Chapter 14
Diagnosis of Petri Nets

Maria Paola Cabasino, Alessandro Giua, and Carla Seatzu

14.1 Introduction

Failure detection and isolation in industrial systems is a subject that has received a lot of attention in the past few decades. A failure is defined to be any deviation of a system from its normal or intended behavior. Diagnosis is the process of detecting an abnormality in the system behavior and isolating the cause or the source of this abnormality.

As usual the first significant contributions in this framework have been presented in the area of time driven systems. However, a quite rich literature on diagnosis of discrete event systems has also been produced in the last two decades. Faults may be described by discrete events that represent abnormal conditions. As an example, in a telecommunication system, a fault may correspond to a message that is lost or not sent to the appropriate receiver. Similarly, in a transportation system, a fault may be a traffic light that does not switch from red to green according to the given schedule. In a manufacturing system, it may be the failure of a certain operation, e.g., a wrong assembly, or a part put in a wrong buffer, and so on.

In the first part of this chapter we recall an approach proposed in [16], that is a generalization of [11]. The main feature of the diagnosis approach here presented is the concept of basis marking. This concept allows us to represent the reachability space in a compact manner, i.e., it requires to enumerate only a subset of the reachability space. In particular, arbitrary labeled Petri nets (PNs) are considered where there is an association between sensors and observable events, while no sensor is available for certain activities — such as faults or other unobservable but regular transitions — due to budget constraints or technology limitations.

It is also assumed that several different transitions might share the same sensor in order to reduce cost or power consumption; in such a case if two transitions
are simultaneously enabled and one of them fires, it is impossible to distinguish which one has fired, thus they are called *undistinguishable*. Four diagnosis states are defined that model different degrees of alarm. The only limitation on the system is that the unobservable subnet should be acyclic. Such an assumption is common to all discrete event systems approaches, not only those based on PNs, but those based on finite state automata as well.

Note that the approach here presented, as most of the approaches dealing with diagnosis of discrete event systems [21, 28, 37, 39], assumes that the faulty behavior is completely known, thus a fault model is available. Such an assumption is applicable to interesting classes of problems, e.g., this is the case of many man-made systems where the set of possible faults is often predictable and finite in number [2, 24, 31, 44].

In the second part of this chapter, several extensions of this basic approach are presented. The first result originates from the requirement of relaxing the assumption of acyclicity of the unobservable subnet. To address this issue fluidification is proposed [7, 8, 14] (see Chapters 18–20 of this book for a comprehensive presentation of fluidification in Petri nets). Second, the diagnosability problem is considered: a system is said to be diagnosable if when a fault occurs, it is possible to detect its occurrence after a finite number of events occurrences. Obviously, this is a major requirement when performing online fault diagnosis. Finally, the problem of designing a decentralized diagnoser is discussed. Indeed, due to the ever-increasing complexity of nowadays systems and the intrinsic distributed nature of many realistic systems, performing online centralized diagnosis often reveals not convenient, or even unfeasible.

### 14.2 Basic Definitions and Notations

As already discussed in the Introduction, the goal of a diagnosis problem consists in reconstructing the occurrence of a fault event based on the observation of the output of some sensors. The association between sensors and transitions can be captured by a *labeling function* \( \mathcal{L} : T \rightarrow L \cup \{\varepsilon\} \) assigns to each transition \( t \in T \) either a symbol from a given alphabet \( L \) or the empty string \( \varepsilon \).

The set of transitions whose label is \( \varepsilon \) is denoted as \( T_u \), i.e., \( T_u = \{ t \in T \mid \mathcal{L}(t) = \varepsilon \} \). Transitions in \( T_u \) are called *unobservable* or *silent*. \( T_o \) denotes the set of transitions labeled with a symbol in \( L \). Transitions in \( T_o \) are called *observable* because when they fire their label can be observed. In this chapter, it is assumed that the same label \( l \in L \) can be associated with more than one transition. In particular, two transitions \( t_1, t_2 \in T_o \) are called *undistinguishable* if they share the same label, i.e., \( \mathcal{L}(t_1) = \mathcal{L}(t_2) \). The set of transitions sharing the same label \( l \) is denoted \( T_l \).

In the following, let \( C_u \) (\( C_o \)) be the restriction of the incidence matrix to \( T_u \) (\( T_o \)) and \( n_u \) and \( n_o \), respectively, be the cardinality of these two sets. Moreover, given a sequence \( \sigma \in T^* \), \( P_u(\sigma) \), resp., \( P_o(\sigma) \), denotes the projection of \( \sigma \) over \( T_u \), resp., \( T_o \).