Abstraction-based failure diagnosis for discrete event systems

Klaus Schmidt*

Chair of Automatic Control, University of Erlangen-Nuremberg, Cauerstrasse 7, 91058 Erlangen, Germany

ARTICLE INFO

Article history:
Received 11 May 2009
Received in revised form
8 September 2009
Accepted 7 November 2009
Available online 22 November 2009

Keywords:
Discrete event systems
Diagnosability
Abstraction

1. Introduction

Failure diagnosis addresses the problem of identifying and isolating deviations of the actual behavior of a dynamic system from its nominal (desired) behavior. In recent years, various approaches that are based on a discrete event systems (DES) modeling formalism were developed. It is a basic premise for the practicability of failure diagnosis that each fault can indeed be uniquely identified based on the partial observation of the actual DES behavior and the characterization of the possibly faulty behavior. In this respect, failure events [1,2] or language specifications [3–5] are used to represent incorrect system behavior, and polynomial time algorithms were developed to solve the associated event diagnosability [6,7,2] or language-diagnosability [3–5] problems, respectively.

A common shortcoming of these approaches is that they involve the enumeration of the overall system state space which makes their application to large-scale systems computationally infeasible. Hence, it is of immediate practical interest to develop diagnosability methods that exploit the system structure in order to avoid the explicit representation of the overall system. Existing approaches that tackle this problem either rely on sufficient conditions that cannot be easily verified [8], or require specific models such as hierarchical finite state machines [9]. In this paper, we make use of vertical and horizontal system structure of DES that consist of multiple subsystems to reduce the computational effort for the verification of language-diagnosability.

We proceed as follows. In Section 3.1, we define a diagnosability problem for an abstraction of the original system model on a smaller state space. Section 3.2 gives sufficient conditions such that the solution of this abstraction-based diagnosability problem implies the solution of the original diagnosability problem, and in Section 3.3, we identify a case where the reverse implication also holds. An efficient method for the computation of the abstracted model without composing the original subsystems is provided in Section 4.

2. Preliminaries

2.1. Basic notation

For a finite alphabet Σ, the set of all finite strings over Σ is denoted Σ*. The empty string is denoted ∈ ∈ Σ*. For any string s ∈ Σ*, |s| denotes the length of s. A language over Σ is a subset L ⊆ Σ*. A language L is prefix-closed if L = L := {s1 ∈ Σ* | ∃s ∈ L s.t. s1 ≤ s}.

The natural projection p : Σ* → ˆΣ*, ˆΣ ⊆ Σ is defined iteratively: (1) let p(ε) := 0; (2) for s ∈ Σ*, σ ∈ Σ, let p(σs) := p(s) if σ ∈ ˆΣ, or p(σs) := p(s) otherwise. The inverse of p is p−1 : ˆΣ* → Σ*, p−1(t) := {s ∈ Σ*|p(s) = t}.

We model a DES by a finite automaton G = (X, Σ, δ, x0) with the states X, the alphabet Σ, the partial transition function δ : X × Σ → X and the initial state x0. We define the closed language L(G) of G and the synchronous composition G1 || G2 of two automata G1 and G2 in the usual way [10].

2.2. Language-diagnosability

As in [3,5], we consider a partially observed DES G = (X, Σ, δ, x0), where the system behavior is seen through a mask M : Σ → ∆ ∪ {ε} that maps each event σ ∈ Σ to its observation M(σ) ∈ ∆ ∪ {ε}. Here, ∆ is the set of observations, and we denote Σo := {σ ∈ Σ|M(σ) ≠ ε} as the set of observable events. M can be recursively extended to strings by defining M(σs) = M(s)M(σ) for s ∈ Σ* and σ ∈ Σ.
We represent a failure by the violation of a given prefix-closed specification language $K = K \subseteq L(G)$. Hence, it is desired to detect by partial observation through the mask $M$ if a faulty string in $L(G) - K$ occurred. The following definition of language-diagnosability as used in [3,5] formalizes this goal.

**Definition 2.1 (Language-Diagnosability).** Let $G$ model a DES and let $K = K \subseteq L(G)$ be a prefix-closed specification language. $K$ is language-diagnosable for $G$ and the observation mask $M : \Sigma \rightarrow \Delta \cup \{\epsilon\}$ if

$$\exists n \in \mathbb{N} \forall s \in L(G) - K \forall s_t \in L(G), |t| \geq n \text{ or st deadlocks} \Rightarrow (\forall u \in M^{-1}M(st) \cap L(G), u \notin K).$$

The smallest $n$ that satisfies (1) is denoted as the worst-case detection delay.

If (1) holds, then every string that deviates from the correct behavior in $K$ can be uniquely distinguished from strings in $K$ after a finite detection delay, i.e., the occurrence of a bounded number of events. It is shown in [5] that language-diagnosability can be verified in polynomial time based on $G$ and an automaton $C$ with $L(C) = K$. If $G$ has $p_C$ states and $q_C$ events, and $C$ has $p_C$ states, then the complexity for this verification is $O(p_C \cdot q_C^2 \cdot p_C^2)$.

### 3. Model abstractions for language-diagnosability

The verification of language-diagnosability addressed in the previous section depends on the explicit enumeration of the state space of the automaton $G$. Hence, a direct application of this method to systems of industrial size is computationally infeasible.

The aim of this section is to develop an approach that enables the language-diagnosability verification of large-scale DES.

#### 3.1. Problem statement

Our considerations are based on the model $G$ and the observation mask $M$ as introduced above. However, different from [5], we assume the practical case$^1$ where the specification $K \subseteq \Sigma^*$ is not given explicitly but rather evaluated using a reduced specification $K' \subseteq \Sigma'^*$ with $\Sigma' \subseteq \Sigma$ such that

$$K = K' || L(G) \subseteq L(G).$$

Instead of verifying language-diagnosability based on $G$, $K$, and $M$ as discussed in Section 2.2, we propose to use an abstracted model $\hat{G}$ over an abstraction alphabet $\hat{\Sigma} \subseteq \Sigma$. Considering that $K' \subseteq \Sigma'^*$, we also require that $\Sigma' \subseteq \hat{\Sigma}$ in order to capture the relevant behavior specified by $K'$. Then, we compute $\hat{G}$ by applying the natural projection $p : \Sigma^* \rightarrow \hat{\Sigma}^*$, and use the abstracted specification $\hat{K} \subseteq \hat{\Sigma}^*$ such that

$$L(\hat{G}) := p(L(G)),$$

$$\hat{K} := K' || L(\hat{G}) = p(K).$$

In addition, the abstracted observation mask is $\hat{M} : \hat{\Sigma} \rightarrow \hat{\Delta} \cup \{\epsilon\}$, where $\hat{\Delta} = \{M(\sigma) | \sigma \in \hat{\Sigma}\}$ contains all possible observations of events in $\hat{\Sigma}$ such that, for all $\sigma \in \hat{\Sigma}$, $M(\sigma) = M(\sigma)$. A graphical illustration of the abstraction methodology in (3) and (4) is provided in Fig. 1, with $\hat{K}$ and $\hat{L}(\hat{G})$.

Using the abstracted entities $\hat{G}$, $\hat{K}$, $\hat{M}$, we study the following problem.

**Problem 1 (Abstraction-Based Diagnosability).** Let $G$ be a model automaton, $K' \subseteq \Sigma'^*$ be a reduced specification and $M : \Sigma \rightarrow \Delta \cup \{\epsilon\}$ be an observation mask. Defining $\hat{G}$, $\hat{K}$ and $M$ as above for the abstraction alphabet $\hat{\Sigma}$ with $\Sigma' \subseteq \hat{\Sigma} \subseteq \Sigma$, we want to find sufficient conditions such that

1. language-diagnosability of $\hat{K}$ for $\hat{G}$ and $M$ implies language-diagnosability of $K := K' || L(G)$ for $G$ and $M$.

2. the abstracted model $\hat{G}$ has a smaller state space than the model $G$.

If condition 1 in Problem 1 holds, it is possible to solve the language-diagnosability problem by applying the algorithm in [5] to $G$, $\hat{K}$ and $M$. Denoting $p_C$ and $q_C$ as the number of states and events of $\hat{G}$, respectively, and $p_C$ as the state size of the automaton $C$, $L(C) = K$, the associated computational complexity is $O(p_C \cdot q_C^2 \cdot p_C^2)$. Then, condition 2 implies that $p_C$ is smaller than $p_C$ and $q_C$ is smaller than $q_C$. Furthermore, using $\hat{K} = K' || L(\hat{K})$ suggests that also $p_C$ is smaller than $p_C$. Together, it is expected that the computational effort for the evaluation of language-diagnosability for $\hat{G}$, $\hat{K}$, and $M$ can be considerably reduced compared to the verification for $G$, $K$, and $M$. The application example in Section 4.3 supports this claim.

**Remark 3.1.** Note that the abstraction using $p : \Sigma^* \rightarrow \hat{\Sigma}^*$ does not ensure that $\hat{G}$ is smaller than the original model $G$, in the worst case, the evaluation of $p$ can lead to an exponential increase in the size of $\hat{G}$ compared to $G$ [11].

### 3.2. Sufficient condition for abstraction-based diagnosability

We first present three counterexamples that lead to a violation of condition 1 in the problem statement. From these counterexamples, we deduce a sufficient condition for the natural projection $p$ that ensures that condition 1 in Problem 1 is satisfied.

Then, we show that this sufficient condition entails the fulfillment of condition 2 in the problem formulation.

We consider $G$ in Fig. 2 over $\Sigma = \{a, b, c, d, e, f, g, h\}$ and a reduced specification $K' = \{a, b\}$ over $\Sigma' = \{a, b, c\}$, i.e., the specification is violated if $b$ occurs. The automaton $C$ generates the associated specification $K' = K' || L(G)$ for the model $G$. Furthermore, we assume that the observation mask $M$ is described by $M(a) = M(b) = M(c) = M(d) = M(g) = M(b) = \epsilon$ and $M(e) = e, M(f) = \epsilon$. Inspecting the failure string $s = bb \in L(G) - K$, it is readily observed that $bb$ deadlocks for $t = c$ but, e.g., $u := ag \in M^{-1}M(st) \cap L(G)$ and $u \notin K$. Hence, with **Definition 2.1**, language-diagnosability of $K$ for $G$ and $M$ is violated. Next, we investigate the abstractions of $G$ and $K$ that are obtained with (3) and (4) using the abstraction alphabet $\hat{\Sigma} = \{a, b, e, f\} \supseteq \Sigma'$. It turns out that the abstracted specification $\hat{K}$ is language-diagnosable for the abstracted model $\hat{G}$ and the abstracted observation mask $\hat{M} = \hat{M}(a) = \hat{M}(b) = \epsilon, \hat{M}(e) = e,$
Definition 3.1 (Observer [12]). Let $L = \mathbb{I} \subseteq \Sigma^*$ be a prefix-closed language. The projection $p : \hat{\Sigma}^* \to \Sigma^*$ is an observer if for all $s \in L$, $t \in \hat{\Sigma}^*$,

$$p(s) t \in p(L) \Rightarrow \exists u \in \Sigma^* \text{ s.t. } su \in L \text{ and } p(su) = p(s)t.$$  

In words, the observer property requires that if the projection $p(s)$ of a string $s \in L$ can be extended by a string $t$ in $p(L)$, then there must be a corresponding string $u$ that projects to $t$ and extends $s$ in $L$.

In the next example, we investigate the situation in Fig. 3. $G$ is defined over $\Sigma = \{a, b, c, d, e, f\}$ and the reduced specification is $K' = \{e, a\}$ over $\Sigma' = \{a, b\}$ such that $G$ generates the specification language $K = K' \parallel L(G)$. $M$ is given such that $M(a) = M(b) = M(c) = M(d) = e$ and $M(e) = M(f) = \epsilon$. Considering the faulty string $b \in L(G) - K$, it holds that an arbitrarily long string can occur before the failure can be distinguished from the correct behavior due to the loop with $c$ and $d$ between the states 3 and 5. Thus, language-diagnosability is violated. On the other hand, using $\hat{\Sigma} = \{a, b, c, e, f\} \supseteq \Sigma'$, it can be verified that language-diagnosability holds for $\hat{\Sigma}$, $\hat{G}$, and $\hat{M}$. In this case, condition 1 in Problem 1 is not satisfied since the projection $p$ erases the local loop with the events $c, d \notin \hat{\Sigma}$.

In order to address the problem identified in the previous example, we introduce a stronger version of the observer property in Definition 3.1. In addition, the natural projection $p$ must not erase any loop of events in $\Sigma - \hat{\Sigma}$.

Definition 3.2 (Loop-preserving Observer). $p$ in Definition 3.1 is a loop-preserving observer for $L$ with the bound $N$ if for all $u \in (S)$, $|u| < N|t|$.

That is, a loop-preserving observer ensures that any loops in the original model $G$ also appear as loops in the abstracted model $\hat{G}$.

In the final example, we study the case where different events generate the same observation via the mask $M$. Fig. 4 shows $G$ and $\hat{G}$ with $L(G) = K' \parallel L(G)$ for the reduced specification $K' = \{e, a\}$ over $\Sigma' = \{a, b\}$. In addition, $M$ fulfills $M(a) = M(b) = e$, $M(c) = M(d) = \epsilon$, and $M(e) = M(f) = \epsilon$. Then it holds for any extension of the faulty strings $b, bd, bdf \in L(G) - K$ that they are indistinguishable from the correct strings $a, ac$ or $ace$. This implies that $K$ is not language-diagnosable for $G$ and $M$.

However, choosing $\hat{\Sigma} = \{a, b, c, e, f\} \supseteq \Sigma'$, $\hat{K}$ is language-diagnosable for $\hat{G}$ and $\hat{M}$ which again violates condition 1 in Problem 1. In this example, the problem is that although one event (c) with the observation $m_1$ is kept in $\hat{\Sigma}$, another event (d) with the same observation is projected away. Since this contradicts the semantics of the observation mask ($c$ and $d$ cannot be distinguished according to $M$), we require that the choice of the abstraction alphabet $\hat{\Sigma}$ is consistent with the observation mask $M$ in the sense that

$$\sigma \in \hat{\Sigma} \cap \Sigma_o \Rightarrow M^{-1}M(\sigma) \subseteq \hat{\Sigma}.$$  

We are now ready to state a sufficient condition that solves Problem 1.

Theorem 3.1 (Abstraction-based Diagnosability). Problem 1 is solved if $p$ is a loop-preserving observer and $\hat{\Sigma}$ is consistent with $M$.

Proof. We first assume that $\hat{K}$ is language-diagnosable for $\hat{G}$ and $\hat{M}$ with the worst-case detection delay $\hat{n}$, and show that $K$ is language-diagnosable for $G$ and $M$ by contradiction. Hence, we assume that $K$ is not language-diagnosable for $G$ and $M$. Then, w.l.o.g., there exists $s \in L(G) - K$ with $st \in L(G)$ such that $|t| > n := N - \hat{n}$. Consider $p(st)$ deadlocks in $G$, but there exists $u \in M^{-1}M(st) \cap L(G)$ s.t. $u \in K$. Then, it holds that $\hat{s} := p(s) \in L(\hat{G}) - K$ and $\hat{t} := p(t) \in L(\hat{G})$. We now investigate the cases (i) and (ii). Here, $\hat{p} : \hat{A}^* \to A^*$ denotes the natural projection from observations over $\Delta$ to observations over $\Delta$.

In case (i), Definition 3.2 implies that $|\hat{t}| > \hat{n}$ since $p$ is a loop-preserving observer with bound $N$. Furthermore, $u \in p^{-1}Mp(u) \subseteq p^{-1}M^{-1}Mp(u)$, and with consistency of $\hat{\Sigma}$ for $M$, $p^{-1}M^{-1}Mp(u) = p^{-1}M^{-1}pM(st) = p^{-1}M^{-1}Mp(st) = p^{-1}M^{-1}M(\hat{s})$. Hence, $\hat{u} \in M^{-1}M(\hat{s})$. Together, this shows that $\hat{s} \in L(G) - K, \hat{t} \in L(\hat{G}), |\hat{t}| > \hat{n}$ but there is $\hat{u} \in M^{-1}M(\hat{s}) \cap L(G)$ s.t. $\hat{u} \in K$, i.e., $\hat{K}$ is not diagnosable for $G$ and $M$.

In case (ii), $st$ deadlocks in $G$. If $|t| > \hat{n}$, the discussion for case (i) shows that the assumption that $K$ is not diagnosable for $G$ and $M$ leads to contradiction. Otherwise, $p$ is a loop-preserving observer, also $\hat{s}t$ deadlocks in $\hat{G}$. That is, we have $\hat{s} \in L(\hat{G}) - \hat{K}, \hat{t} \in L(G), \hat{s}\hat{t}$ deadlocks in $G$ but there is $\hat{u} \in M^{-1}M(\hat{s}) \cap L(G)$ s.t. $\hat{u} \in K$.

Hence, in both cases, diagnosability of $\hat{K}$ for $\hat{G}$ and $\hat{M}$ is contradicted.

To address condition 2 in Problem 1, we note that it is shown in [11] [Theorem 3.1.1] that the abstraction $\hat{G}$ cannot have a larger state space than the original model $G$ if the projection $p$ is an observer.$^2$

Remark 3.2. Note that the application of the above theorem relies on finding a subset $\hat{\Sigma} \subseteq \Sigma$ such that the projection $p$ is a loop-preserving observer. In the scope of this paper, we briefly describe an iterative approach to determining $\hat{\Sigma}$ based on an initial abstraction alphabet $\hat{\Sigma}_{init}$.$^3$ We first suggest to use the observer extension algorithm in [14] (complexity $O(p^2 \cdot q^2)$) in order to find a projection $\hat{p} : \Sigma^* \to \hat{\Sigma}^*, \hat{\Sigma}_{init} \subseteq \hat{\Sigma}$ that fulfills the observer property in Definition 3.1. As a result, the state space of the model $G$ is partitioned into equivalence classes such that each equivalence class corresponds to a unique state in the abstracted model $\hat{G}$. Now, it holds that $\hat{p}$ is a loop-preserving observer, if the subautomata of $G$ that correspond to the different equivalence

--

$^2$ In practical examples, a considerable reduction is reported [13].

$^3$ Since $\Sigma' \supseteq \hat{\Sigma}$ for $K' \subseteq \Sigma'$, a valid choice for the initial alphabet is $\hat{\Sigma}_{init} = \Sigma'$. 
Problem 1

Theorem 3.1 is beneficial if the abstraction-based language-diagnosability holds. However, if this verification fails, it cannot be concluded whether language-diagnosability for the original system is fulfilled or not. In this section, we identify a case such that both verifications are equivalent.

We study $G$ in Fig. 5 with the reduced specification $\mathcal{K}' = \{\epsilon, a, a\}$. $M$ is defined by $M(a) = M(b) = \epsilon$ and $M(c) = c$. $M(d) = d$, $M(e) = e$. Then, it can be observed that $K = K' \parallel L(G)$ is language-diagnosable for $G$ and $M$. Now assume that the abstraction $\hat{\Sigma} = \{a, b, e\} \supseteq \Sigma'$ is chosen. In this case, all possible extensions of the failure strings $b$, $c$, and $d$ cannot be distinguished from the correct strings $a$, $e$, i.e., $\hat{K} = K' \parallel L(\hat{G})$ is not language-diagnosable for $G$ and $\hat{M}$. Here, language-diagnosability for the abstraction fails since the observable events $c$ and $d$, that allow us to distinguish failure strings from correct strings, are not included in $\hat{\Sigma}$.

In accordance with the above example, the next theorem shows that the reverse implication of Problem 1, condition 1 holds if all observables in $\Sigma_o$ belong to $\hat{\Sigma}$, i.e., all possible observations are retained in the abstraction.

Theorem 3.2 (Equivalence). Consider the situation in Theorem 3.1. If $\Sigma_o \subseteq \hat{\Sigma}$, it holds that $\hat{K}$ is language-diagnosable for $\hat{G}$ and $\hat{M}$ iff $\hat{K}$ is language-diagnosable for $G$ and $M$.

Proof. “$\Rightarrow$”. This implication holds because of Theorem 3.1.

“$\Leftarrow$”. We have that $\Sigma_o \subseteq \hat{\Sigma}$ and $K$ is diagnosable for $G$ and $M$ with the worst-case detection delay $n$. We show that $\hat{K}$ is diagnosable for $\hat{G}$ and $\hat{M}$ by contradiction. W.l.o.g., we assume that, for $\hat{n} = n$, there is $s \in L(\hat{G}) \setminus K$ and $s\hat{t} \in L(G)$ s.t. $|s| > \hat{n}$ or (ii) $s\hat{t}$ deadlocks in $\hat{G}$ but there is $\hat{u} \in \hat{M}^-\hat{M}(s\hat{t}) \cap L(\hat{G})$ s.t. $\hat{u} \notin K$.

In case (i), there is $s \in p^{-1}(s) \cap L(G)$ and $t \in p^{-1}(t)$ s.t. $st \in L(G)$. Since $s \in L(\hat{G}) \setminus K$, also $s \in L(G) \setminus K$. Furthermore, since $\hat{n} = n$, $|t| > n$. Considering that $\hat{u} \in \hat{M}^-\hat{M}(s\hat{t}) \cap L(\hat{G})$ and $\hat{u} \notin K$, there is $u \in p^{-1}(\hat{u}) \cap L(G)$ and $u \in K' \parallel L(G) \setminus K = L(G)$.

Now, $\Sigma_o \subseteq \hat{\Sigma}$ implies that also $u \in p^{-1}(\hat{u}) \subseteq p^{-1}M^-M\hat{M}(\hat{t}) = p^{-1}M^-M\hat{M}\hat{p}(st) = M^-M\hat{M}\hat{p}(st)$. (here, we use the fact that $\Sigma_o \subseteq \hat{\Sigma}$ implies $M(s) = M\hat{p}(s)$ for all $s \in \Sigma'$). Hence, we found $s \in L(G) \setminus K$, st $u \in L(G)$, $|t| > n$ and $u \in M^-M\hat{M}\hat{p}(st) \cap L(G)$ s.t. $u \in K$ which contradicts that $K$ is diagnosable for $G$ and $M$.

In case (ii), since $\hat{p}$ is a loop-preserving observer, there is $s \in p^{-1}(s) \cap L(G)$ and $t \in p^{-1}(t)$ s.t. $st \in L(G)$ deadlocks in $G$ (if no such $s$ exists, $\hat{t} \hat{p}$ cannot deadlock in $\hat{G}$). If $|t| > n$, the same argument as in case (i) leads to contradiction. Otherwise, we have $s \in L(G) \setminus K$, $st$ deadlocks in $G$ but analogous to case (i) we can find $u \in M^-M\hat{M}\hat{p}(st) \cap L(G)$ s.t. $u \in K$ which contradicts that $K$ is diagnosable for $G$ and $M$. □

4. Language-diagnosability for composed systems

The approach presented in the previous section allows us to verify language-diagnosability based on the abstracted model $\hat{G}$ which is expected to result in computational savings. However, the construction of $\hat{G}$ still requires the enumeration of the state space of the original model $G$. In this section, the practical situation with system models that are composed of multiple subsystems is considered. It is shown that $\hat{G}$ can be efficiently computed using abstractions of the subsystem models.

4.1. Model abstractions for composed systems

We assume that the system model $G$ is composed of several subsystems $G_i = (X_i, \Sigma_i, \delta_i, \delta_{ob_i}, i = 1, \ldots, m$ such that $G := \biguplus_{i=1}^m G_i$ over the alphabet $\Sigma := \bigcup_{i=1}^m \Sigma_i$ (see the lower part of Fig. 6). In addition, a reduced specification $K' \subseteq \Sigma'^n$ describes the correct system behavior, and partial observation is possible via the observation mask $M : \Sigma \rightarrow \Delta \cup \{\epsilon\}$.

In order to exploit the composed structure of the model, we first compute abstractions $\hat{G}_i$ of the subsystems $G_i$ using abstraction alphabets $\hat{\Sigma}_i \subseteq \Sigma_i$ with $\Sigma_i \subseteq \Sigma' \subseteq \hat{\Sigma}_i$ and the natural projections $p_i : \hat{\Sigma}_i \rightarrow \Sigma_i, i = 1, \ldots, m$ such that $L(\hat{G}_i) = p_i(L(G_i))$. Then, the abstracted subsystems are composed to obtain $\hat{G} = \biguplus_{i=1}^m \hat{G}_i$ over the alphabet $\hat{\Sigma} := \bigcup_{i=1}^m \hat{\Sigma}_i$ as illustrated in the upper part of Fig. 6.

Our main goal is again the solution of Problem 1.

4.2. Conditions for language-diagnosability

We consider the general case where subsystems are allowed to share events, i.e., it is possible that $\Sigma_i \cap \Sigma_j \neq \emptyset$ for $i, j = 1, \ldots, m, i \neq j$. The set of shared events is $\Sigma_{i,j} := \bigcup_{i 
eq j} (\Sigma_i \cap \Sigma_j)$ for each subsystem $G_i$.

The following theorem states sufficient conditions that reduce the solution of Problem 1 with the abstracted model $\hat{G}$ according to (7) to the results obtained in Sections 3.2 and 3.3.

Theorem 4.1 (Composed Systems). Let $G_i, p_i, i = 1, \ldots, m,$ and $p, \hat{\Sigma}$ be defined as in Section 4.1. Problem 1 is solved if $1. \Sigma_i \subseteq \hat{\Sigma}_i$ for $i = 1, \ldots, m$. 2. $p_i$ is a loop-preserving observer for all $\hat{\Sigma}_i$. 3. $\hat{\Sigma} = \bigcup_{i=1}^m \hat{\Sigma}_i$ is consistent with $M$. Furthermore, equivalence holds if $\Sigma_o \subseteq \hat{\Sigma}$.

Conditions 1 and 2 in Theorem 4.1 ensure that only computations on the subsystems have to be carried out. Hence, instead of the overall model $G$, only the abstracted model $\hat{G}$ on a potentially smaller state space has to be constructed. Consequently, both the
verification of abstraction-based language-diagnosability in Section 3 and the proposed abstraction method for composed systems in Section 4.1 result in computational savings.

The proof of Theorem 4.1 relies on the following lemmas. Lemma 4.1 is adopted from [10] [Exercise 3.3.7], while Lemma 4.2 constitutes a new result.

**Lemma 4.1.** Let \( \Sigma_i, p_i, \Sigma_i, i = 1, \ldots, m \) and \( p \) be defined as above. Furthermore, assume that \( L_i \subseteq \Sigma_i \) and \( \Sigma_i \cap \Sigma_i = \emptyset \) for \( i = 1, \ldots, m \). Then, it holds that \( \| p \|_{i=1}^m L_i = \| p \|_{i=1}^m p_i(L_i) \). In particular, this implies for \( G_i, \hat{G}_i \) and \( G \) as above that \( p(L_i) = p(\| p \|_{i=1}^m L_i) = \| p \|_{i=1}^m p_i(L_i) = \| p \|_{i=1}^m L_i) \).

**Lemma 4.2 (Loop-Preserving Observer).** Let \( G, p, i = 1, \ldots, m, G \), p be defined as above. Then \( p \) is a loop-preserving observer for \( G \) with the bound \( N := \sum_{i=1}^m N_i \) if \( p_i \) is a loop-preserving observer for \( G_i \) with the bound \( N_i \) for \( i = 1, \ldots, m \).

**Proof.** Assume that \( p_i \) is a loop-preserving observer for \( G_i \) for \( i = 1, \ldots, m \) and let \( s \in L(G), t \in \Sigma^* \) s.t. \( p(s) \in p(L(G)) \). It has to be shown that there is \( u \in \Sigma^* \) s.t. \( su \in L(G) \) and \( p(su) = p(s)t \), and that for all such \( u, |u| < N[t] \).

We define the natural projections \( \theta_i : \Sigma^* \rightarrow \Sigma^* \) with \( \theta_i(u) = u_i \in \Sigma_i^* \) s.t. \( u_i \in L(G_i) \) and \( p_i(s)u_i = p_i(s)t_i \). Then, according to Lemma 4.1, \( \| p \|_{i=1}^m \theta_i(u) = \| p \|_{i=1}^m p_i(u_i) = \| p \|_{i=1}^m t_i = \sum_{i=1}^m \theta_i^{-1}(t_i) \neq \emptyset \). In particular, since \( t \in \Sigma^* \) t_i, there must be \( u \in \| p \|_{i=1}^m u_i \) s.t. \( p(u) = t \). Observing that \( \| p \|_{i=1}^m u_i \subseteq \| p \|_{i=1}^m s_i u_i \) it also holds that \( su \in L(G) \). It remains to show that for all such \( u, |u| < N[t] \). By assumption, we know that for all \( i, |u| < N_i[t] \). Furthermore, \( u \in \| p \|_{i=1}^m u_i \) implies that \( |u| \leq \sum_{i=1}^m |u_i| \). Hence, \( |u| < \sum_{i=1}^m N_i[t] \leq \sum_{i=1}^m N_i = N[t] \). □

Based on the above lemmas, Theorem 4.1 can be proved.

**Proof.** We first show sufficiency by verifying that the conditions in Theorem 3.1 are fulfilled. Because of Lemma 4.1, the abstracted plant in (7) generates the same language as \( G \) in (3), i.e., \( \| p \|_{i=1}^m p_i(L(G_i)) = p(L(G)) \). Furthermore, Lemma 4.2 implies that \( p \) is a loop-preserving observer, and \( \hat{S} \) is consistent with \( M \) by assumption.

Finally, with \( \Sigma_i \subseteq \Sigma \), equivalence directly follows from Theorem 3.2. □

**Remark 4.1.** Note that, in the case of composed systems as in the above theorem, the initial alphabet for computing loop-preserving observers as described in Remark 3.2 is given by \( \Sigma_i \cap (\Sigma_i \cap \Sigma') \) for each \( i = 1, \ldots, m \).

4.3. Application example

We study a small manufacturing unit that is part of a laboratory machine at the Chair of Automatic Control, University of Erlangen-Nuremberg. It consists of a stack feeder (SF) and a conveyor belt (C1) as depicted in Fig. 7. The SF comprises a tower that can hold wooden parts and a belt that can move parts until they reach the neighboring conveyor belt C1, which is described by the unobservable event pass. A light barrier detects if parts arrive at or leave the belt of SF which is modeled by the events sf\_a and sf\_f, respectively. In addition, the belt of the SF can start and stop moving (events sf\_mv and sf\_a). Its motion is initiated by the event sf\_cl that is shared with C1. The desired behavior of SF according to a supervisor design in [13] is given by the subautomaton of \( G_{SF} \) in Fig. 8 that consists of the states with a white background. Similarly, the desired behavior of C1 is characterized by \( G_{C1} \). After the transport of a part is initiated by sf\_cl, C1 starts to move (c1\_mv) and the part reaches C1 after some time (pass). As soon as the part arrives at the sensor of C1 (c1\_a), C1 stops (c1\_a) and becomes ready for a new transport whenever the part is removed from C1 (c1\_l).

One possible failure occurs if a part gets stuck between SF and C1. In SF, we characterize this failure by the unobservable event stuck that can occur after the part has left the sensor (af\_1) and before it reaches C1 via pass (shaded states of \( G_{SF} \) in Fig. 8). In C1, the failure occurrence is modeled by a timer that elapses if the part does not arrive on time (timer in \( G_{C1} \)).

We define the reduced specification \( K' = \{ \epsilon \} \) over the alphabet \( \Sigma' = \{ \text{stuck}, \text{timer} \} \) to capture that stuck and timer should not occur. Furthermore, the observation mask is given by \( M(\text{stuck}) = M(\text{pass}) = \epsilon \), while all remaining events can be directly observed.

In the next step, we choose the abstraction alphabets \( \hat{S}_{SF} = \{ \text{af\_1, pass, stuck} \} \cap \hat{S}_{C1} = \{ \text{af\_1, pass, stuck} \} \cap \hat{S}_{C1} = \{ \text{af\_1, pass, stuck} \} \), both natural projections \( p_{SF} \) and \( p_{C1} \) are loop-preserving observers as can be seen by the respective abstractions \( \hat{G}_{SF} \) and \( \hat{G}_{C1} \) in Fig. 8. Noting that also \( M \) is consistent for \( \hat{S} = \hat{S}_{SF} \cup \hat{S}_{C1} \), all conditions in Theorem 4.1 are fulfilled. Hence, it is sufficient to verify language-diagnosability based on \( \hat{G} = \hat{G}_{SF} \parallel \hat{G}_{C1} \) and \( \hat{K} = \hat{K}' \parallel \hat{L} = \hat{L} \) as shown in Fig. 9. Since each failure string in \( L(G) = \hat{L} \) can be uniquely distinguished from correct strings in \( \hat{K} \), \( K \) is language-diagnosable for \( G \) and \( \hat{M} \), which implies language-diagnosability for the original system with \( G = G_{SF} \parallel G_{C1} \). Hence, \( M \) is consistent for \( \hat{S} = \hat{S}_{SF} \cup \hat{S}_{C1} \), all conditions in Theorem 4.1 are fulfilled. Hence, it is sufficient to verify language-diagnosability for the original system with \( G = G_{SF} \parallel G_{C1} \). Because of \( M(\text{pass}) = \epsilon \), while all remaining events can be directly observed.

5. Conclusion

In this paper, the idea of abstraction-based language-diagnosability was introduced in order to avoid the enumeration of the overall system state space for the diagnosability verification of discrete event systems. To this end, a version of the observer condition that is originally used in the abstraction-based supervisory control was adopted to compute an abstracted system model on a smaller state space. Then, sufficient conditions for the verification of language-diagnosability using the abstracted model were developed, and it was proved that abstraction-based diagnosability and diagnosability for the original system are equivalent if all possible observations are retained in the abstracted model. Finally, the practical case of large-scale DES that are given in the form of multiple subsystem models was considered. It was shown that the
model abstraction can be applied to the subsystems instead of the overall system which can result in considerable computational savings. The benefits of the proposed method were illustrated by a manufacturing unit.

References


[10] W.M. Wonham, Supervisory control of discrete-event systems, Department of Electrical and Computer Engineering, University of Toronto, URL http://www.control.utoronto.ca/DES.


