition in definition 8. Lemma 3 can then be applied on the subpart of π that contains the transition $a_i \to a_j$ not in $\mathrm{tr}(\mathcal{B})$ and that concretizes either $a_0 \rightsquigarrow a_p$ or $a_p \rightsquigarrow a_k$ to identify a smaller loop containing $a_i \to a_j$.

Lemma 4. *Let us assume $a \in \Sigma$ and $q \in [1; |\pi|]$ with $a_i \to a_j \in \pi^q \setminus \mathrm{tr}(\mathcal{B})$. There exists $m, n \in [1; |\pi|]$ with $m \leq q \leq n$ such that $\forall t \in \mathrm{tr}(\pi^{m+1..n})$, $\mathrm{enab}(t) \cap S(a) \neq \emptyset \Rightarrow t \notin \mathrm{tr}(\mathcal{B})$, and, if $a = g$ or $\exists t \in \mathrm{tr}(\pi^{n+1..|\pi|}) \cap \mathrm{tr}(\mathcal{B})$ with $\mathrm{enab}(t) \cap S(a) \neq \emptyset$, then $(\pi^{1..m-1})^{\bullet} \cap S(a) = (\pi^{1..n})^{\bullet} \cap S(a)$.*

Proof. First, let us assume that $a \neq g$ and for any $t \in \pi^{q+1..|\pi|}$, $\mathrm{enab}(t) \cap S(a) \neq \emptyset \Rightarrow t \notin \mathrm{tr}(\mathcal{B})$: the lemma is verified with $m = q$ and $n = |\pi|$.

Then, let us assume there exists $v \in [q + 1; |\pi|]$ such that $\exists t \in \mathrm{tr}(\pi^v) \cap \mathrm{tr}(\mathcal{B})$ with $a_k \in \mathrm{enab}(t)$. By definition 8, this implies $a_0 \rightsquigarrow a_k \in \mathcal{B}$. By lemma 3, there exists $m, n \in [1; v-1]$ with $m \leq q \leq n$ such that $(\pi^{1..m-1})^{\bullet} \cap S(a) = (\pi^{1..n})^{\bullet} \cap S(a)$.

Otherwise, $a = g$, and by lemma 3 with $a_k = g_\top$, there exists $m, n \in [1; |\pi|]$ with $m \leq q \leq n$ and $m \neq n$ such that $(\pi^{1..m-1})^{\bullet} \cap S(a) = (\pi^{1..n})^{\bullet} \cap S(a)$. Remark that it is necessary that $n < |\pi|$: if $n = |\pi|$, $g_\top \in (\pi^{1..m-1})^{\bullet}$, so π would be not minimal.

In both cases, if there exists $r \in [m + 1; n]$ such that $\exists a_p \in S(a)$ and $\exists t \in \pi^r$ with $a_p \in \mathrm{enab}(t)$, then $t \in \mathrm{tr}(\mathcal{B})$ implies that $a_0 \rightsquigarrow a_p \in \mathcal{B}$ and $a_p \rightsquigarrow a_k \in \mathcal{B}$ (definition 8). If $r > q$, by lemma 3 with $a_k = a_p$ and $v = r$, there exists $m', n' \in [m+1; n]$ such that $m' \leq q \leq n' < r \leq n$ with $(\pi^{1..m'-1})^{\bullet} \cap S(a) = (\pi^{1..n'})^{\bullet} \cap S(a)$. If $r \leq q$, by lemma 3 with $a_0 = a_p$ and $u = r$, there exists $m', n' \in [m + 1; n]$ such that $r \leq m' \leq q \leq n'$ with $(\pi^{1..m'-1})^{\bullet} \cap S(a) = (\pi^{1..n'})^{\bullet} \cap S(a)$. Therefore, by induction with lemma 3, there exists $m, n \in [1; |\pi|]$ such that $\forall t \in \mathrm{tr}(\pi^{m+1..n})$, $\mathrm{enab}(t) \cap S(a) \neq \emptyset \Rightarrow t \notin \mathrm{tr}(\mathcal{B})$. ☐ ☐

Using lemma 4, we show how we can identify a subset of transitions in π that can be removed to obtain a sub-trace for $g_\top$ reachability. In the following, we refer to the couple (m, n) of lemma 4 with $\mathrm{cb}(\pi, a, q)$ (definition 9).

Definition 9 ($\mathrm{cb}(\pi, a, q)$)**.** Given $a \in \Sigma$, $q \in [1; |\pi|]$ with $t \in \pi^q \setminus \mathrm{tr}(\mathcal{B})$ and $\Sigma(t) = a$, we define $\mathrm{cb}(\pi, a, q) = (m, n)$ where $m, n \in [1; |\pi|]$ such that:

- $\forall t \in \mathrm{tr}(\pi^{m+1..n})$, $\mathrm{enab}(t) \cap S(a) \neq \emptyset \Rightarrow t \notin \mathrm{tr}(\mathcal{B})$;
- $a = g \vee \exists t \in \mathrm{tr}(\pi^{n+1..|\pi|}) \cap \mathrm{tr}(\mathcal{B})$ with $\mathrm{enab}(t) \cap S(a) \neq \emptyset \implies (\pi^{1..m-1})^{\bullet} \cap S(a) = (\pi^{1..n})^{\bullet} \cap S(a)$. Moreover, if $a = g$, then $n < |\pi|$.

We use lemma 4 to collect the portions of π to redact according to each automaton. We start from the last transition in π that is not in $\mathrm{tr}(\mathcal{B})$: if $\mathrm{tr}(\pi) \not\subseteq \mathrm{tr}(\mathcal{B})$, there exists $l \in [1; |\pi|]$ such that $\pi^l \not\subseteq \mathrm{tr}(\mathcal{B})$ and $\forall n > l, \pi^n \subseteq \mathrm{tr}(\mathcal{B})$. By lemma 2, we know that $l < |\pi|$. Let us denote by $b_i \to b_j$ one of the transitions in π^l which is not in $\mathrm{tr}(\mathcal{B})$.

We define $\Psi \subseteq \Sigma \times [1; |\pi|] \times [1; |\pi|]$ the smallest set which satisfies:

- $(b, m, n) \in \Psi$ if $\mathrm{cb}(\pi, l, b) = (m, n)$
- $\forall (a, m, n) \in \Psi$, $\forall q \in [m + 1; n]$, $\forall t \in \pi^q$, $\mathrm{enab}(t) \cap S(a) \neq \emptyset \implies (\Sigma(t), m', n') \in \Psi$ where $\mathrm{cb}(\pi, q, \Sigma(t)) = (m', n')$.

Finally, let us define the sequence of steps ϖ as the sequence of steps π where the transitions delimited by Ψ are removed: for each $(a, m, n) \in \Psi$, all the transitions of automaton a occurring between π^m and π^n are removed. Formally, $|\varpi| = |\pi|$ and for all $q \in [1; |\pi|]$, $\varpi^q \triangleq \{t \in \pi^q \mid \nexists (a, m, n) \in \Psi : a = \Sigma(t) \wedge m \leq q \leq n\}$.

From lemma 4 and Ψ definition, ϖ is a valid trace. Moreover, by lemma 4, there is no $q \in [1; |\pi|]$ such that $(g, q, |\pi|) \in \Psi$, hence $g_\top \in \varpi^{\bullet}$. Therefore, π is not minimal, which contradicts our hypothesis. ☐

Example 2. Let us consider the reachability of c_2 in the AN of figure 1 from state $\langle a_0, b_0, c_0, d_0\rangle$. The transitions $\mathrm{tr}(\mathcal{B})$ preserved by the reduction for that goal are listed in figure 2.

Let π be the following trace in the AN of figure 1:

$$\pi = \{a_0 \xrightarrow{\{b_0\}} a_1\} :: \{b_0 \xrightarrow{\{a_1\}} b_1, c_0 \xrightarrow{\{a_1\}} c_1\} :: \{a_1 \xrightarrow{\emptyset} a_0\} :: \{b_1 \xrightarrow{\{a_0\}} b_0\} \\ :: \{c_1 \xrightarrow{\{b_0\}} c_2\} \ .$$

The latest transition not in $\mathrm{tr}(\mathcal{B})$ is $b_1 \xrightarrow{\{a_0\}} b_0$ at step 4. One can compute $\mathrm{cb}(\pi, 4, b) = (2, 4)$, and as there is no transition involving b between steps 3 and 4, $\Psi = \{(b, 2, 4)\}$; therefore, the sequence

$$\varpi = \{a_0 \xrightarrow{\{b_0\}} a_1\} :: \{c_0 \xrightarrow{\{a_1\}} c_1\} :: \{a_1 \xrightarrow{\emptyset} a_0\} :: \{\} :: \{c_1 \xrightarrow{\{b_0\}} c_2\}$$

is a valid sub-trace of π reaching c_2, proving π non-minimality.

In conclusion, if π is a minimal trace for $g_\top$ reachability from state s, then, $\mathrm{tr}(\pi) \subseteq \mathrm{tr}(\mathcal{B})$.

B Experiments with partial reduction

The goal-oriented reduction relies on two intertwined analyses of the local causality in ANs: (1) the computation of potentially involved objectives (section 3.2) and (2) the filtering of objective that can be proven impossible (section 3.1). The second part can be considered optional: one could simply define the predicate $\mathbf{valid}_s$ to be always true. In order to appreciate the effect of this second part, we show here the intermediary results of model reduction without the filtering of impossible objectives. It is shown in table below, in the lines in *italic*. As we can see, for some models it has no effect on the reduction, for some others the filtering parts is necessary to obtained important reduction of the state space (e.g., MAPK, TCell-r (94), TCell-d).

Model	# tr	# states	|unf|
EGF-r (20)	68	4,200	1,749
	43	*722*	*336*
	43	**722**	**336**
Wnt (32)	197	7,260,160	KO
	134	*241,060*	*217,850*
	117	**241,060**	**217,850**
TCell-r (40)	90	$\approx 1.2 \cdot 10^{11}$	KO
	46	*25,092*	*14,071*
	46	**25,092**	**14,071**
MAPK (53) profile 1	173	$\approx 3.8 \cdot 10^{12}$	KO
	147	$\approx 9 \cdot 10^{10}$	*KO*
	113	$\approx \mathbf{4.5 \cdot 10^{10}}$	**KO**
MAPK (53) profile 2	173	8,126,465	KO
	148	*1,523,713*	*KO*
	69	**269,825**	**155,327**
VPC (88)	332	KO	KO
	278	$\approx 2.9 \cdot 10^{12}$	*185,006*
	219	$\mathbf{1.8 \cdot 10^{9}}$	**43,302**
TCell-r (94)	217	KO	KO
	112	*KO*	*KO*
	42	**54.921**	**1,017**
TCell-d (101) profile 1	384	$\approx 2.7 \cdot 10^{8}$	257
	275	$\approx 1.1 \cdot 10^{8}$	*159*
	0	**1**	**1**

TCell-d (101) profile 2	384	KO	KO
	253	$\approx 2.4 \cdot 10^{12}$	*KO*
	161	**75,947,684**	**KO**
EGF-r (104) profile 1	378	9,437,184	47,425
	120	*12,288*	*1,711*
	0	**1**	**1**
EGF-r (104) profile 2	378	$\approx 2.7 \cdot 10^{16}$	KO
	124	$\approx 2 \cdot 10^{9}$	*KO*
	69	**62,914,560**	**KO**
RBE2F (370)	742	KO	KO
	56	*2,350,494*	*28,856*
	56	**2,350,494**	**28,856**