Diagnosability Of Asynchronous Discrete Event Systems in Partial Order Semantics

Stefan Haar

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

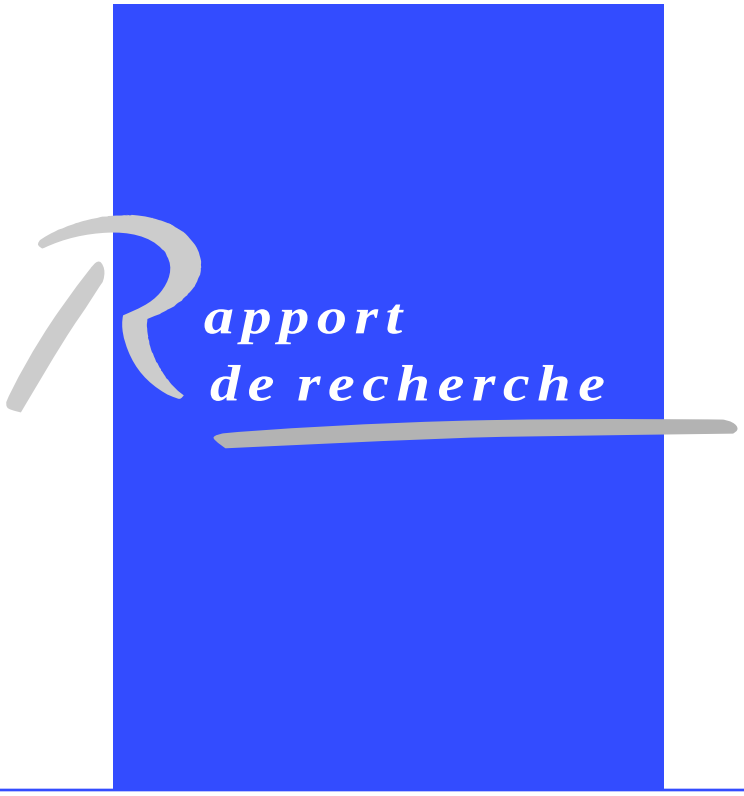
Diagnosability Of Asynchronous Discrete Event Systems in Partial Order Semantics

Stefan Haar

N°5248

Juillet 2004

Systèmes communicants



ISSN 0249-6399 ISRN INRIA/RR--5248--FR+ENG

Diagnosability Of Asynchronous Discrete Event Systems in Partial Order Semantics

Stefan Haar*

Systèmes communicants
Projets DistribCom

Rapport de recherche n ° 5248 — Juillet 2004 — 25 pages

Abstract: In truly asynchronous, distributed systems, neither global state nor global time are available. Automata-based diagnosis therefore reaches its limitations there; a different approach, based on Petri net unfoldings, was proposed in [9]. It is motivated by the problem of event correlation in telecommunications network management, and uses only local states, in combination with a partial order model of time. Diagnosis is performed by correlation of the observed partial order alarm patterns and partial order executions of the system model. As in the classical automata setting, the presence of invisible transitions raises the problems of observability and diagnosability of a given system. In this paper, we give a definition of weak and strong *observability* and *diagnosability* in terms of partially ordered executions, and characterize diagnosable systems; the characterizing property can be effectively verified using a finite complete prefix of the net unfolding.

Key-words: asynchronous diagnosability, Petri nets, unfoldings, alarm correlation.

(Résumé : tsvp)

* Supported by the MAGDA2 project, RNRT; see the sites `http://www.telecom.gouv.fr/rnrt/index_net.htm` and `http://www.magda.elibel.tm.fr` for more information.

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : 02 99 84 71 00 - International : +33 2 99 84 71 00
Télécopie : 02 99 84 71 71 - International : +33 2 99 84 71 71

Diagnosticabilité des Systèmes à Evénements Discrets en Sémantique d'Ordre Partiel

Résumé :

Mots-clé : Diagnosticabilité asynchrone, Réseaux de Petri, dépliages, corrélation d'alarmes.

Contents

1 Introduction

The Diagnosis problem . Failure diagnosis for discrete event systems is a crucial task in automatic control. The *diagnosis problem* has received much attention in the literature, see [3, 31, 32]: it consists, in abstract terms, in determining possible behaviors in a partially observable system that are compatible with the observed pattern of alarms, i.e. that explainthe observable behavior. In the discrete event approach, system behavior is modeled as a regular language, and the system itself as well as the diagnoser are modeled as finite state machines (FSM); and they synchronize on *observable events*. Then, faulty behavior is determined by reading the state of the diagnoser.

Diagnosability The key property that has to be verified by the setup (that is, the subset of observable letters and the FSMs involved) for this to work, is *diagnosability*. It stipulates existence of some constant maximal length n such that, whatever the circumstances, if a fault occurs now, the longest sequence leading to a diagnoser state that indicates that fault, takes at most n steps from now. For the more formal definition, following Sampath et al. [30], let $\mathcal{L}$ be a prefix-closed language (the behavior of the system to be diagnosed) over the event alphabet $\mathfrak{A}$, denote $O \subseteq \mathfrak{A}$ the set of *observable* and $UO \triangleq \mathfrak{A} \backslash O$ that of *unobservable* events. Denote $P : \mathfrak{A}^* \to O^*$ the projection to observable words, that is, the homomorphism that erases all unobservable events and leaves observable ones unchanged; moreover, let $\Phi \subseteq UO$ be the set of *faults*[1]. Then $\mathcal{L}$ is *diagnosable* iff there exists $n \in \mathbb{N}$

[1] for simplicity, we drop the generalization made in [30], with Φ further partitioned into *fault types*; the results given below extend to that case.

such that, for any word $\mathcal{L} \ni w = w'f$ with $f \in \Phi$, any $v \in \mathfrak{A}^*$ s. th. $wv \in \mathcal{L}$ and $|v| \geq n$ satisfies

$$x \in P^{-1}\left[P\left(wv\right)\right] \quad \Rightarrow \quad |x|_\Phi \geq 1. \tag{1}$$

Here, $|u|$ denotes total length, and $|u|_\Phi$ the number of fault events of word u. Condition (1) means that every behavior x that produces the same sequence of observable events as wv does, contains at least one fault event: all extensions of w of at least length n will make the fault apparent.

Asynchronous diagnosis The present paper is motivated by the diagnosis of truly asynchronous systems, in which the above definition is not adequate as it stands since it is based on a global state automaton model. To see the limitations of that model, consider networked systems, such as shown on the left hand side in Figure 1. There, the sensor system

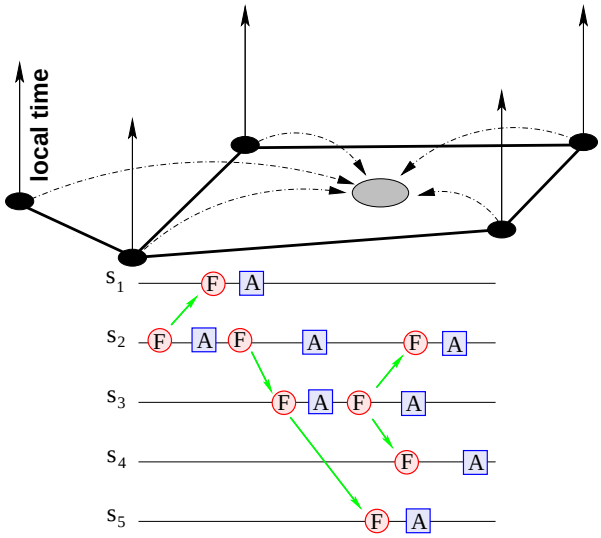


Figure 1: *Left : Supervision of networked systems; right: Alarms (squares) and faults (circles)*

is distributed: several local sensors are attached to some nodes of the network (shown in black). Each local sensor has only a partial view of the system, and its local time is not synchronized with that of other sensors. Alarms are reported asynchronously to the global supervisor, depicted in grey, which performs diagnosis; this is the typical architecture in telecommunications network management systems today (see [24]).

The need for a partial order semantics Even if the order of events may be correctly observed locally by each individual sensor, communicating alarm events via the network causes a loss of synchronization: as a result, the interleaving of events communicated to the supervisor is *nondeterministic*. The right picture of what the supervisor collects is therefore a *partially ordered* set, rather than a *sequence*, of alarms; see the pattern formed by the squares on the right hand side of Figure 1. The system itself being distributed, we also have partially ordered scenarios of *faults* (circles in Figure 1), as candidates for explaining the alarm pattern observed by the sensors. That is, we need to compute, for diagnosis, partially ordered sets of events that may have occurred and, had they occurred, would have produced the observed pattern of alarms. We emphasize that each such partial order represents in its turn an equivalence class of sequences, thus passing to partial orders allows, in general, a substantial gain in efficiency. In a second step (not treated here), the *likelihood* of scenarios has to be compared, to select, among the scenarios offered by the model, the most probable explanation for the given observation; see [1] for a construction of such likelihoods.

The present article recalls the formal framework for diagnosis for partially ordered runs given in [9], and establishes the adequate definition of *diagnosability*. The definition generalizes the classical one given, for the automaton case, in [30], to partial order behavior, where weak and strong diagnosability will have to be distinguished. We give a characterization of diagnosable systems, and show how observability and diagnosability can be checked on a finite construct (namely, a finite prefix of the system unfolding).

2 Petri Nets and Branching Processes

Petri nets are well-established formal models, whose adequacy for modelling distributed systems is widely recognized. A Petri net is a bipartite graph composed of *places* representing possible local states, and *transitions* that lead from an input set of places to an output set of places ; *tokens* indicate that a local condition holds. Petri nets - for the formal definition see below - can be seen as natural generalization of finite automata : indeed, every FA can be represented as a Petri net whose transitions have exactly one input and one output place, and with exactly one token on the initial state.

Accordingly, the semantics of PN allows a generalization over that of automata: Automaton behaviour can be expressed by *words*, which are linearly ordered and have a length given by counting the letters. Petri net behaviours, on the other hand, can be represented by *configurations*, that is, sets bearing a *partial order* that is not linear in general; instead of length, we will be manipulating *height* of such processes, a notion of which there exists a lower and an upper version since one can measure the progress of the *least advanced local process*, or the *most advanced one*.

Before coming back to these notions, some more background of the work presented here. This paper explores the asynchronous diagnosis approach from [9], using Petri nets (PNs) as its main tool ; compare [3, 4, 27] on the Theory and application of PNs. For the use of PNs in diagnosis, compare [19, 29, 16, 17]). *Branching processes* and *unfoldings* of PNs were originally proposed by Nielsen, Plotkin and Winskel [26], and used for *model checking*, see Mc

Millan [25] and Esparza et al. [6, 7]; they have been used for supervisory control in [22, 23]. Branching processes represent the set of executions of a Petri net in a net structure, using

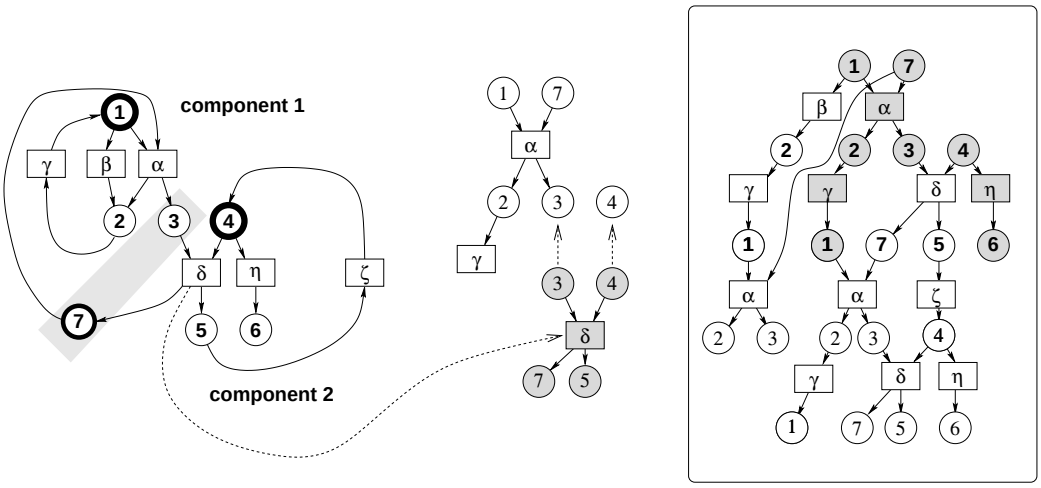


Figure 2: *Running example: $\mathcal{N}$ (left), a branching process for $\mathcal{N}$ (right), and a configuration of $\mathcal{N}$ being extended by a new event (center).*

an asynchronous semantics with local states and partially ordered time. Common prefixes of executions are shared, and executions differing only in some interleaving of independent transitions are represented only once; this meets the needs of asynchronous diagnosis, where some recorded alarm sequences differ only via the interleaving of concurrent alarms, hence it is desirable not to distinguish them, and similarly for the interleaving of concurrent faults.

Nets and homomorphisms We will now give some basic definitions to lay the grounds.

Definition 1. *[Nets and homomorphisms.] A* net *is a triple $N = (\mathcal{P}, \mathcal{T}, \rightarrow)$, where $\mathcal{P}$ and $\mathcal{T}$ are disjoint sets of* places *and* transitions*, and $\rightarrow \subset (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ is the* flow relation*. Denoting by $\circ$ the relational product, and by R^* the transitive closure of binary relation R, set $\leqslant \triangleq \rightarrow^*$ and $< \triangleq \rightarrow \circ \leqslant$. For a node $x \in \mathcal{P} \cup \mathcal{T}$, denote ${}^\bullet x \triangleq \{x' \mid x' \rightarrow x\}$ the* preset *and $x^\bullet \triangleq \{x' \mid x \rightarrow x'\}$ the* postset *of x. A* net homomorphism *from N to N' is a map $\phi : \mathcal{P} \cup \mathcal{T} \longmapsto \mathcal{P}' \cup \mathcal{T}'$ such that:*

1. *$\phi(\mathcal{P}) \subseteq \mathcal{P}'$, $\phi(\mathcal{T}) \subseteq \mathcal{T}'$, and*
2. *for every node x of N, $\phi_{|{}^\bullet x} : {}^\bullet x \rightarrow {}^\bullet\phi(x)$ and $\phi_{|x^\bullet} : x^\bullet \rightarrow \phi(x)^\bullet$ induce bijections.*

Occurrence nets Two nodes x, x' of a net N are *in conflict,* written $x \# x'$, if there exist transitions $t, t' \in \mathcal{T}$ such that (i) $t \neq t'$, (ii) ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$, and (iii) $t \leqslant x$ and $t' \leqslant x'$. A node x is said to be in *self-conflict* iff $x \# x$.

Definition 2. *An* occurrence net *(ON) is a net $ON = (\mathcal{B}, \mathcal{E}, \rightarrow)$, with the elements of $\mathcal{B}$ called* conditions *and those of $\mathcal{E}$* events*, satisfying the additional properties:*

1. *no self-conflict:* $\forall x \in \mathcal{B} \cup \mathcal{E} : \neg[x \# x]$;
2. $\leqslant$ *is a partial order:* $\forall x \in \mathcal{B} \cup \mathcal{E} : \neg[x < x]$;
3. $\forall x \in \mathcal{B} \cup \mathcal{E} : |\{x' : x' < x\}| < \infty$;
4. *no backward branching:* $\forall b \in \mathcal{B} : |{}^\bullet b| \leq 1$.

Occurrence nets are useful to represent executions of Petri nets, see below: essential dynamical properties are visible via the topological structure of the bipartite graph—unlike for Petri nets. We assume that the set $\mathbf{c}_0 \triangleq \min(ON)$ of minimal nodes of ON is contained in $\mathcal{B}$. Nodes x and x' are *concurrent,* written $x \, co \, x'$, if neither $x \leqslant x'$, nor $x' \leqslant x$, nor $x \# x'$ hold. A *co-set* is a set $\mathcal{A}$ of pairwise concurrent conditions; a maximal co-set $\mathcal{A}$ w.r.t. set inclusion is called a *cut.*

Petri nets The static structure of net is given a dynamics via markings : A *marking* of net N is a multi-set $M : \mathcal{P} \longmapsto \{0, 1, 2, \dots\}$ of places.

Definition 3. *A* Petri net *(PN) is a pair $\mathcal{N} = (N, M_0)$, where N is a net with* finite *node set, and M_0 an* initial *marking. $\mathcal{T} \in \mathcal{T}$ is* enabled *at M, written $M \xrightarrow{t}$, if $M(p) > 0$ for every $p \in {}^\bullet t$. If $M \xrightarrow{t}$, t can* fire*, leading to $M' = (M - {}^\bullet t) + t^\bullet$; write $M \xrightarrow{t} M'$. The set $\mathbf{R}(M_0)$ of* reachable *markings of $\mathcal{N}$ is the smallest set $\mathbf{R}(M_0)$ containing M_0 and such that $M \in \mathbf{R}(M_0)$ and $M \xrightarrow{t} M'$ together imply $M' \in \mathbf{R}(M_0)$. Petri net $\mathcal{N}$ is* safe *if $M(\mathcal{P}) \subseteq \{0, 1\}$ for every $M \in \mathbf{R}(M_0)$. If $\mathcal{N}$ is safe, it has a finite number of reachable states; denote this number as $K(\mathcal{N})$.*

We consider only safe Petri nets here, hence markings can be regarded as place sets.

Branching processes and unfoldings Occurrence nets can serve as a semantics for Petri nets, i.e. a Petri net's behaviour can be unfolded into an occurrence net whose structure represents the possible executions:

Definition 4. *A* branching process *of Petri net $\mathcal{N}$ is a pair $\pi = (N, \phi)$, where N is an occurrence net, and ϕ is a homomorphism from N to N, such that:*

- *the restriction of ϕ to $min(N)$ is a bijection between $min(N)$ and M_0 (the set of initially marked places), and*
- *for all $e, e' \in \mathcal{E}$, ${}^\bullet e = {}^\bullet e'$ and $\phi(e) = \phi(e')$ together imply $e = e'$.*

For π, π' two branching processes, π' is a prefix *(see below) of π, written $\pi' \sqsubseteq \pi$, if there exists an injective homomorphism ψ from N' into N, such that ψ induces a bijection between the initial cuts $\mathbf{c}_0$ and $\mathbf{c}'_0$, and the composition $\phi \circ \psi$ coincides with ϕ'.*

By theorem 23 of [5], there exists a unique (up to an isomorphism) $\sqsubseteq$-maximal branching process, called the *unfolding* of $\mathcal{N}$ and denoted $\mathcal{U}_{\mathcal{N}}$.

Definition 5. *A sub-net*[2] $\mathcal{R}$ *of* N *is a* (structural) prefix *of* N*, written* $\mathcal{R} \sqsubseteq N$*, iff*

1. $\mathbf{c}_0 \subseteq \mathcal{R}$,
2. $\mathcal{R}$ *is* causally closed*: if* $x' \leqslant x$ *and* $x \in \mathcal{R}$*, then* $x' \in \mathcal{R}$*, and*
3. *for all events* e*,* $e \in \kappa$ *implies* $e^\bullet \subseteq \kappa$.

The set of prefixes is denoted **Pref**. *A prefix* κ *of* ON *is a* configuration *if* κ *is* conflict-free *, i.e. no two nodes from* κ *are in conflict; denote as* **Con** *the set of all configurations, and as* **FCon** *the subset of* **Con** *containing the finite configurations.*

Because of Condition 3, a finite configuration κ terminates at a cut, which we denote $\mathbf{c}(\kappa)$ or $\mathbf{c}_\kappa$. Denote the set of configurations as **Con**; a *maximal* configuration (w.r.t. set inclusion) is called a *run* and generically denoted ω; we denote the set of runs as Ω. For an event e, denote as $\lceil e \rceil$ the smallest configuration containing e, and as $\langle \lfloor e \rfloor \rangle$ the smallest configuration containing ${}^\bullet e$. We note that $(\mathbf{Pref}, \sqsubseteq)$ is a complete lattice (with union as join and intersection as meet operation), and that $(\mathbf{Con}, \sqsubseteq)$ is a complete lower semilattice having $(\mathbf{FCon}, \sqsubseteq)$ as a complete lower sub-semilattice. Two configurations κ and κ' are *compatible*, written $\kappa \parallel \kappa'$, iff $\kappa \cup \kappa'$ is a configuration. Obviously, $\kappa \sqsubseteq \kappa'$ implies $\kappa \parallel \kappa'$, but not vice versa. Note further that $\parallel$ is *not* an equivalence relation: in Figure 2, take as κ_0 the initial cut, $\kappa_1 \triangleq [\alpha_1]$, where α_1 is the grey α-labeled event, and $\kappa_2 \triangleq [\beta]$, with β the only β-labeled event on the lower right hand side of Figure 2. Then $\kappa_0 \parallel \kappa_1$ and $\kappa_0 \parallel \kappa_2$, but $\neg(\kappa_1 \parallel \kappa_2)$. For an event $e \notin \kappa$ such that ${}^\bullet e \subseteq \kappa$ and such that no event in κ is in conflict with e, κ can be *concatenated* with e, written $\kappa \odot e$; in this case, one deduces easily that ${}^\bullet e$ must be a coset of $\leqslant$-maximal conditions in κ, and the *concatenation* $\kappa \cdot e \triangleq \kappa \cup \{e\} \cup e^\bullet$ is the smallest configuration that contains $\{e\} \cup \kappa$. The relation $\odot$ and the concatenation operation extend to prefixes: write $\mathcal{R} \odot e$ iff *(i)* ${}^\bullet e \subseteq \mathcal{R}$ (i.e. ${}^\bullet e$ is not necessarily maximal) and *(ii)* $e \in \mathcal{E} \backslash \mathcal{R}$, and denote, in that case, as $\mathcal{R} \cdot e \triangleq \mathcal{R} \cup \{e\} \cup e^\bullet$ the *concatenation* of $\mathcal{R}$ and e. The concatenation operation generalizes to pairs of configurations, in the following way.

Shift of configurations Any finite configuration $\kappa_{\mathcal{N}}$ from the unfolding $\mathcal{U} = (\mathcal{B}, \mathcal{E}, \rightsquigarrow, \phi)$ of $\mathcal{N} = (N, M_0)$ induces a *shift* from the original net $\mathcal{N} = (N, M_0)$ to $\mathcal{N}_\kappa = (N, M_\kappa)$ with unfolding $\mathcal{U}_\kappa = (\mathcal{B}_\kappa, \mathcal{E}_\kappa, \rightsquigarrow_\kappa, \phi_\kappa)$, where

$$M_\kappa \quad \triangleq \quad \phi\left(\mathbf{c}\left(\kappa\right)\right)$$

is the marking associated to the cut generated by κ. Then, for any $\overline{\kappa} \in \mathbf{Con}_\kappa$, there is a unique configuration $\kappa \circ \overline{\kappa} \in \mathbf{Con}$ that has κ as prefix and such that $(\kappa \circ \overline{\kappa}) \backslash \kappa$ is isomorphic,

[2] we will not distinguish between κ seen as a *net* and as a *set* of nodes, using set extensions etc. to operate on the corresponding nets, obtained as the induced subnets of the ambient net.

as a labelled graph, to $\overline{\kappa}$. If, conversely, $\kappa_1 \in \mathbf{Con}$ satisfies $\kappa \circ \kappa_2$ with $\kappa \in \mathbf{FCon}$, i.e. κ_2 is the suffix of κ_1 after κ, we say that κ_2 is the *κ-shift of* κ_1, written

$$\kappa_2 \quad = \quad \theta_\kappa \kappa_1. \tag{2}$$

Further, if $\kappa_1 \circ \kappa_2^n \circ \kappa_3 \in \mathbf{Con}$ for all $n \in \mathbb{N}$, write

$$\kappa_1 \circ \kappa_2^+ \circ \kappa_3 \quad \triangleq \quad \{\kappa_1 \circ \kappa_2^n \circ \kappa_3 \mid n \in \mathbb{N}\}.$$

Running example In Figure 2, a Petri net $\mathcal{N}$ is shown on the left, a branching process $N = (ON, \phi)$ of $\mathcal{N}$ on the right hand side. Conditions are labeled by places, events by transitions. A configuration is shown in grey. The mechanism for constructing the unfolding of Petri net $\mathcal{N}$ is illustrated in the middle. Informally, take the three conditions labeled by the initial marking of $\mathcal{N}$ as the minimal branching process of $\mathcal{N}$. Then, for each branching process $\pi = (N, \phi)$ already constructed, select a co-set $\mathcal{A}$ of N that (i) is labeled by the preset ${}^\bullet t$ of some transition t of $\mathcal{N}$, and (ii) has no t-labeled event in its postset within N. Append to $\mathcal{A}$ a net isomorphic to ${}^\bullet t \to t \to t^\bullet$ (recall ${}^\bullet t = \phi(\mathcal{A})$), and label its additional nodes by t and $t^\bullet$, respectively. One thus obtains recursively all possible finite branching processes of $\mathcal{N}$; their union is the unfolding $\mathcal{U}_\mathcal{N}$.

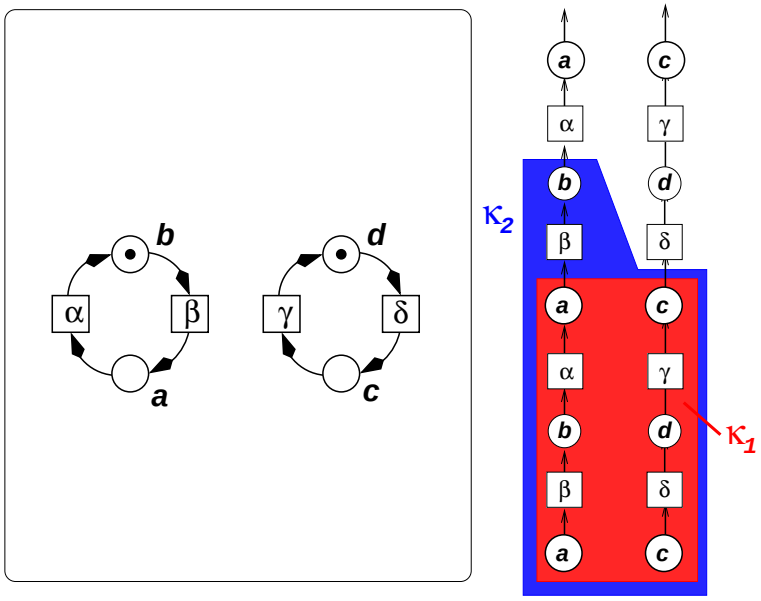


Figure 3: *On upper and lower heights of configurations : example 1, unfolding (right) of a two-component Petri net (left)*

Height To measure linearly ordered words, the length is simply given by the number of letters. This can be applied to partial orders as well, yet for diagnosability issues, *height* is a much more adequate measure: For a prefix $\mathcal{R} \in \mathbf{Pref}$, the *height* $\mathcal{H}(\mathcal{R})$ of $\mathcal{R}$ is the maximal length of strings of causally ordered events in $\mathcal{R}$. It is defined inductively as follows:

Definition 6. *Let* $\mathfrak{l} : \mathcal{E} \to \mathbb{R}_{\geqslant 0} \triangleq [0, +\infty)$ *a* length function, $e \in \mathcal{E}$, $\mathcal{R} \in \mathbf{Pref}(N)$. *Set*

$$\mathcal{H}_{\mathfrak{l}}(\lceil e \rceil) \triangleq \mathfrak{l}(e) + \mathcal{H}_{\mathfrak{l}}(\lfloor e \rfloor) \quad (3)$$

$$\mathcal{H}_{\mathfrak{l}}(\mathcal{R}) \triangleq \sup\{\mathcal{H}_{\mathfrak{l}}(\lceil e \rceil) \mid e \in \mathcal{E} \cap \mathcal{R}\}, \quad (4)$$

where $\sup(\emptyset) \triangleq 0$, *hence* $\mathcal{H}_{\mathfrak{l}}(\mathbf{c}_0) = 0$.

Prime prefixes Fix a length function $\mathfrak{l}$. For $n \in \mathbb{N}_0$, let $\mathbf{Pref}_n^{\mathfrak{l}}$ be the set of prefixes whose height does not exceed n, i.e. set

$$\mathbf{Pref}_n^{\mathfrak{l}} \triangleq \{\mathcal{R} \in \mathbf{Pref} \mid \mathcal{H}(\mathcal{R}) \leqslant n\};$$

each $\mathbf{Pref}_n^{\mathfrak{l}}$ has a (unique) $\sqsubseteq$-maximum

$$\mathbf{Pref}_n^{\mathfrak{l}} \ni \mathcal{R}_n^{\mathfrak{l}} = \bigcup_{\mathcal{R} \in \mathbf{Pref}_n^{\mathfrak{l}}} \mathcal{R}.$$

Call $\mathcal{R}_n^{\mathfrak{l}}$ the *nth prime prefix* for $\mathcal{N}$ and $\mathfrak{l}$; any configuration κ_n such that there exists $\omega \in \Omega$ with $\kappa_n = \omega \cap \mathcal{R}_n^{\mathfrak{l}}$ is called an *(nth) prime configuration* for $\mathfrak{l}$. We will assume henceforth that $\mathfrak{l}(e) \in \{0, 1\}$ for all e : "invisible" events will be assigned 0, all others "counted", i.e. assigned 1. When no confusion can arise, we will drop the explicit mention of $\mathfrak{l}$.

Upper / lower height and progress in configurations Any finite configuration κ is itself a prefix, and thus Definition 6 can be applied to obtain $\mathcal{H}(\kappa)$. Besides this height function, which we will henceforth call *upper height*, we introduce the *lower height* $\underline{\mathcal{H}}(\kappa)$ of κ as follows:

Definition 7. *Fix a length function* $\mathfrak{l}$. *With the above notations, set*

$$\underline{\mathcal{H}}(\kappa) \triangleq \sup\{n \in \mathbb{N} \mid \exists\, \omega \in \Omega . \omega \sim_{\mathcal{R}_n} \kappa\}, \quad (5)$$

where $\sup(\emptyset) = +\infty$.

Example 1. *On the right hand side of Figure 3, assuming* $\mathfrak{l} \equiv 1$, *we have* $\mathcal{H}(\kappa_1) = \underline{\mathcal{H}}(\kappa_1) = 2$ *and* $\mathcal{H}(\kappa_2) = 3$, *whereas* $\underline{\mathcal{H}}(\kappa_1) = 2 < \mathcal{H}(\kappa_2)$. *It is tempting to believe that evaluating the lower height of a configuration reduces to taking the minimal height of a maximal event : for* κ_2, *the first occurrence of* γ *is maximal, and its height yields* $\underline{\mathcal{H}}(\kappa_2)$. *This would of course simplify the above definition, but it would be* **wrong**; *lower height is not quite as local a property as it seems. Consider the occurrence net in Figure 4. Configuration* κ' *is a proper prefix of* κ; *the prime prefixes are* $\mathcal{R}_0 = \{a, b\}$, $\mathcal{R}_1 = \kappa_1$, *and* $\mathcal{R}_2 = \mathcal{R}_1 \cup \{\gamma, c\}$. *One finds directly that* $\underline{\mathcal{H}}(\kappa) = \mathcal{H}(\kappa) = \mathcal{H}(\kappa') = 1$; *more surprisingly, one also has* $\underline{\mathcal{H}}(\kappa') = 1$. *In fact,* $\omega' \sim_{\mathcal{R}_1} \kappa'$; *and* 1 *is the maximal value with this property since* $\omega' \not\sim_{\mathcal{R}_2} \kappa'$.

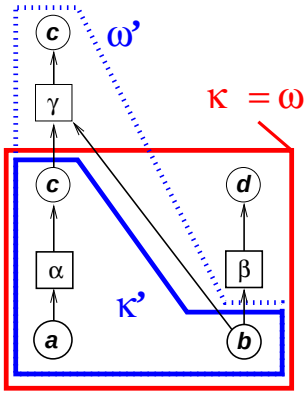


Figure 4: *Configurations κ and κ' have the same lower heights*

Of course, $\underline{\mathcal{H}}(\kappa) \leqslant \mathcal{H}(\kappa)$. Note that equality holds iff

$$\forall\, \omega,\ \omega' \in \Omega\ :\ \left[(\kappa \sqsubseteq \omega\ \wedge\ \kappa \sqsubseteq \omega') \quad \Rightarrow \quad \left(\omega \sim_{\mathcal{R}_{\mathcal{H}(\kappa)}} \kappa\right)\right]\ ; \tag{6}$$

denote the configurations that satisfy (6) as **progressive**. Note that $\kappa \in \mathbf{Con}$ is progressive iff for all $\kappa' \in \mathbf{Con}$ such that $\kappa \parallel \kappa'$ and $\mathcal{H}_{\mathfrak{l}}(\kappa) = \mathcal{H}_{\mathfrak{l}}(\kappa')$, it holds that $\kappa' \sqsubseteq \kappa$. In Figure 2, κ shaded in grey is progressive: with $\mathfrak{l} \equiv 1$, one has $\mathcal{H}_{\mathfrak{l}}(\kappa) = 2$, and any other configuration of that height is either a prefix of κ or incompatible. The notion of progressive configuration serves to discern behaviours according to "balance": in progressive behaviours, all sub-processes advance at an (approximately) equal pace, no local process is left behind. In Figure 2, not all configurations are progressive: take for example the sub-configuration of the grey κ with the v-labeled event chopped off. More drastically, the net has an infinite run with only β-and γ-labeled events; on this run, there is an infinity of non-progressive configurations. Only the occurrence of iv will lead to progressive configurations; in fact, this is the only progress possible in component 2 as long as i never fires.

Remark: *our notion of progressive configuration captures the* progress assumption *often used in the literature on distributed algorithms, see [28] for an overview.*

In Figure 3, the situation is more symmetric with respect to both components: all configurations are compatible with the unique run. However, there are non-progressive configurations that can be decomposed into an infinite execution of one component and a finite execution of the other. We also note:

$$\underline{\mathcal{H}}(\kappa) = +\infty \quad \Rightarrow \quad \kappa \in \Omega. \tag{7}$$

The converse is not true since maximal runs may be finite.

3 Asynchronous Diagnosis

Computation of Diagnosis We recall briefly the diagnosis approach of [9]: the asynchronous diagnosis problem is solved using *diagnosis nets*, introduced to express the solution of asynchronous diagnosis: Compute branching processes of the product net $\mathcal{N} \times \mathcal{A}$ obtained from $\mathcal{N}$ and an alarm pattern $\mathcal{A}$, where $\mathcal{A}$ is given as a configuration (unbranched occurrence net) itself. The product glues together transitions of $\mathcal{N}$ with corresponding alarms in $\mathcal{A}$ (this is the *synchronized product* for Petri nets); the unfolding $\mathcal{U}_{\mathcal{N}\times\mathcal{A}}$ thus obtained consists of all the explanations that $\mathcal{N}$ can give for $\mathcal{A}$. In fact, the configurations κ of $\mathcal{N}$ that *explain* $\mathcal{A}$ are those for which $\mathcal{U}_{(\mathcal{N}\times\mathcal{A})}$ contains a corresponding configuration $\overline{\kappa}$ whose projection (i) to the alarm set yields $\mathcal{A}$, and (ii) to $\mathcal{N}$-nodes yields κ. In Figure 5, diagnosis is shown for the running example. In the alarm pattern on the left hand side, consider only the initial segment with a dark background; the center par on the right hand side, call it $\mathcal{A}$. The center part shows the unfolding of the product $\mathcal{N} \times \mathcal{A}$. Note that the product actually multiplies some events by joining different alarm histories. Further, note the structural conflict indicated by "#" between the two ρ-labeled events at the bottom; it is inherited from the branching condition labeled "ii" above, with"#" indicating the immediate conflict at "ii". This conflict reflects the restriction imposed by the alarmpattern: since $\mathcal{A}$ contains only one ρ-labeled event, no explanation can contain more than one such event. The right hand part of Figure 5 now shows the projection of $\mathcal{U}_{\mathcal{N}\times\mathcal{A}}$ to elements of $\mathcal{N}$, i.e. a prefix of $\mathcal{U}_{\mathcal{N}}$; the projection enriches this prefix with the inherited conflict between the two ρ's which is the contribution from the observation. The diagnosis set thus consists of it is justified by the branching three configurations : the one with all events white in the figure, and the two configurations formed by the shaded events with labels α and β plus one of the two shaded ρ's. The interested reader may continue unfolding with the *full* alarm pattern, and will notice that the "white" configuration will be ruled out since it does not allow for a continuation with a second α-event as required by the alarm pattern.

Now, for the formal definitions. An alarm pattern is an $\mathfrak{A}$-labeled partially ordered set; we can formally represent an alarm pattern as a conflict-free $\mathfrak{A}$-labeled occurrence net $\mathcal{A} = (\mathcal{B}_\mathcal{A}, \mathcal{E}_\mathcal{A}, \rightsquigarrow_\mathcal{A}, \lambda_\mathcal{A})$, obtained as follows:

1. $\lambda_\mathcal{A} : \mathcal{E}_\mathcal{A} \to \mathfrak{A}\backslash\varepsilon$.
2. To obtain the conditions of $\mathcal{B}_\mathcal{A}$, we add two auxiliary elements $\top, \bot$ such that for all $e \in \mathcal{E}_\mathcal{A}$, one has $\bot < \alpha < \top$, then compute the predecessor relation $\prec_O$:

$$e \prec_O e' \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad e < e' \wedge \forall e'' : \; e < e' < e' \Rightarrow e'' \in \{e, e'\},$$

and set

$$\mathcal{B}_\mathcal{A} \quad \triangleq \quad \{(e, e') \mid e \prec_O e'\} \cup \{(\bot e') \mid \bot \prec_O e\} \cup \{(e, \top) \mid e \prec_O \top\}.$$

Denote the initial cut of $\mathcal{A}$ as $\mathbf{c}_\mathcal{A}$; clearly,

$$\mathbf{c}_\mathcal{A} \quad = \quad \{(e, \top) \mid e \prec_O \top\}.$$

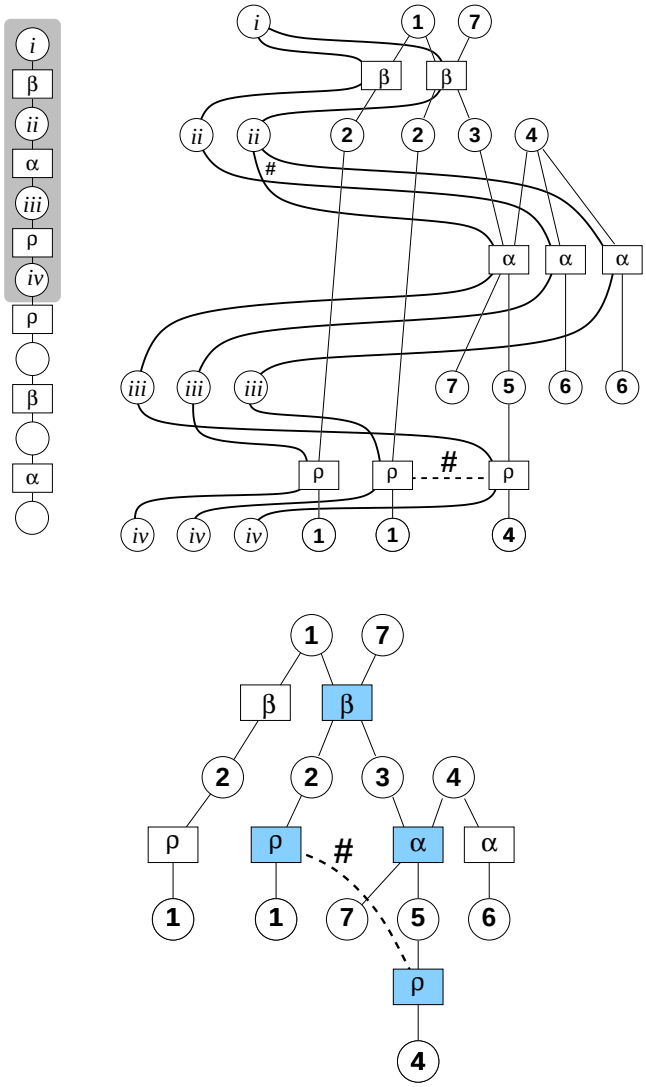


Figure 5: Illustration of Diagnosis for the running example. Left: Alarm pattern $\mathcal{A}$ (only the part shaded in grey is used); center: unfolding $\mathcal{U}_{\mathcal{N}\times\mathcal{A}}$ of $\mathcal{N}\times\mathcal{A}$; right: projection of $\mathcal{U}_{\mathcal{N}\times\mathcal{A}}$ to nodes from $\mathcal{N}$, with extra conflict inherited under the projection

3. Set $\rightarrow_{\mathcal{A}} = \{(e,b),(b,e') \mid e \prec_O e' \wedge b = (e,e')\}$.

For given alarm pattern $\mathcal{A}$ and model $\mathcal{N}$, let $\lambda : \mathcal{T} \rightarrow \mathfrak{A}$ and $\mathcal{A} : \mathcal{E}_{\mathcal{A}} \rightarrow \mathfrak{A}\backslash\{\varepsilon\}$ be the respective labelling functions, and consider the synchronized product of $\mathcal{N}$ and $\mathcal{A}$ w.r.t. $\mathfrak{A}$, that is: the Petri net $\mathcal{N} \times \mathcal{A} = (\mathcal{P}_{\mathcal{N}\times\mathcal{A}}, \mathcal{T}_{\mathcal{N}\times\mathcal{A}}, \rightarrow_{\mathcal{N}\times\mathcal{A}}, \lambda_{\mathcal{N}\times\mathcal{A}}, M_{\mathcal{N}\times\mathcal{A}})$, where $\mathcal{P}_{\mathcal{N}\times\mathcal{A}}$ is the disjoint union of $\mathcal{P}$ and $\mathcal{B}$,

$$
\begin{array}{lcl}
\mathcal{T}_{\mathcal{N}\times\mathcal{A}}n & \triangleq & \{t \in \mathcal{T} \mid \lambda(t) = \varepsilon\} \cup \{\langle t,e\rangle \in \mathcal{T}\times\mathcal{E}_{\mathcal{A}} \mid \lambda(t) = \lambda_{\mathcal{A}}(e)\} \\
\rightarrow_{(\mathcal{N}\times\mathcal{A})} & \triangleq & \{(p,t) \in \mathcal{P}\times\mathcal{T} \mid \lambda(t) = \varepsilon \ \wedge \ p \rightarrow t\} \cup \{(t,p) \in \mathcal{T}\times\mathcal{P} \mid \lambda(t) = \varepsilon \ \wedge \ t \rightarrow p\} \\
 & & \cup \{(p,\langle t,e\rangle) \in \mathcal{P}\times(\mathcal{T}\times\mathcal{E}_{\mathcal{A}}) \mid \lambda(t) = \lambda_{\mathcal{A}}(e) \wedge \ p \rightarrow t\} \\
 & & \cup \{(\langle t,e\rangle,p) \in (\mathcal{T}\times\mathcal{E}_{\mathcal{A}})\times\mathcal{P} \mid \lambda(t) = \lambda_{\mathcal{A}}(e) \wedge \ t \rightarrow p\} \\
 & & \cup \{(b,\langle t,e\rangle) \in \mathcal{B}_{\mathcal{A}}\times(\mathcal{T}\times\mathcal{E}_{\mathcal{A}}) \mid \lambda(t) = \lambda_{\mathcal{A}}(e) \wedge \ b \rightarrow_{\mathcal{A}} e\} \\
 & & \cup \{(\langle t,e\rangle,b) \in (\mathcal{T}\times\mathcal{E}_{\mathcal{A}})\times\mathcal{B}_{\mathcal{A}} \mid \lambda(t) = \lambda_{\mathcal{A}}(e) \wedge \ e \rightarrow_{\mathcal{A}} b\} \\
M_{\mathcal{N}\times\mathcal{A}} & \triangleq & M_0 \cup \mathbf{c}_{\mathcal{A}},
\end{array}
$$

and $\lambda_{\mathcal{N}\times\mathcal{A}}$ is given by $\lambda_{\mathcal{N}\times\mathcal{A}}(\langle t,e\rangle) = \lambda(t) = \lambda_{\mathcal{A}}(e)$ for synchronization transitions, and by λ and $\lambda_{\mathcal{A}}$ otherwise. Denote as $\mathbf{proj}_{\mathcal{N}}$ and $\mathbf{proj}_{\mathcal{A}}$ the projections from $\mathcal{N}\times\mathcal{A}$ to $\mathcal{N}$ and $\mathcal{A}$, respectively.

Diagnosis set Consider now the unfolding $\mathcal{U}_{(\mathcal{N}\times\mathcal{A})}$, and denote the set of its (finite) configurations as $\mathbf{Con}_{(\mathcal{N}\times\mathcal{A})}$, and the subset of *finite* configurations as $\mathbf{FCon}_{(\mathcal{N}\times\mathcal{A})}$. For every $\kappa_{(\mathcal{N}\times\mathcal{A})} \in \mathbf{Con}_{(\mathcal{N}\times\mathcal{A})}$, denote - by abuse of notation - as $\mathbf{proj}_{\mathcal{N}}(\kappa_{(\mathcal{N}\times\mathcal{A})})$ and $\mathbf{proj}_{\mathcal{A}}(\kappa_{(\mathcal{N}\times\mathcal{A})})$ the "pure" subnets of $\mathcal{U}_{\mathcal{N}}$ and $\mathcal{A}$, respectively, that are induced by the projections $\mathbf{proj}_{\mathcal{N}}$ and $\mathbf{proj}_{\mathcal{A}}$.

For diagnosis, we are interested in those configurations of the net model that *explain* $\mathcal{A}$; according to Theorem 1 of [9], these configurations are the elements of

$$\mathbf{diag}\,(\mathcal{A}) \ \triangleq \ \{\kappa \in \mathbf{Con} \mid \exists\, \overline{\kappa} \in \mathbf{Con}_{\mathcal{N}\times\mathcal{A}} : \ \mathbf{proj}_{\mathcal{A}}\,(\overline{\kappa}) = \mathcal{A} \ \textit{and} \ \mathbf{proj}_{\mathcal{N}}\,(\overline{\kappa}) = \kappa\}\,. \quad (8)$$

We note that for all $\kappa \in \mathbf{Con}$, there is exactly one alarm pattern $\mathcal{A}$ according to λ such that $\kappa \in \mathbf{diag}(\mathcal{A})$, which we denote as $\mathbf{patt}(\kappa)$.

Let $\mathbf{Con}_O$ be the set of all effective alarm patterns of $\mathcal{N}$, that is

$$\mathbf{Con}_O \ \triangleq \ \{\mathcal{A} | \exists\, \kappa \in \mathbf{Con} : \ \mathcal{A} \in \mathbf{patt}(\kappa)\}.$$

4 Diagnosability

The question is now what the system and the diagnosis setup have to satisfy for this method to detect effectively all faults: that is, we want to formally characterize *diagnosability*. Intuitively, the properties that can prevent a system from being diagnosable, as in the classical automaton setting, arise from the possibility of invisible cycles allowing executions of arbitrary length without a decision about failure. However, the partial order setting requires extra care in re-defining "cycles", as it did for "length".

4.1 Preparations and Main Definition

Definition 8. *Let $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \rightarrow, M_0)$ be a Petri net with unfolding $\mathcal{U} = (\mathcal{B}, \mathcal{E}, \rightsquigarrow, \phi)$, and $\mathfrak{A}$ an alarm alphabet containing the empty word ε; further, let $\chi : \mathcal{T} \rightarrow \mathfrak{A}$, for $\mathfrak{A}$ some non-empty alphabet, be a* labelling function *associating alarms to system transitions.*

- *Call* invisible *or* unobservable *transitions the elements of $UO \triangleq \chi^{-1}(\varepsilon)$, and let*
- *$O \triangleq \mathcal{T} \backslash UO$ be the set of* observable *transitions, and*
- *$\Phi \subseteq UO$ the set of* faults *to be diagnosed.*

Here, $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \rightarrow, M_0)$ is the underlying "true" system, with the places in $\mathcal{P}$ representing the unobservable local state variables. This framework allows for *erasing* (i.e. labeling by ε) and *ambiguity* (the same label for distinct events). Without loss of generality, $\Phi \cap O = \emptyset$: a fault that is indicated by an alarm needs not be diagnosed; the diagnosis problem concerns *silent* faults, whose associated "alarm" is ε. Further, set $\mathcal{E}_\Phi \triangleq \phi^{-1}(\Phi)$, $\mathcal{E}_O \triangleq \phi^{-1}(O)$, and $\mathcal{E}_{UO} \triangleq \mathcal{E} \backslash \mathcal{E}_O$. Denote as $\mathcal{L} \triangleq \mathbf{FCon}(\mathcal{N})$ the set of $\mathcal{N}$'s finite configurations, and as $\mathcal{L}_{\mathbf{prog}}$ the set of *progressive* configurations; observe that $\mathcal{L}$ and $\mathcal{L}_{\mathbf{prog}}$ are prefix closed, and partially ordered by $\sqsubseteq$. For $\kappa \in \mathcal{L}$ let $M(\kappa)$ be the marking obtained after κ, and as κ_O the labeled partial order induced by κ on $\kappa \cap \mathcal{E}_O$, and write

$$\kappa \sim_O \kappa' \quad \textit{iff} \quad \kappa_O \textit{ and } \kappa'_O \textit{ are isomorphic partial orders} ;$$

$\sim_O$ is an equivalence. Further, let $\sim_M$ and $\sim_\Phi$ be the equivalences on $\mathcal{L}$ given by

$$\begin{aligned} \kappa \sim_M \kappa' \quad &\textit{iff} \quad M(\kappa) = M(\kappa') \\ \kappa \sim_\Phi \kappa' \quad &\textit{iff} \quad [\kappa \cap \mathcal{E}_\Phi = \emptyset \iff \kappa' \cap \mathcal{E}_\Phi = \emptyset] . \end{aligned}$$

Some further preparations are in order before defining diagnosability.

Height revisited In the definition of height given above, we mentioned that the length function $\mathfrak{l}$ can take values $\mathfrak{l}(e) = 0$ or $\mathfrak{l}(e) = 1$, depending on whether e is visible or not. In fact, consider the length functions $\mathfrak{l}, \mathfrak{l}' : \mathcal{E} \rightarrow \{0, 1\}$ given by: $\mathfrak{l} \equiv 1$, and $\mathfrak{l}'$ is the indicator function of O, i.e. $\mathfrak{l}'(e) = 1$ if $e \in O$ and $\mathfrak{l}'(e) = 0$ otherwise.

Definition 9. *Let $\mathcal{H} \triangleq \mathcal{H}_\mathfrak{l}$ and $\mathcal{H}_O \triangleq \mathcal{H}_{\mathfrak{l}'}$, and denote as $\underline{\mathcal{H}}$ and $\underline{\mathcal{H}}_O$, respectively, the associated lower heights according to Definition 7.*

Then $\mathcal{H}$ ($\underline{\mathcal{H}}$) is simply "counting height" ("lower counting height") for configurations, and $\mathcal{H}_O$ ($\underline{\mathcal{H}}_O$) measures the "observable height" ("lower observable height") of a configuration.

Liveness In [30], the authors assumed *liveness*, i.e. that all finite executions can be extended. Let us say that a configuration κ is **dead** iff $\kappa \sqsubseteq \kappa'$ implies $\kappa' \in \mathbf{FCon}$. We will take those configurations into account in the definition of diagnosability given below; we consider the presence of dead configurations as no obstacle to diagnosability : on a finite run, absence or presence of faults can eventually be verified, it is *infinite runs* that may pose problems for fault diagnosis. Hence we define:

Definition 10. *Petri net $\mathcal{N}$ satisfies*

- *($\mathcal{OBS}$) iff for all $\kappa, \kappa' \in \mathcal{L}$:*

$$(\kappa \sqsubseteq \kappa') \wedge (\kappa \neq \kappa') \wedge (\kappa \sim_M \kappa') \quad \Rightarrow \quad (\kappa \not\sim_O \kappa') \tag{9}$$

- *($\mathcal{WOBS}$) iff (9) holds for all $\kappa, \kappa' \in \mathcal{L}_{\mathbf{prog}}$.*

Defining Diagnosability These considerations motivate the following definition, which is central for the present article.

Definition 11. *[Diagnosability] A safe Petri Net $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \rightarrow, M_0)$ is called*

1. (strongly) diagnosable *w.r.t. O and Φ iff*
   (a) *$\mathcal{N}$ satisfies ($\mathcal{OBS}$), and*
   (b) *there exists $n \in \mathbb{N}$ such that for all $\kappa_\Phi \in \mathcal{L}$ having a maximal event $e \in \mathcal{E}_\Phi$, it holds that every $\kappa \in \mathcal{L}$ such that*
      i. *$\kappa_\Phi \sqsubseteq \kappa$,*
      ii. *κ is not dead, and*
      iii. *$\mathcal{H}(\kappa) \geq \mathcal{H}(\kappa_\Phi) + n$*

   satisfies:

$$\forall \kappa' \in \mathcal{L} : \ \kappa' \sim_O \kappa \quad \Rightarrow \quad \mathcal{E}_\Phi \cap \kappa' \neq \emptyset; \tag{10}$$

2. weakly diagnosable *w.r.t. O and Φ iff*
   (a) *$\mathcal{N}$ satisfies ($\mathcal{WOBS}$), and*
   (b) *there exists $n \in \mathbb{N}$ such that (10) holds for all $\kappa \in \mathcal{L}_{\mathbf{prog}}$ such that*
      i. *$\kappa_\Phi \sqsubseteq \kappa$,*
      ii. *κ is not dead, and*
      iii. *$\mathcal{H}(\kappa) \geq \mathcal{H}(\kappa_\Phi) + n$.*

We will say that $\mathcal{N}$ satisfies $\mathbb{D}$ iff it is strongly, and $\mathbb{W}$ iff it is weakly diagnosable.

Strong diagnosability trivially implies weak diagnosability. Figure 3 illustrates that the converse is not true: Suppose β is a fault event to be detected, $O = \{\alpha\}$, and for $m \in \mathbb{N}$, let κ_m be the smallest configuration such that *(i)* β never occurs on κ_m, and *(ii)* δ occurs exactly n times on κ_m. Then the height of κ_m is $\mathcal{H}(\kappa_m) = 2m + 1$, yet $\kappa_m \sim_O \kappa_1$, so we conclude that the system is not strongly diagnosable. On the other hand, the κ_m are not progressive. For weak diagnosability, note that since all *progressive* configurations of height at least $2k + 1$ contain at least k instances of $\alpha \in O$, from which it follows directly that the system is weakly diagnosable.

4.2 Observable Diagnosability

Note that the conditions for $\mathbb{D}$ and $\mathbb{W}$ given in Definition 10 use the counting height, which is of no great use in practice since it depends on unobservable events. Hence the following definition is of interest, and gives the analogue of that in [30] (compare (1)):

Definition 12. *Define properties $\mathbb{D}_O$ and $\mathbb{W}_O$ by replacing $\mathcal{H}$ by $\mathcal{H}_O$ in the conditions for $\mathbb{D}$ and $\mathbb{W}$, respectively, in Definition 10.*

Since $\mathcal{H}_O \leqslant \mathcal{H}$, we immediately obtain

Lemma 1. *$\mathbb{D}$ implies $\mathbb{D}_O$ and $\mathbb{W}$ implies $\mathbb{W}_O$.*

However, more is true:

Lemma 2. *If $\mathcal{OBS}$ holds for $\mathcal{N}$, then $\mathbb{D}_O$ implies $\mathbb{D}$, and under $\mathcal{WOBS}$, $\mathbb{W}_O$ implies $\mathbb{W}$.*

Proof: It suffices to show, in $\mathcal{L}$ and $\mathcal{L}_{\mathbf{prog}}$, that the existence of some $n_O \in \mathbb{N}$ such that κ not dead, $\kappa_\Phi \sqsubseteq \kappa$, and

$$\mathcal{H}_O(\kappa) \quad \geq \quad \mathcal{H}_O(\kappa_\Phi) + n_O,$$

implies (10), implies existence of n such that (10) holds for all κ not dead, $\kappa_\Phi \sqsubseteq \kappa$, and $\mathcal{H}(\kappa) \geq \mathcal{H}(\kappa_\Phi) + n$, (for any faulty configuration κ_Φ) First, note that n can be replaced by any $n' > n$ without falsifying the hypothesis. Now, since there are only $K = K(\mathcal{N}) \in \mathbb{N}$ reachable states,

$$\mathcal{H}(\kappa) \quad \geq \quad \mathcal{H}(\kappa_\Phi) + K$$

implies that $\kappa \not\sim_O \kappa_\Phi$: in fact, $\kappa \backslash \kappa_\Phi$ must contain observable eventsn since silent cycles are ruled out by $\mathcal{OBS}$,. Reasoning in this way shows that, if $|\mathcal{P}|$ denotes the (finite) number of places in $\mathcal{N}$, then for every $m \in \mathbb{N}$,

$$\mathcal{H}(\kappa) \quad \geq \quad \mathcal{H}(\kappa_\Phi) + K \cdot |\mathcal{P}| \cdot m$$

implies

$$\mathcal{H}_O(\kappa) \quad \geq \quad \mathcal{H}_O(\kappa_\Phi) + m;$$

thus the desired implication holds. □

In fact, if there exist invisible cycles, $\mathbb{D}_O$ does not imply $\mathbb{D}$ and $\mathbb{W}_O$ does not imply $\mathbb{W}$: in the worst case, a faulty configuration κ_Φ may be lead to a silent cycle after the fault, in such a way that all runs ω such that $\kappa_\Phi \sqsubseteq \omega$ are infinite, and satisfy $\omega \sim_O \kappa$. Then, the conditions in $\mathbb{D}_O$ and $\mathbb{W}_O$ on the extensions κ of κ_Φ hold vacuously : one always has $\mathcal{H}_O(\kappa) = \mathcal{H}_O(\kappa_\Phi)$, but, clearly, $\mathbb{D}$ and $\mathbb{W}$ are violated.

We conclude that $\mathcal{OBS}/\mathcal{WOBS}$ makes the observational and non-observational definitions of diagnosability equivalent, and should be required of distributed systems for diagnosability. Below, we will see that $\mathcal{OBS}$ can be effectively verified on a finite "complete" prefix of the unfolding.

5 Characterization of Diagnosability

Diagnosability is *violated* iff the system is able to perform two indiscernible, non-fault-equivalent cycles. That is, iff there are O-equivalent configurations κ_1 and κ_2 having O-equivalent extensions κ_1' and κ_2' such that κ_i' leads to the same marking M_i, and such that κ_1' and κ_2' are not Φ-equivalent; then the system may repeat that cyclic behavior indefinitely, without a decision about occurrence of faults. This is confirmed by our main result :

Theorem 1. *Let $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \rightarrow, M_0)$ a Petri net, with labelling $\lambda : \mathcal{T} \rightarrow \mathfrak{A}$, and Φ, O, UO, $\mathcal{L}$ and $\mathcal{L}_{\mathbf{prog}}$ as above. $\mathcal{N}$ is* **strongly diagnosable** *w.r.t. O and Φ iff it satisfies $\mathcal{OBS}$ and*

$$\forall\ \kappa_1, \kappa_2, \kappa_1', \kappa_2' \in \mathcal{L} : \left[\left\{ \begin{array}{rcl} & \kappa_1 \sim_O \kappa_2 \ \wedge\ \kappa_1' \sim_O \kappa_2' \ \wedge\ \kappa_1 \neq \kappa_1' & \\ \wedge & \forall\ i \in \{1,2\} : \left(\begin{array}{rcl} & \kappa_i \sim_M \kappa_i' & \\ \wedge & \kappa_i & \sqsubseteq\ \kappa_i' \end{array} \right) & \end{array} \right\} \Rightarrow \kappa_1' \sim_\Phi \kappa_2' \right] . \quad (11)$$

$\mathcal{N}$ is **weakly diagnosable** *w.r.t. O and Φ iff it satisfies $\mathcal{WOBS}$, and (11) holds restricted to $\mathcal{L}_{\mathbf{prog}}$.*

Note that (11) allows $\kappa_2 = \kappa_2'$ in the assumption.

Proof: We show the strong diagnosability case; the characterization of weak diagnosability is obtained replacing $\mathcal{L}$ by $\mathcal{L}_{\mathbf{prog}}$.

"only if" let $\kappa_i \sqsubseteq \kappa_i'$, $i \in \{1,2\}$, constitute a violation of (11), i.e.

1. without loss of generality, $\kappa_2' \cap \mathcal{E}_\Phi \neq \emptyset$ and $\kappa_1' \cap \mathcal{E}_\Phi = \kappa_1 \cap \mathcal{E}_\Phi = \emptyset$;
2. $\kappa_i' = \kappa_i \circ \mu_i$, such that $\mu_1 = \vartheta_{\kappa_1} \kappa_1'$ according to (2) contains at least one event,
3. $\kappa_i' \sim_O \kappa_i$ and $\kappa_i \sim_M \kappa_i'$.

From 2.), it follows that a copy of μ_i can be appended to κ_i' as well, and so forth; let κ_i^k be the configuration obtained after appending k copies of μ_i to κ_i. Observe that

$$\mathcal{H}(\kappa_1^k) \quad \geq \quad \max\left(k, \mathcal{H}\left(\kappa_1\right)\right). \quad (12)$$

Thus $\mathcal{H}(\kappa_1^k) \rightarrow \infty$ as $k \rightarrow \infty$. Now, by assumption we have $\kappa_2^k \sim_O \kappa_2$; further, by construction, $\mu_2 \cap \mathcal{E}_\Phi$ and therefore $\kappa_2^k \cap \mathcal{E}_\Phi = \emptyset$ It follows that (10) is violated.

"if" suppose (10) does *not* hold, i.e. for every $n \in \mathbb{N}$, there exists a configuration $\kappa(n) \in \mathcal{L}$ such that

1. some $e \in \mathcal{E}_\Phi$ is $\leqslant$-maximal in $\mathcal{E} \cap \kappa(n)$, and
2. there exist $\kappa_1(n), \kappa_2(n) \in \mathcal{L}$ such that

$$\kappa(n) \quad \sqsubseteq \quad \kappa_1(n) \quad (13)$$

$$\mathcal{H}(\kappa_1(n)) \quad \geq \quad \mathcal{H}(\kappa(n)) + n \quad (14)$$

$$\kappa_2(n) \sim_O \kappa_1(n) \quad \textit{and} \quad \kappa_2 \cap \mathcal{E}_\Phi = \emptyset. \quad (15)$$

Suppose first that one can choose κ_1' with $\kappa_1 \sqsubseteq \kappa_1' \sqsubseteq \kappa_1(n)$ such that $\kappa_1 \sim_M \kappa_1'$, $\kappa_1 \sim_O \kappa_1'$, and $\kappa_1 \neq \kappa_1'$; then we are done, taking $\kappa_2' \triangleq \kappa_2$. Hence we can assume that

$$\forall\ \kappa_1' : \left[\left\{ \begin{array}{l} \kappa_1 \sqsubseteq \kappa_1' \sqsubseteq \kappa_1(n) \\ \kappa_1 \sim_O \kappa_1' \\ \kappa_1 \sim_M \kappa_1' \end{array} \right\} \Rightarrow \kappa_1 = \kappa_1' \right]. \tag{16}$$

For any $\kappa_1 \sqsubseteq \kappa_1(n)$, let $U(\kappa_1, n)$ be the set of configurations $\kappa_2 \sqsubseteq \kappa_2(n)$ such that $\kappa_2 \sim_O \kappa_1$. For any reachable marking M of $\mathcal{N}$, let $S_1(M,n)$ be the set of configurations κ_1 such that *(i)* $\kappa_1 \sqsubseteq \kappa_1(n)$ and *(ii)* $M(\kappa_1) = M$. Let $K = K(\mathcal{N}) \in \mathbb{N}$ be the number of reachable states of $\mathcal{N}$. Then for all $n > K$, there is at least one marking M such that $|S_1(M,n)| \geq 2$; repeating the argument, one finds using (16) that for all $n > K^2$ there exists a marking M such that $|S_1(M,n)| > K$. With

$$U_2(M,n) \quad \triangleq \quad \left\{ \begin{array}{l} \kappa_2 \in \mathcal{L}, \\ \kappa_2 \sqsubseteq \kappa_2(n) \end{array} \;\middle|\; \begin{array}{r} \exists \kappa_1 \in S_1(M,n) : \\ \kappa_1 \sim_O \kappa_2 \end{array} \right\},$$

we thus have $|U_2(M,n)| > K$. This in turn implies that there exist $\kappa_2, \kappa_2' \in U_2(M,n)$ such that $\kappa_2 \neq \kappa_2'$ and $\kappa_2 \sim_M \kappa_2'$. By definition of $U_2(M,n)$, $\kappa_1 \sim_O \kappa_2$ and $\kappa_1' \sim_O \kappa_2'$. Since, by construction,

$$\kappa_1 \sqsubseteq \kappa_1' \sqsubseteq \kappa_1(n) \ \textit{and}\ M(\kappa_1) = M(\kappa_1') = M \tag{17}$$

property (11) is violated, q.e.d. □

6 Checking Diagnosability

We give two criteria for effective verification of diagnosability. The first is a linear algebra technique from the theory of Petri net invariants, which yields a quick but only necessary criterion; the second follows precisely the unfolding approach, yielding necessary and sufficient conditions for diagnosability based on a finite prefix whose size has off-line bounds.

6.1 A Necessary Condition

We follow the terminology and notation of [4].

Definition 13. *For a net $N = (\mathcal{P}, \mathcal{T}, \rightarrow)$, the* **incidence matrix** $\mathbf{N} : (\mathcal{P} \times \mathcal{T}) \rightarrow \{-1, 0, 1\}$ *is given by*

$$\mathbf{N}(p,t) \quad \triangleq \quad \left\{ \begin{array}{rcl} 0 & : & (p \rightarrow t \rightarrow p) \vee \neg(p \rightarrow t \vee t \rightarrow p) \\ 1 & : & (p \rightarrow t) \wedge \neg(t \rightarrow p) \\ -1 & : & (t \rightarrow p) \wedge \neg(p \rightarrow t) \end{array} \right. .$$

For a sequence $\sigma \in \mathcal{T}^*$ of transitions, the **Parikh vector** $\overline{\sigma} : \mathcal{T} \rightarrow \mathbb{N}$ *is given by $\overline{\sigma}(t) \triangleq |\sigma|_t$, i.e. the number of occurrences of transition t in σ.*

The action of transitions of N can be described by $\mathbf{N}$:

Lemma 3. (Marking Equation Lemma, Lemma 2.12 in [4]) *For $\sigma \in \mathcal{T}^*$ and markings M, M' of N such that $M \xrightarrow{\sigma} M'$, one has the following* **Marking Equation***:*

$$M' \quad = \quad M + \mathbf{N} \tag{18}$$

Note that $\mathbf{N}$ is independent of the marking, i.e. describes a *net* $(\mathcal{P}, \mathcal{T}, \rightarrow)$ rather than a particular *Petri net*.

Definition 14. *Let $N = (\mathcal{P}, \mathcal{T}, \rightarrow)$ be a net. A* **$\mathcal{T}$-Invariant** *of N is rational-valued solution of the equation $\mathbf{N} \cdot x = 0$. Equivalently ([4], Proposition 2.36), a mapping $J : \mathcal{T} \rightarrow \mathbb{Q}$ is a $\mathcal{T}$-Invariant of N iff for all $p \in \mathcal{P}$,*

$$\sum_{t \in {}^\bullet p} J(t) \quad = \quad \sum_{t \in p^\bullet} . \tag{19}$$

The importance of T-invariants lies in the following property:

Theorem 2. ([4], Proposition 2.37) *Suppose M is a marking of N and $\sigma \in \mathcal{T}^*$ such that $M \xrightarrow{\sigma}$. Then $\overline{\sigma}$ is a $\mathcal{T}$-invariant of N iff it reproduces M, i.e. $M \xrightarrow{\sigma} M$.*

See also [14] for an example of the use of T-invariants in analysis of systems. From Theorem 2, we thus know that $\kappa \sim_M \kappa'$ holds iff the "Parikh vector" $\overline{(\kappa', \kappa)}$ given by

$$\overline{(\kappa', \kappa)}(t) \quad \triangleq \quad |\{e \in \kappa' \backslash \kappa \mid \phi(e) = t\}|,$$

satisfies Equation (19). In fact, any linearization σ of the events in $\kappa' \backslash \kappa$ has same parikh vector, and so the above results apply simultaneously to any such σ. Therefore, Equation (19) can be used to check whether a given marking can *possibly* be reproduced in an unobservable way: in that case, Equation (19) must have a semi-positive solution (i.e. with all entries non-negative and at least one positive entry). As a consequence, we have:

Lemma 4. *If for all semi-positive solutions $v \in \mathbb{N}^{\mathcal{T}}$ of (19) , there exists $t \in O$ such that $v(t) > 0$, then $\mathcal{OBS}$ and $\mathbb{D}$ hold.*

However, this yields only a necessary condition since $\overline{\sigma} \in \mathbb{N}^{\mathcal{T}}$ may satisfy Equation (19) without corresponding to any firing sequence enabled in M; the solutions of (19) are candidates for cycles.

Examples In the net from Figure 3, the T-invariants are (with coordinates ordered by alphabetic order on $\{\alpha, \beta, \gamma, \delta\}$) $(1, 1, 0, 0)$, $(0, 0, 1, 1)$ and their linear combinations. For the

net in Figure 2, again with alphabetic ordering, the incidence matrix is

$$\mathbf{N} = \begin{pmatrix} -1 & -1 & 1 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} ;$$

the $\mathcal{T}$-invariants are $(0,1,1,0,0,0)$, $(1,0,1,1,0,1)$, and their linear combinations. In both examples, all T-invariants are firable; as we indicated above, this need not be true for any marking, and indeed, taking away one or more tokens in either example will disable invariants, as is easily checked.

6.2 Complete Prefix

If the use of invariants gives a quick, linear-algebra-founded criterion for diagnosability, effective verification needs a richer data structure. A 1-safe net has only finitely many reachable markings ; in fact, its reachability graph can be seen as a finite automaton. ONe can thus return to an automaton model and follow the literature in this framework; however, there is a more adequate representation of the behaviours of Petri net, used successfully in Model Checking, see below. Observe that all infinite runs of the unfolding will repeatedly pass through states that have already been visited before; conversely, there exist finite prefixes of the unfolding that contain already all information about the possible behaviours of the net. This is what allows using branching processes in Model Checking [6, 7, 25]; the different ways of obtaining and optimizing the *complete prefix* have received considerable attention in the literature, see [21] for a comprehensive treatment.

To the best of our knowledge, the problem of diagnosis is not treated in the literature on complete prefixes; the results obtained there do not carry over immediately. However, it is natural to expect such complete finite prefixes for deciding diagnosability to exist; the following existence theorem confirms that intuition.

Theorem 3. *For a given net $N = (\mathcal{P}, \mathcal{T}, \rightarrow)$, there exists a finite number $Z = Z(\mathcal{N})$ such that for any 1-safe marking $M_0 \subseteq \mathcal{P}$ of N, the Z-th prefix $\mathcal{R}_Z$ of the unfolding of $\mathcal{N} = (N, M_0)$ is sufficient to verify diagnosability: if there exist any $\kappa_1, \kappa_1', \kappa_2, \kappa_2'$ such that (11) is violated, one can choose them with this property such that* $\max(\mathcal{H}(\kappa_1'), \mathcal{H}(\kappa_2')) \leqslant Z$.

Proof: We begin with some preparations. Call an alarm pattern $\mathcal{A}$ *reducible* iff for all $\kappa \in \mathbf{diag}(\mathcal{A})$, there exist $\kappa_1, \kappa_2, \kappa_3$ such that

1. $\kappa = \kappa_1 \circ \kappa_2 \circ \kappa_3$,
2. $\mathcal{H}(\kappa_2) > 0$,
3. $\kappa_1 \circ \kappa_2^+ \circ \kappa_3 \subseteq \mathcal{L}$,

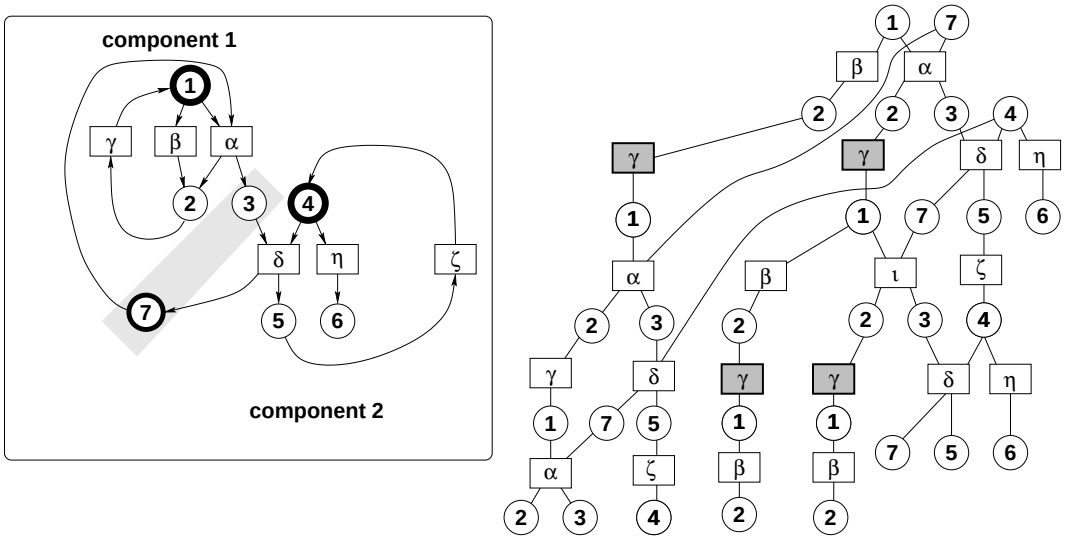


Figure 6: *A prefix (right) for the neton the left, compare Fig.2*

and *irreducible* otherwise. In the above definition, κ_1, κ_2 and κ_3 belong to different Petri nets, given by the net N with markings M_0, $M(\kappa_1)$ and $M(\kappa_2)$, respectively. Now, since $m \leqslant 2^{|\mathcal{P}|}$, we are done once the following claim is proved: *The number J of irreducible alarm patterns of $\mathcal{N}$ is bounded above by $m!$, where m denotes the number of reachable markings of $\mathcal{N}$.* To prove this, note that the height $\mathcal{H}(\kappa)$ of any configuration $\kappa \in \mathcal{L}$ that does not repeat a marking, i.e. such that $\kappa_1 \sqsubseteq \kappa_2 \sqsubseteq \kappa$ and $\kappa_1 \sim_M \kappa_2$ imply $\kappa_1 = \kappa_2$, is bounded above by m. Hence, all alarm patterns $\mathcal{A}$ whose height exceeds m are reducible, since $\kappa \in \mathbf{diag}(\mathcal{A})$ implies $\mathcal{H}(\kappa) \geqslant \mathcal{H}(\mathcal{A})$; the number of patterns of height m or less is obviously bounded above by $m!$.

□

Examples Consider Figure 6. Let $\Phi \triangleq \{\beta, \eta\}$; hence $O \subseteq \{\alpha, \gamma, \delta, \zeta\}$. We ask under which choices of O the net $\mathcal{N}$ satisfies $\mathcal{OBS}$, and if so, whether $\mathcal{N}$ is then diagnosable for that O. First, we claim that $\mathcal{OBS}$ (and even $\mathcal{WOBS}$) is equivalent with $\gamma \in O$. In fact, every infinite configuration of $\mathcal{N}$ contains γ-labeled events, so the implications

$$(\gamma \in O) \;\Rightarrow\; \mathcal{OBS} \;\Rightarrow\; \mathcal{WOBS}$$

are immediate. On the other hand, suppose $\gamma \notin O$; then the run ω formed by one occurrence of η and infinitely many alternating occurrences of β and γ contains no observable event, and ω constitutes a violation of both $\mathcal{OBS}$ and $\mathcal{WOBS}$. Now, let us check whether $O = \{\gamma\}$

makes $\mathcal{N}$ diagnosable. Suppose configuration κ contains an η-event. Inspection of Figure 6 shows that any extension κ' of κ such that either $\mathcal{H}(\kappa') > \mathcal{H}(\kappa) + 1$ or $\underline{\mathcal{H}}(\kappa') > \underline{\mathcal{H}}(\kappa) + 2$ contains some instance of γ. For the other fault label, β, one has that the conjunction of (i) $\phi^{-1}(\beta) \cap \kappa \neq \emptyset$ and (ii) $\mathcal{H}(\kappa') > \mathcal{H}(\kappa)+1$ or $\underline{\mathcal{H}}(\kappa') > \underline{\mathcal{H}}(\kappa)+1$, implies $\phi-1(\gamma) \cap \kappa \neq \emptyset$. Thus we conclude that $\gamma \in O$ is necessary and sufficient for $\mathcal{OBS}$, $\mathcal{WOBS}$, $\mathbb{D}$, and $\mathbb{W}$. Moreover, the prefix shown on the right hand side of Figure 6 is sufficient to decide observability and diagnosability; this illustrates that the worst-case bounds on sufficient prefixes given in the proof of Theorem 3 can be far off, due to effects of concurrency. For the net in Figure 3, we have different results depending on whether weak or strong responsability is sought: the net is

- *weakly* diagnosable iff $|O| \geqslant 1$, and
- *strongly* diagnosable iff $(O \cap \{\alpha, \beta\} \neq \emptyset) \wedge (O \cap \{\gamma, \delta\} \neq \emptyset)$.

Efficiency Fig. 6 shows that the upper bounds on the size of the complete prefix are far from sharp; in fact, the size of sufficient prefixes will be moderate as long as there is a high degree of parallelism in $\mathcal{N}$ and no excessive branching (unfoldings can do nothing to reduce branching but reduce the state space by exploiting concurrency). In many situations, exhibiting a very high degree of parallelism combined with a moderate degree of branching behavior, Γ can be considerably smaller than the reachability graph of $\mathcal{N}$, that is, the automaton representation of $\mathcal{N}$; in particular, the more parallelism there is in the application, the more is gained from the partial order representation. The computational complexity of the unfolding approach will thus compare favorably to the polynomial complexity of diagnosability verification shown in [36].

Weak vs strong The necessity of discerning weak and strong diagnosability arises from the distributed nature of large networks, and more generally in physically large systems with global time and state not observable. While strong diagnosability is the natural translation of classical diagnosability from the automata to the partial order framework, weak diagnosability is often more easily guaranteed. It should be noted that systems that weakly but not strongly diagnosable arise often as composition of simpler systems, whose components are , or can be made, strongly diagnosable; in the example of Figure 4, the two cyclic parts form such components. This motivates the study of truly *distributed* diagnosis, where each supervisor is given a limited domain of supervision, and communication among diagnosers is necessary to establish (local views of) global diagnosis; a work in progress, see [10].

Acknowledgments I am gratefully indebted to A. Benveniste, E. Fabre, and C. Jard for introducing me to diagnosis, and for countless enriching debates.

References

[1] A. Benveniste, E. Fabre, S. Haar. Markov nets: probabilistic models for distributed and concurrent systems. *IEEE Transactions on Automatic Control* **48**(11):1936–1950, November 2003.

[2] A.T. Bouloutas, S. Calo, and A. Finkel. Alarm correlation and fault identification in communication networks. *IEEE Trans. on Communication* **42**(2-4), 1994.

[3] C. Cassandras and S. Lafortune. *Introduction to discrete event systems.* Kluwer Academic Publishers, 1999.

[4] J. Desel and J. Esparza. *Free Choice Petri Nets.* Cambridge University Press, 1995.

[5] J. Engelfriet. *Branching Processes of Petri Nets.* Acta Informatica **28**:575–591, 1991.

[6] J. Esparza. Model Checking Using Net Unfoldings. *Science of Computer Programming* **23**:151–195, 1994.

[7] J. Esparza, S. Römer, and W. Vogler. An improvement of McMillan's unfolding algorithm. *Formal Methods in System Design* **20**(3):285-310, 2002.

[8] E. Fabre, A. Benveniste, C. Jard, L. Ricker, and M. Smith. Distributed state reconstruction for discrete event systems. *Proceedings CDC'2000.*

[9] E. Fabre, A. Benveniste, C. Jard, and S. Haar. Diagnosis of Asynchronous Discrete Event Systems, a Net Unfolding Approach. *IEEE Trans. Aut. Control***48**(5)714–727, May 2003.

[10] E. Fabre, A. Benveniste, C. Jard, and S. Haar. Distributed monitoring of a concurrent and asynchronous systems. *Proceedings CONCUR 2003*, LNCS **2761**, Springer Verlag 2003.

[11] E. Fabre, A. Benveniste, and C. Jard. Distributed diagnosis for large discrete event dynamic systems. *IFAC Cong.* 2002.

[12] E. Fabre. Convergence of Turbo Algorithms for Systems Defined by Local Constraints. INRIA Report 4860, 2003; URL: http://www.inria.fr/rrrt/rr-4860.html

[13] E. Fabre. Factorization of Unfoldings for Distributed Tile Systems. Part 1 : Reduced Interaction case. INRIA Report 4829, 2003; URL: http://www.inria.fr/rrrt/rr-4829.html

[14] B. Gaujal, S. Haar, and J. Mairesse. Blocking a Transition in a Free Choice Net, and what it tells about its throughput. *Journal of Computer and System Sciences* **66**(3):515–548, 2003.

[15] S. Haar, A. Benveniste, E. Fabre, and C. Jard. Partial Order Diagnosability of Discrete Event Systems Using Petri Net Unfoldings. In: *Proceedings of 42nd IEEE Conference on Decision and Control (CDC),* 2003.

[16] A. Giua. Petri net state estimators based on event observation. *Proc. CDC 1997.*

[17] A. Giua and C. Seatzu. Observability of Place/Transition Nets. *IEEE Trans. Aut. Control* **47**(9):1424–1437, 2002.

[18] S. Haar. Probabilistic Cluster Unfoldings. *Fundamenta Informaticae.* **53**(3-4):281–314, 2002.

[19] C.N. Hadjicostis, and G.C. Verghese. Monitoring discrete event systems using Petri net embeddings. in *Proc. 20st (ICATPN)*, LNCS **1639**:188–208, Springer Verlag 1999.

[20] I. Katsela, A.T. Bouloutas, S. Calo. Centralized vs distributed fault localisation. *Integrated Network Management IV,* A.S. Sethi, Y. Raynaud, F. Faure-Vincent (eds.), 251-261. Chapman and Hall 1995.

[21] V. Khomenko, M. Koutny, and W. Vogler. Canonical Prefixes of Petri Net Unfoldings. *Acta Informatica* **40**:95–118, 2003. Preliminary Version in: D. Brinskma and K.G. Larsen (eds.), *Proc. CAV 2002*, LNCS **2404**:582–595, Springer Verlag 2002.

[22] K.X. He and M.D. Lemmon. Liveness verification of discrete-event systems modeled by n-safe Petri nets. *Proc. 21st ICATPN* 2000, LNCS **1825**:227–243, Springer Verlag.

[23] K.X. He and M.D. Lemmon. On the existence of liveness-enforcing supervisory policies of discrete-event systems modeled by n-safe Petri nets. *Proc. IFAC'2000 Conf. on Cont. Syst. Design,* special session on Petri nets.

[24] MAGDA project. See URL: http://magda.elibel.tm.fr

[25] K. McMillan. Using Unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. *4th Workshop on Computer Aided Verification* 164–174, 1992.

[26] M. Nielsen, G. Plotkin, and G. Winskel. Petri nets, event structures, and domains, Part I. *Theor. CS* **13**:85–108, 1981.

[27] W. Reisig. *Petri nets.* Springer Verlag, 1985.

[28] W. Reisig. *Elements of Distributed Algorithms. Modelling and Analysis with Petri Nets.* Springer Verlag, 1998.

[29] A. Sahraoui, H. Atabakhche, M. Courvoisier, and R. Valette. Joining Petri nets and knowledge-based systems for monitoring purposes. *Proc. IEEE Int. Conf. on Robotics Automation,* 1160–1165, 1987.

[30] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *IEEE Trans. Aut. Control* 40(9), 1555-1575, 1995.

[31] R. Sengupta. Diagnosis and communications in distributed systems. *Proceedings* WODES *1998*, 144-151.

[32] S. Tripakis. Undecidable problems of decentralized observation and control. *Proceedings CDC 2001.*

[33] F. Vaandrager. A simple definition for parallel composition of prime event structures. *CWI report* **CS-R8903**, Amsterdam, March 1989.

[34] G. Winskel. Event structures. *Advances in Petri nets,* LNCS **255**: 325–392, Springer Verlag, 1987.

[35] G. Winskel. Categories of Models for Concurrency. *Seminar on Concurrency, Carnegie Mellon University July 1984.* LNCS **197**: 246–267, Springer Verlag, 1985.

[36] T. Yoo and S. Lafortune. Polynomial-Time Verification of Diagnosability of Partially-Observed Discrete-Event Systems. *IEEE Trans. Aut. Control* **47**(9):1491-1495 , 2002.

[37] T. Yoo and S. Lafortune. NP-completeness of Sensor Selection Problems Arising in Partially-Observed Discrete-Event Systems. *IEEE Trans. Aut. Control* **47**(9):1495–1499, 2002.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399