Fault diagnosis for large Petri nets

George Jiroveanu

Promotor: prof. dr. ir. R.K. Boel
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen

Vakgroep Elektrische Energie, Systemen, en Automatisering
Voorzitter: Prof. dr. ir. J. Melkebeek
Faculteit Ingenieurswetenschappen
Academiejaar 2005–2006

Fault diagnosis for large Petri nets
George Jiroveanu

Supervisor:
prof. dr. ir. René K. Boel

Address:
Onderzoeksgroep SYSTeMS
Vakgroep Elektrische Energie, Systemen, en Automatisering (IR08)
Universiteit Gent
Technologiepark Zwijnaarde 914
B-9052 Zwijnaarde
België

This book was typeset using LaTeX.

ISBN 90-8578-047-0
NUR 992,919
Wettelijk depot D/2006/10.500/5

To my parents

Acknowledgments

The first person I would like to thank is my supervisor Prof. René K. Boel for his detailed and constructive comments, encouragements, and guidance in the development of my research work.

I also acknowledge the personnel of our department for creating a wonderful ambiance in our group. It was an honor and also a pleasure to be part of the SYSTeMS Research Group for more than 4 years. It is a pleasure to thank Prof. D. Aeyels, Prof. De Cooman, and Prof. Em. J.L. Willems, to our secretary Margot, and to my colleagues Matthias, Jonathan, Luc, Erik, Valery, Jana, Enrique, and Filip, for all these years together.

Many thanks go to my friends who, by a smile or a benevolent word, by endless discussions or by touristic trips, by sport activities or by saturday-night meetings at Carola's pub made these years in Gent wonderful. I mention Alexandru, Ionel, Andu, Krisztina, Dan, Iustina, Cosmin, Monica, Radu, Lucia, Patrick, Simona, Gunther, Cristina, Carmen, Julien, Luiza, Katrien, Miki, Gisela, Cecilia, Jose, Costel, Monica, Pr. Vasile, Siluan, Laurentiu, Ioana, Marius, Mihaela, Florin, Tudor, Gabi, Carola, Oana and many others whose names only by lack of memory I failed to include in this list.

I want to express my thanks to a few of wonderful scientists that helped me to enrich the content of my thesis by valuable comments and fruitful discussions, particularly Prof. J. van Schuppen, Prof. B. De Schutter, and Prof. B. Bordbar.

I also thank those who agreed to be the referees of this thesis and allocated their valuable time in order to evaluate the quality of this work: Prof. K. De Bosschere, Prof. J. van Schuppen, Prof. B. De Schutter, Prof. De Cooman, Prof. L. Vandevelde, and Prof. Em. J.L. Willems.

The financial support for this research is also acknowledged. This research was sponsored by The Belgian Found for Scientific Research (BOF), by the Belgian Program on Inter-university Poles of Attraction (IAP) initiated by the Belgium State, Prime Minister's Office for Science, Technology and Culture, and by a European Union Marie Curie Fellowship (during my stay at Delft University of Technology).

Finally, *Un Grand Merci* for my family for their love and support, as well as their understanding and patience during all these years.

Nederlandse samenvatting

Dit proefschift behandelt het ontwerpen van algoritmes voor foutdiagnose voor grote en complexe systemen die gemodelleerd worden als Petri-netten. De toestand van het systeem evolueert onder de invloed van gebeurtenissen. Een gebeurtenis in het systeem stemt overeen met de uitvoering van een transitie in het Petri-netmodel. De toestand van het Petri-netmodel wordt beschreven door het aantal tokens in de verschillende plaatsen in de netwerkgraaf. Een transitie kan uitgevoerd worden - en de corresponderende toestandsverandering is mogelijk - als de ingangsplaatsen van die transitie genoeg tokens bevatten.

De evolutie van het systeem wordt extern waargenomen door sensoren die het uitvoeren van sommige transities detecteren. Sommige van de niet-waarneembare transities worden beschouwd als fouten (b.v. kortsluitingen in een transmissienet). Bedoeling van dit proefschrift is om algoritmes te ontwerpen die de waargenomen rij transities gebruiken om te bepalen of er al dan niet fouten optraden (met zekerheid opgetreden, of misschien opgetreden).

Foutdiagnose gebeurt in twee stappen. Eerst wordt nagerekend welke rijen van opeenvolgende transities toegelaten zijn volgens het Petri-netmodel, voor een gekende begintoestand van het systeem, en zo dat de waargenomen transities overeenstemmen met de waarneembare transities in de gegenereerde rij. Dit kan bijvoorbeeld gerealiseerd worden door, vertrekkend uit de begintoestand van het systeem, alle mogelijke rijen transities te genereren, en die rijen te verwerpen welke niet overeenstemmen met de waargenomen transities. Nadien wordt nagekeken of alle toegelaten rijen, of sommige toegelaten rijen, transities bevatten die fouten voorstellen.

Dit probleem is computationeel moeilijk op te lossen via een gecentraliseerde, monolithische berekeningsmethode wegens de exponentiële groei van de toestandsruimte van een groot Petri-net. In dit proefschrift tonen we aan dat volledige enumeratie kan vermeden worden door enkel die rijen transities te genereren die zeker moeten uitgevoerd worden om de waargenomen rij gebeurtenissen te verklaren. Dit verkleint de verzameling van minimale verklarende rijen aanzienlijk, en we tonen aan dat die verzameling volstaat om die fouten te detecteren die, gegeven een rij waarnemingen, zeker plaats hadden.

Tevens tonen we aan dat het genereren van alle minimale verklarende rijen gebeurtenissen voor een waarneming ook mogelijk is door achterwaarts te redeneren. Vertrekkend van de waargenomen gebeurtenis zoekt men de voorwaarden die moeten vervuld zijn opdat die gebeurtenis kan uitgevoerd worden. Om aan die voorwaarden te voldoen moeten vroeger andere gebeurtenissen uitgevoerd geweest zijn, waartoe weer nieuwe voorwaarden moeten vervuld geweest zijn. Dit wordt verder gezet tot men de begintoestand bereikt. De complexiteit van deze berekeningen hangt enkel af van de grootte van een deel van het Petri-net dat enkel niet-waarneembare transities bevat.

Ten einde de computationele complexiteit verder te beperken bestuderen we in dit proefschrift gedistribueerde algoritmes. Het systeemmodel wordt opgesplitst in verschillende componenten. Elke component wordt voorgesteld door een Petri-net dat interageert met naburige componenten door het uitwisselen van tokens via gemeenschappelijke grensplaatsen. Elke component wordt bestuurd door een lokale agent die enkel lokale waarnemingen ontvangt. Het uitwisselen van tokens via grensplaatsen is niet direct waarneembaar. De agent kent enkel het lokale Petri-netmodel, de lokale begintoestand, en de naam van de naburige componenten waarmee tokens uitgewisseld worden.

Voorwaartse analyse is dan niet mogelijk omdat de lokale begintoestand niet gekend is, maar achterwaartse analyse kan wel. Wanneer de lokale agent voorwaarden berekent die afhangen van het ontvangen van tokens uit naburige componenten, dan wordt dit als een grensvoorwaarde genoteerd door de lokale agent. Lokale agenten kunnen dus lokale verzamelingen minimale verklaringen genereren via achterwaartse analyse. De bijhorende foutdiagnose geldt dan onder expliciet gekende onderstellingen op het uitwisselen van tokens. In dit proefschrift worden voorwaarden gegeven waaronder de lokale foutdiagnose alle zeker opgetreden fouten detecteert, maar eventueel ook meer fouten dan een centrale diagnose-agent zou vinden. Een centrale diagnose-agent is een agent die alle waarnemingen van alle componenten zou ontvangen, en die alle modellen van alle componenten, en alle begintoestanden zou kennen.

Om de verzameling zeker opgetreden fouten in een lokale agent te reduceren tot de verzameling fouten die door een centrale diagnose-agent zouden gedetecteerd worden moet er informatie uitgewisseld worden tussen agenten van naburige componenten. Deze informatie-uitwisseling moet nagaan of de randvoorwaarden van alle paren lokale verklaringen in naburige componenten compatibel zijn met elkaar. In dit proefschrift wordt een communicatie-algoritme ontwikkeld tussen naburige agenten die toelaat om die compatibiliteit in een eindig aantal stappen te bekomen (voor een systeem met slechts 2 componenten volstaan 3 boodschappen per agent).

In dit proefschrift wordt aangetoond dat onmiddellijk nadat een informatie-uitwisseling met naburige agenten heeft plaats gevonden volgens het beschreven protocol, elke lokale agent dezelfde lokale fouten als zeker opgetreden fouten detecteert als een globale agent zou detecteren. Dit geldt onafhankelijk van de keuze van het tijdstip van de informatie-uitwisseling. Informatie-uitwisseling kan dus geïnitieerd worden ofwel door een centrale superviserende regelaar, of door een willekeurige lokale agent.

In het tweede deel van dit proefschrift wordt foutdetectie besproken voor een systeemmodel dat ook gedetailleerde informatie kan bevatten over de tijdstippen waarop gebeurtenissen plaats vinden. Time Petri-netten modelleren ook de tijdsvertraging tussen het ogenblik waarop de voorwaarden voor het uitvoeren van een transitie voldaan zijn en het effectief uitvoeren van die transitie. We onderstellen dan dat het exacte tijdstip van uitvoering van waarneembare transities beschikbaar is voor de lokale agenten. Die bijkomende informatie maakt sommige verklarende rijen uitgevoerde transities onmogelijk omdat die niet aan het time Petri-netmodel voldoen.

Voor de time Petri-netmodellen lossen we dezelfde vragen op als voor modellen zonder tijd. De analyse is veel ingewikkelder omdat de toestand van een time Petri-net in principe niet aftelbaar is (vermits men het exacte tijdstip moet onthouden waarop vroegere gebeurtenissen plaats hadden). We tonen aan dat er methodes bestaan om een eindig aantal equivalentie klassen van toestanden te vinden die voldoende informatie bevatten om centrale en gedistribueerde foutdetectie te doen. Onder meer restrictieve voorwaarden op de lokale Petri-netmodellen bestaan er ook in dit geval protocollen voor informatie-uitwisseling zodanig dat een lokale agent dezelfde fouten als zeker opgetreden markeert als een centrale foutdetector. Tevens is de lokale foutdetectie vóór de informatie-uitwisseling heeft plaats gevonden steeds een overdiagnose.

English Summary

In this thesis we study the diagnosis of large and complex systems modeled as Petri Nets (PNs). The fault diagnosis problem is to detect the occurrence of a fault exploiting the model and the received observation. In the PN model the faults are explicitly modeled as unobservable transitions and the plant observation is considered via a subset of events whose occurrence is (always) reported. The plant diagnosis is obtained in two phases: first the plant behavior that obeys the received observation is derived (as a set of legal traces in the PN model) and then it is checked if some or all of the legal traces include fault transitions.

It is well known that for large PNs the monolithic calculations cannot be performed because of the state space explosion. This is because the set of complete explanations of the plant observation is computed. However we show that only a small subset (called the minimal explanations) of the set of complete explanations is sufficient to design a diagnoser that has the same performance in detecting the faults that for sure happened in the plant as the one based on the set of complete explanations.

The set of minimal explanations is calculated backwards starting from the observation and deriving traces that lead back to the initial marking. The computation complexity of the backward search is however not comparable with the computational complexity of the forward search since they explore different state spaces but has the advantage that it does not depend on the size of the PN model but only on the size of the largest sub-net in the model that includes only unobservable transitions. Moreover and very important the set of complete explanations can be calculated from the set of minimal explanations whenever this is required.

We consider then a distributed setting where the plant consists of different components, and associated with each component there is a local agent (diagnoser-agent). Each component is modeled as a Petri Net while the interactions between components are modeled by common border places that allow tokens from one component to enter/exit to a neighbouring component. As a novelty we consider the case of unobservable interactions between components. Thus tokens can enter/exit unobservably (silent) a component.

Each local agent only knows the model of the local component and of

its interactions with its neighbours and only receives signals from the monitoring system for the local events. The local agents are linked by communication channels that allow them to exchange limited information. The distributed plant calculation comprises two phases: local calculations and then inter-agent communication for consistency. We consider that the communication is not driven by local observations, and moreover we require each local agent to derive a preliminary calculation of its component such that the preliminary diagnosis result of its component can be used for taking mandatory isolation/control actions in absence of communication with the other agents.

The main difficulty in this general setting is to design a procedure for deriving the preliminary local calculations of a component. This is because the initial marking is only partially known since tokens can enter/exit the local component.

The solution that we propose is to use the concept of minimal explanations of the received observation. Thus using a backward search method each agent derives starting from its local observations a set of local minimal explanations. A minimal local explanation comprises the unobservable events in the component that must have happened before the observation together with the minimal number of tokens that must have entered the component.

We show that if the PN model of each component satisfies the condition that each oriented path that starts from an input place and leads to an output place of the component contains an observable transition then the preliminary local diagnosis provides an over-diagnosis of the component w.r.t. the fault events that would be detected by a centralized diagnoser that for sure happened in the local component. If this condition is not satisfied the local preliminary diagnosis can omit the detection of some fault transitions that are included on unobservable paths that link the input and output places of a component but this would generally mean that the plant decomposition is not appropriate.

When communication is allowed the agents first extend their minimal local explanations for deriving the tokens that could have been delivered to the neighbouring components. Then they exchange information about the marking of the border places (the tokens assumed to enter/exit the components) to check the consistency of their results. Local explanations are found consistent if the agents agree on the marking of the border places. The communication protocol that we design is proved to terminate recovering the centralized diagnosis result by consistent local diagnosis results.

Then we extend these results for the case when the models include timing information about the plant operation. For Time Petri Nets models we design diagnosis algorithms considering the exact observation of the time when the observable events are executed in the plant. The timing information allows for more accurate models and consequently more accurate diagnosis but the price to be paid is that analysis of timed models becomes very complicated. As a result, for a similar distributed setting as considered for untimed models, we need some simplifying assumptions in order to meet the requirement that the communication protocol terminates (recovering the centralized diagnosis result) and the requirement that the preliminary local calculations can be used for tacking control/isolation actions. This is because Time Petri Nets (TPNs) lack a monotonicity property w.r.t. the initial marking and thus the analysis of a TPN model of a component with an unknown marking of the border places becomes extremely complicated. The simplifications that we consider are that the models of the components are required to be free-choice TPN and each oriented path that starts in an input place of and leads to an output place of the component should include an observable event.

Contents

Chapter 1

Introduction

1.1 General introduction to fault diagnosis

Man made systems are becoming more and more complex thanks to rapid developments in different technological areas that allow for powerful computational devices and fast communication facilities. The growth in the complexity of the modern systems allows for high performances but the price that is paid is that their analysis becomes more difficult, especially for plants where society expects a very high reliability (e.g. power systems, transportation, health care).

In a general description a system can be viewed as comprising different components (sub-systems) that interact one with each other in order to achieve the goals for which the plant was designed. Thus the overall plant performance relies on the performance of each individual component as well as on the quality of the interaction between the components. No matter how simple or well designed a man made system is, it is subject to abnormal behavior. The abnormal behavior of a component can be understood as any deviation from its designed behavior.

The causes of such deviations can be very complex and not all the time known. The abnormal behavior of a component of a system is in many cases due to the occurrence of a fault within the component itself or due to abnormal behavior that is propagated via the interactions between the components. Thus a fault in the plant can be defined as an unexpected deterioration (malfunction) of one of its components as a result of execution of an undesirable event. The execution of such an undesirable event (e.g. a valve sticks, a short circuit) is called a fault.

To prevent the cascading propagation of the faults in the plant it is crucial to detect and diagnose the faults accurately and in the shortest time after their occurrence. Thus the diagnosis task is of crucial importance for every made man system. The diagnoser of a plant is required to answer the following questions: *"Did a fault happen or not?"*(fault detection), *"What kind of fault happened if any?"* (fault isolation) and *"How did the fault happen?"*(explanations).

In order to answer these questions one must have some knowledge about the plant that comprises knowledge on the plant description and operation as well as knowledge on the normal (designed) behavior, and of the possible faults that can occur.

The first approach to solve the diagnosis problem was to use the knowledge of human experts in order to derive a set of rules about the relationships between the faults and the symptoms in the plant under investigation. The main inconvenient of the *rule-based diagnosis* is that it cannot be practically applied to large and complex systems since the knowledge acquisition from human experts is difficult and time consuming. The set of rules grows exponentially with the plant size and it is hard to maintain. Moreover it cannot be reused for other systems, and it can not cope with changes of the plant structure (components that are plugged in or removed).

The natural way to overcome these limitations is to exploit an explicit model of the plant model for deriving the plant diagnosis [Rei87]. The advantages are that the plant model encodes the set of all the rules that govern the plant behaviour. Rules derived by human experts are derived from observing a plant that behaves like this model. The quality of model-based diagnosis relies on the accuracy of the model rather than on the subjective knowledge of the human experts. The model of the overall plant can be accurately built up from small models, each modeling a component of the plant and its interactions with its neighbouring components. The overall plant model is simply adjusted when its structure changes by adjusting the model of only those components that are changed instead of updating a huge database of rules.

Thus *the model-based diagnosis* exploits an explicit model of the plant and the plant observation in order to derive what faults have happened in the plant and to predict the future behaviour. We refer to [LZ03] for an extensive classification of the model-based diagnosis approaches.

1.2 Model-Based Diagnosis for Discrete Event Systems

In this thesis we study fault diagnosis for plants that can be modeled as a Discrete Event Systems (DES). A DES can be thought as a dynamic system whose state changes with an event occurrence (event driven) rather than as time elapses. A DES can be modeled in different ways, among others as a finite-state automaton or a Petri Net (PN). The model-based diagnosis of

DES has received a lot of consideration over the past few years being applied in various technological areas e.g. telecommunication networks [Pen00], [PCR01], [BFHJ03], [FBHJ05], power systems [YOYS92], [BL99], [HV00], [JB03], FMS and production systems [BSC02], [HA02], [MLRV02], [ZKW03].

Most commonly the faults are represented in a DES model as fault events in automaton, respectively fault transitions in PN model while the plant observation is represented by a subset of the set of events (transitions). The occurrence of an observable event is reported to the monitoring system. The subset of events that are not observable (the unobservable events) are considered executed in the plant silently, i.e. their occurrence is not reported to the monitoring systems. Obviously the set of fault events is included in the subset of unobservable events, otherwise the detection of faults would be trivial.

Among the two above mentioned methods to model a DES we chose in this thesis the Petri Net approach to represent the plant model. In a basic formulation the *model-based diagnosis* of Petri Net models can be formulated as follows:

Given the plant model as a Petri Net and given the observation generated by the plant up to the current time determine whether an unobservable transition that models a fault event has happened in the plant or not.

We consider an *a priori* and complete description of the plant and also we consider that the faults to which the plant is exposed are known in advance, and included in the model. Moreover the plant observation is assumed correct, i.e. the occurrence of an observable event is all the time reported (no loss of observation) without any delay. The assumption that there are no-delays in receiving the plant observation simply means that the observation includes the time of the execution of an observable transition and this is measured with perfect accuracy (according to a global clock).

The model-based diagnosis for DES models comprises two stages. First the set of traces that are legal from the initial marking and obey the received observation is derived and then the diagnosis result of the plant is obtained checking if some or all of the legal traces include fault transitions. If all the traces include fault transitions the fault is declared *to have happened for sure*; if none of the legal traces include a fault event the diagnosis result is *normal*; while if there are legal traces that include fault transitions as well as legal traces that do not include fault transitions the diagnosis result is *uncertain* [SSL$^+$95].

The model-based diagnosis of DES can be classified as follows:

1. Centralized approaches: There is one centralized diagnoser that derives the plant diagnosis based on its (complete) knowledge of the overall plant model and the overall plant observation. The centralized approach can be further classified as:

   (a) diagnoser approach [SSL$^+$95] where a diagnoser automaton is derived off line and the on-line plant analysis is carried out by eliminating the diagnoser-states that are not consistent with the plant observation.

   (b) active system approach [BL99] where the diagnosis result is derived *a posteriori* when the system is in a quiescent state (out of work or idle).

2. Decentralized approaches: There is one centralized agent receiving information from several local diagnosers, each of which performs some local diagnosis of the plant with incomplete knowledge (e.g. based on a sub-set of sensor readings or a partial knowledge of the overall plant model). The local diagnosis results are compiled in a consistent diagnosis result for the overall plant by the centralized diagnoser [DLT00], [Pen00], [PCR01], [BvS02], [DLT03].

3. Distributed approaches: The overall plant consists of different components, and associated with each component there is a local agent (diagnoser-agent) that derives the local diagnosis of its component. Each local agent only knows the model of the local component and of its interactions with its neighbours. Each local agent moreover only receives signals from the monitoring system for the local events. No centralized structure is assumed to coordinate the results of the local agents but from time to time the local agents may exchange messages over communication channels linking them. Thus the local agents derive the distributed plant diagnosis by local calculations and by information exchange [Pro02], [KKSW02], [BFHJ03], [GL03], [SW04], [BJ04], [FBHJ05], [GL05], [JB05b].

The main disadvantage of a centralized approach is its high computational complexity. It requires a centralized plant model and generates a centralized plant diagnoser. Since the diagnoser-automaton can be viewed as a special observer-automaton its size may become too large to be practically stored [OW90]. Even if a centralized diagnoser can be constructed it has the following disadvantages [Su04]:

1. weak robustness - a partial malfunction of the centralized diagnoser affects the overall plant diagnosis.

2. low maintainability - a change in the plant structure requires a complete re-calculation of a new centralized diagnoser. This may be serious problem when the plant structure is known to change often.

The decentralized approaches overcome the high complexity and the low maintainability limitations of the centralized approach by calculating local state spaces (of size a lot smaller than the size of the overall plant) that are maintained consistent by a centralized structure (agent). But the existence of a a centralized agent does not eliminate the disadvantage of a weak robustness.

The distributed approaches consider that there is no centralized structure that coordinates the results of the local diagnosers. The local results are checked for consistency by pairwise communication between the neighbouring agents. Thus the price that is to be paid for eliminating the disadvantages of the centralized approach is that an adequate communication protocol should be derived to guarantee that the centralized diagnosis result is recovered by consistent local diagnosis results. Moreover to design such a communication protocol requires some structural assumptions to be fulfilled by the distributed plant description e.g. the interaction between the local components is restricted to the structure of a so-called hyper-tree [BFHJ03]. However the distributed diagnoser will be robust against many failures in the agents and in the communication system provided the local diagnosis obtained by each agent provide an acceptable approximation of the optimal fault diagnosis.

To prevent the deterioration of the plant behavior and possibly catastrophic failures the diagnosis of a fault occurrence must be followed by immediate isolation actions. In an electric power system the diagnosis of a short circuit must be followed by opening of circuit breakers removing power supply to the short-circuited lines. Thus the diagnosis task should be viewed in a broader supervisory architecture that comprises also control and isolation modules.

1.3 Centralized diagnosis

The chief drawback of the centralized approaches is its high computational complexity due to state space explosion. This is because the size of a diagnoser automaton may be exponential in the number of places of the underlying plant automaton model [OW90]. For large systems this means that the diagnoser-automaton cannot be stored on a computer. This problem persists even if the plant diagnosis is obtained *a posteriori* [BL99] by deriving the part of the diagnoser-automaton that corresponds with the received observation.

As already mentioned the diagnosis result should be viewed as an input to control and isolation modules that clear up the effects of the faults. For a certain plant observation the diagnosis result may be either *sure* that a fault happened or *uncertain* whether a fault happened or *sure* that a fault did not happen, corresponding with the states in the diagnoser-automaton `F`, `UF`, and `N` respectively [SSL^{+}95]. Thus the diagnoser state `F` requires actions to be taken whereas the diagnoser state `N` does not require any action. The diagnoser state `UF` is delicate since the plant observation and the model indicate that it is possible that some fault has happened but this is not for sure.

In absence of any probabilistic information about the fault occurrences it is natural to assume that no control actions are taken unless the diagnoser is sure that a fault happened in the plant, except in cases where consequences of a fault could be catastrophic.

In the model-based fault diagnosis paradigm the faults are explicitly modeled by a subset of the unobservable events. Since the occurrence of a fault is unexpected it is natural to assume that there is no reachable state in the plant from where only fault transitions can be executed otherwise the fault occurrence would be predictable before its occurrence. The straightforward implication of this assumption is that the faults are represented as unobservable choice transitions. Moreover a fault can be detected for sure only if the occurrence of the fault has an observable manifestation in the future. Thus the fault transitions that can be detected that for sure happened are only those that are predecessors of the transitions that were observed to be executed in the plant.

It means that it is sufficient to derive the minimal explanations of the received observation in order to derive a diagnosis result that has the same quality in detecting the faults that for sure happened as the diagnosis result based on the complete explanations of the observation [JB04a]. For large plants the set of minimal explanations is a lot smaller than the set of complete explanations. Notice also that all the complete explanations can be calculated by extending the minimal explanations. Thus the complete diagnosis result including faults that happened for sure and faults that may have happened can be derived at any time if required.

The computation of the set of minimal explanations for PN models can be achieved by a backward calculation starting from the observed events. The backward calculation can be seen as a forward calculation in the reverse PN obtained by reversing the direction of the arcs in the original PN and modifying the enabling and firing rule of a transition.

Methods based on backward calculations for PNs were proposed in [LA94], [SJ94], [CKV95] for diagnosis purpose, in [NAH$^+$98], [AIN00], [FRSB02], [DRvB04] for model checking, and in [GS02], [GCS05] for state estimation of a PN model with uncertain initial marking.

The backward algorithm that we design for computing the set of minimal explanations is similar to the algorithms used in model checking for deciding whether a bad marking can be covered from a given initial state [AIN00], [DRvB04]. In our case the bad marking is represented by the marking of the input places of an observed event and we must calculate backwards all the traces that cover it unobservably. For diagnosis we need all the traces not just showing the existence of one explanatory trace. The computational complexity of the backward calculations is not easily comparable with the forward calculations since they explore different state spaces. However its efficiency can be increased using place invariants and other heuristics [FRSB02] to drive the backward search, as well as the backward unfolding technique [AIN00] to avoid the consideration of all the possible interleavings of the concurrent events. Notice that the backward calculations in our case involve only unobservable transitions. Thus the method is potentially very efficient for practical applications that do not comprise large sub-nets containing only unobservable transitions.

1.4 Distributed diagnosis

To overcome the limitations of the centralized approach this thesis presents an efficient distributed diagnosis algorithm using as plant description a collection of place-bordered PNs [Val94], [GL03], [BFHJ03], [BJ04], [FBHJ05], [GL05], [JB05a]. Each PN models a component (local site) of the overall plant while the border places model the interactions between the components.

The distributed diagnosis problem can be formulated as follows. First the local agents perform a preliminary local diagnosis, then they exchange information updating their preliminary calculations until the consistency is achieved. We require that at the time the agents achieve consistency of their local results, the agents recover the result that would have been derived by a centralized agent that knows the overall plant model and receives the whole plant observation.

A very useful property is also that a preliminary local diagnosis (calculated locally in absence of information exchange) is useful for control and isolation actions that are necessary after a fault occurrence. This ensures robustness against communication channel break downs, even when the (global) consistency of a local site result was not achieved yet. This problem is of practical importance for spatially distributed large systems with unreliable communication between sites and is related to the question of how the diagnosis result relates with some control/isolation actions that may be required to be taken in response to fault occurrences. Thus before being able to communicate, a local agent may receive a sequence of observed events and is required to have a local preliminary calculation that *explains* what was locally observed.

A difficult problem arises when no assumption is made on the observabiltiy of the border places (i.e. the observability of the input/output transitions of the border places). When the input/output transitions of the border places are unobservable the number of tokens in the input places is unknown and the problem we face is to analyze a PN model with an uncertain initial marking.

When *a priori* knowledge of the token traffic between two sites is assumed known the problem can be solved by considering for the preliminary calculations upper bounds (maximum number of tokens that could have entered a local site) that result in the preliminary calculation of an over-estimate of the local site behavior. Based on this overestimate each local agent computes an over-diagnosis of the local site. This may be useful for very conservative applications [SW04]. This method is a translation of the methods proposed in [BL99], [KKZ01], and [SW04] for the plant model given as a network of communicating automata. This translation is not straightforward. Structural assumptions must be satisfied, otherwise the calculation of the upper-bounds is not possible unless first generating the overall plant state space. This is exactly what we want to avoid.

We assume that the unobservable transitions are silent: tokens can move unobservably from one PN model to another. Thus we extend the distributed diagnosis methodology to the situation when a sensor failure is reported to the supervisor, the plant operation cannot be stopped, and the sensor is not repaired immediately. To avoid that local calculations would have the same magnitude as the global plant calculation [Val94] we have proposed in [BJ04] a backward search method that starts from the locally observed events and derives the minimum number of tokens required to have entered from the neighboring sites. In this way we derive the set of minimal explanations of the local observation.

After locally computing the set of minimal explanations of the local observation based on the minimal number of tokens required to have entered the border places, a local agent extends (forwardly) the minimal explanations for estimating the tokens that could have exited the PN model of its component.

Then the algorithm checks whether local preliminary traces from one component can be matched to consistent traces from another (neighbouring) component, or not. Preliminary traces that can not be matched are discarded. Some new traces may be also generated because by communication ″new things may be found to be possible″. This is because initially a minimum number of tokens was assumed to have entered while later it may be found that more tokens than this minimal number may have entered the border places of the PN model of the component. Since at each communication round new traces are generated, we need to show that the algorithm terminates by achieving a fix point when no new traces are generated by any agent. In Section 4.3 of this thesis we show that this can indeed be achieved. Moreover this thesis proves that the sets of local traces that were found consistent recovers the result of a centralized agent.

To increase the computational efficiency we use the unfolding technique for both forward [McM93], [Esp94] and backward [AIN00] calculations. Beside the advantage that a configuration in an unfolding compactly represents a family of traces (obtained by linearizing the partial order relation between the event nodes of the configuration) there is also the advantage that the partial order between the nodes induces the time information on the border-conditions (ordering times when tokens must have entered and tokens that could have exited).

The information exchanged between agent allows each local agent to check the consistency of its local results with the results of the neighboring agents. We show that if the plant description satisfies the condition that all the unobservable circuits in the overall plant PN model contain transitions of at most two components then the local agents can achieve the global consistency of their local results exchanging information regarding only their common border.

Moreover we analyze what the preliminary local diagnosis includes. We show that if each component satisfies the condition that each oriented path that starts in an input place and ends in an output place contains at least one observable event then the local preliminary diagnosis is an over-diagnosis of the centralized diagnosis of the component w.r.t. the detection of the faults, i.e. each fault that is detected to have happened for sure by a centralized diagnoser is also detected in the preliminary local diagnosis. If this condition is not satisfied then some fault transitions that are situated along the unobservable paths that start in input places and end in output places of the component may not be diagnosed properly.

1.5 The diagnosis of Time Petri Nets

Often a user has precise information about the time intervals for the execution of operations in the plant, while accurate measurements of the execution times of the observed events are possible in many cases (e.g. if a GPS system provides a global time for the plant).

PN models where the time is considered as a quantifiable and continuous parameter allow for a more accurate analysis of the plant. A formalism that is convenient for expressing temporal constraints regarding the execution and the duration of the events is represented by Time Petri Nets [Mer74].

The diagnosis of Time Petri Nets (TPN) has only recently received some consideration [GBT05], [CJ05] in the framework of [SSL$^+$95]. Both papers use a centralized approach and assume no global clock available. The plant observation in these papers does not include the exact time when an observable event is executed in the plant. These papers are directed at applications such as communication networks where the inherent time scale of the plant is so fast that no accurate time measurement is possible.

In this thesis we treat models that are inherently slower. Hence in this thesis it is assumed that accurate global clocks are available and that the plant diagnosis uses the exact time when the observable events occurred in the plant. This is motivated by our interest in the diagnosis of electrical power systems [BJ03b] where the availability of a global clock is very common [YOYS92], [MDHM04].

Since a transition in a TPN can fire at any time in some predefined interval, TPN models have in general infinite state spaces because a state may have an infinite number of successor states. Methods based on grouping states that are equivalent under a certain equivalence relation in to so called *state classes* were proposed in [BM83], [YR98], [BV02] where it was shown that for bounded PNs the state class graph is finite. Thus the potentially infinite state space of a TPN can be finitely represented and the analysis of TPN models is computable.

In order to represent more accurately the plant behavior we consider the plant analysis based on the atomic state class graph [YR98], [BV02]. The atomic state class graph is a refinement of the linear state class graph. It is based on the observation that a state in a linear state class may contain states that do not have successors in all the successor state classes.

The on-line monitoring algorithm for fault diagnosis works as follows. When the process describing the behaviour of the plant starts we derive paths in the atomic state class graph up to the first observable event. If either the observable event is not observed as executed in the plant, or if it is executed sooner than it is allowed by the plant model then the path is deleted. Otherwise an equality relation is added to the characteristic system to express the fact that the observable event occurred at the time given by the received observation. Adding equality relation destroys in general the atomicity property and thus to restore it one must refine the predecessor state classes.

A similar algorithm was proposed in [Vic01] for the timeliness analysis of time-dependent systems but considering the plant analysis based on the linear state class graph.

Since the TPN analysis based on state-classes becomes computationally unfeasible for models of reasonable size because of the state space explosion due to the interleaving of the unobservable concurrent events methods based on partial orders were proposed in [HB95], [SY96], [AL97], [YS97], [Lil98].

The diagnosis algorithm that we propose first considers a centralized plant analysis based on time-configurations (time-processes [AL97]). A time-configuration is an untimed configuration with a valuation of the execution time for its events. A time-configuration is valid if there is a time trace in the original TPN that can be obtained from a linearization of the events of the configuration where the occurrence time of the transitions in the trace are identical with the valuation of their images in the time-configuration. To check whether a time-configuration is valid or not requires to solve a $(max, +)$-linear system of inequalities called the characteristic system of the configuration (a $(max, +)$ algebra is like a standard $(+, \times)$-algebra but with maximization as first binary operation and addition as second binary operation).

Since the number of valid time-configurations is uncountable we introduce the concept of time-interval configurations to finitely represent the set of all possible valid time-configurations. The idea is simple. The set of all solutions of the characteristic system of a configuration (the set of all valid times) is represented as a cover of subsets of solutions such that each subset of solutions has a time independence property for the concurrent events in the configuration. The time independence property of a subset of solutions of the characteristic system of a configuration can be intuitively understood as follows: *given any set of concurrent events in the configuration and fixing the execution times of their predecessors, their executions times belong to a hyper-rectangle in high dimensional space*. The execution time-intervals for the events in the configuration are obtained from the smallest hyperbox (of dimension equal with number of events in the configuration) that includes a given subset of solutions of the characteristic system.

We present efficient algorithms to derive such a partition of the solution set of the characteristic system of a configuration and show how the method can handle the addition of extra inequalities (constraints) that are due to the received observation.

Then we extend the distributed diagnosis algorithm to the case of TPN models. The distributed setting is very similar with the one that we consider for the untimed models. However some simplifying assumptions are needed in order to prove important properties of the distributed fault diagnosers, namely the overall plant model is a free-choice net and on each component a path that leads from an input place to an output place contains an observable event. These conditions are required in order to meet the requirement that the preliminary local calculations can be used for taking some control/isolation actions.

As for the untimed PN models, the preliminary local calculations of the TPN model of a component give rise to a major difficulty namely the analysis of a model with uncertain initial conditions. We adapt the backward unfolding method to Time Petri Net models and show for the case of two components that the distributed algorithm that we propose recovers the diagnosis result of a centralized agent by consistent pairs of local diagnosis results.

Chapter 2

Mathematical background for Petri Net models

2.1 Sets, numbers, and relations

Let X and Y be sets. We write $X \subseteq Y$ if X is a subset of Y, including the case $X = Y$. $X \subset Y$ denotes that $X \subseteq Y$ and $X \neq Y$. $X \setminus Y$ denotes the set of elements of X that do not belong to Y. $\mid X \mid$ denotes the cardinality of X and $Pwr(X)$ is the power set of X, that is, the set of all subsets of X. Given $f : X \to Y$ and $A \subseteq X$ then $f(A) = \bigcup_{x \in A} f(x)$.

$\mathbb{N}$ denotes the set of natural numbers including 0. $\mathbb{N}_+$ denotes the set of natural numbers excepting 0. $\mathbb{Q}, \mathbb{Z}$ and $\mathbb{R}$ denote respectively the set of rational, integer, and real numbers.

Given two vectors A, B of dimension m we have that: $i)$ $A \leq B$ if for $q = 1, \ldots, m$, $A[q] \leq B[q]$ and $ii)$ $A < B$ if $A \leq B$ and $\exists q$ s.t. $A[q] < B[q]$.

Definition 1. *A set X is a collection of distinct elements. Then given a non-empty set X and a function $\mu : X \to \mathbb{N}$ we say that X_μ is multi set over X where:*

$$X_\mu = \{(x, \mu(x)) \mid x \in X\}$$

and μ represents the number of appearances of x in X_μ. Thus a set X can be understood as a multi-set that has no repeated elements.

Example 1. *Let $X = \{x, y, z, u, v\}$ be a set and then consider the function μ given as $\mu(x) = 3$, $\mu(y) = 2$, $\mu(z) = 4$, $\mu(u) = 0$, and $\mu(v) = 0$. Then we have the multi-set $X_\mu = \{x, x, x, y, y, z, z, z, z\}$ or in a shorter notation $X_\mu = \{(x, 3); (y, 2); (z, 4)\}$.*

Whenever clear from the context we drop the lower index μ of a multi-set X_μ, since a set is a multi set where $\mu(x) = 1$ for all $x \in X$.

A (binary) relation R on a non-empty set X is a subset of the cartesian product $X \times X$. We use the notation:

$id_X = \{(x,x) \mid x \in X\}$ is the identity relation

$R^{-1} = \{(y,x) \mid (x,y) \in R\}$ is the inverse of R

For $v \in \{1,2,3,\ldots\}$, R^v is inductively defined by $R^1 = R$ and for $v > 1$:

$$R^v = \big\{(x,z) \mid (x,y) \in R^{v-1} \text{ and } (y,z) \in R \text{ for some } y \in X\big\}$$

$R^+ = R^1 \cup R^2 \cup R^3 \cup \ldots$ is the transitive closure of R

$R^* = id_X \cup R^+$ is the reflexive and transitive closure of R

In the following for $(x,x') \in R$ we use also the notation xRx'.

A (binary) relation $R \subseteq X \times X$ is an equivalence relation if:

$$\begin{array}{lr}
(\forall x \in X) \;\; xRx & (R \text{ is reflexive}) \\
(\forall x,x' \in X) \;\; xRx' \;\Rightarrow\; x'Rx & (R \text{ is symmetric}) \\
(\forall x,x',x'' \in X) \;\; xRx' \;\wedge\; x'Rx'' \;\Rightarrow\; xRx'' & (R \text{ is transitive})
\end{array}$$

We have that $(R \cup R^{-1})^*$ is the least equivalence relation that includes R.

For $x \in X$ denote $[x]_R$ the equivalence class of R that includes x that is:

$$\text{for } x \in X,\ [x]_R = \{y \in X \mid (x,y) \in R\}$$

and then denote $X_{/R} = \{[x]_R \mid x \in X\}$ the set of equivalence classes of R.

Let $\preceq$ be binary relation on X. $\preceq \subseteq X \times X$ is a partial order relation on X if it is:

$$\begin{array}{lr}
(\forall x \in X) \;\; x \preceq x & (\preceq \text{ is reflexive}) \\
(\forall x,x',x'' \in X) \;\; x \preceq x' \;\wedge\; x' \preceq x'' \;\Rightarrow\; x \preceq x'' & (\preceq \text{ is transitive}) \\
(\forall x,x' \in X) \;\; x \preceq x' \;\wedge\; x' \preceq x \;\Rightarrow\; x = x' & (\preceq \text{ is antisymmetric})
\end{array}$$

If $\forall x,x' \in X$ either $x \preceq x'$ or $x' \preceq x$ then $\preceq$ is a total order on X.

In the following for a partial order relation $\preceq$ on an nonempty set X we use the notation $(X,\preceq)$. Then $\mathtt{max}_{\preceq}(X)$ and $\mathtt{min}_{\preceq}(X)$ denote the set of maximal respectively minimal elements of X w.r.t. $\preceq$:

$$\mathtt{max}_{\preceq}(X) = \{x \in X \mid (x' \in X \wedge x \preceq x') \Rightarrow x' = x\}$$

$$\mathtt{min}_{\preceq}(X) = \{x \in X \mid (x' \in X \wedge x' \preceq x) \Rightarrow x' = x\}$$

2.2 Sequences and languages

Let Σ be a finite set of symbols a,b, ... We refer to Σ as an alphabet. Let Σ^+ denote the set of all sequences, of the form $a_1a_2\ldots a_\upsilon$ where $\upsilon \geq 1$ is arbitrary and $a_\lambda \in \Sigma$ for $\lambda = 1, \ldots, \upsilon$. Denote then by ϵ the empty sequence (sequence with no symbols) where $\epsilon \notin \Sigma$.

Then for $\Sigma^* = \Sigma^+ \cup \{\epsilon\}$ we say that an element of Σ^* is a word or a string while ϵ is the empty string. We refer in the following to Σ^* as the *Kleene-closure* of the alphabet Σ .

A language $\mathcal{L}$ over a set of symbols Σ is a subset of Σ^*, i.e. an element of the power set $Pwr(\Sigma^*)$.

A language may be thought as a formal way of describing the behavior of a discrete event system (DES). It specifies all admissible sequences of events that are allowable (legal) under the DES model.

Given $u, v, w \in \Sigma^*$ and $uvw = s$ we say that:

- u is called a prefix of s
- v is called a substring of s
- w is called a suffix of s

The catenation of two strings u and v ($u, v \in \Sigma^*$) is the string s obtained from u followed by v, that is $s \in \Sigma^*$ and $s = uv$.

Let the following operations be defined for languages:

- *Concatenation:* For $\mathcal{L}', \mathcal{L}'' \subseteq \Sigma^*$:

$$\mathcal{L}'\mathcal{L}'' \triangleq \{s \in \Sigma^* : (s = s's'') \wedge (s' \in \mathcal{L}') \wedge (s'' \in \mathcal{L}'')\}$$

- *Prefix-closure:* For $\mathcal{L} \subseteq \Sigma^*$:

$$\overline{\mathcal{L}} \triangleq \{u \in \Sigma^* : \;\; \exists v \in \Sigma^* \;\;\; s.t. \;\;\; uv \in \mathcal{L}\}$$

($\overline{\mathcal{L}}$ is the language that contains all the prefixes of all the strings of $\mathcal{L}$)

- *Kleene-closure:* Let $\mathcal{L} \subseteq \Sigma^*$:

$$\mathcal{L}^* \triangleq \epsilon \cup \mathcal{L} \cup \mathcal{L}\mathcal{L} \cup \mathcal{L}\mathcal{L}\mathcal{L} \ldots$$

Consider a string $s \in \mathcal{L} \subset \Sigma^*$. Denote $\Sigma(s)$ the set of symbols of Σ that appear in s. Then for each symbol $a \in \Sigma(s)$ denote $\mu_s(a) \in \mathbb{N}_+$ the number of appearances of a in s. Denote by $\Sigma_\mu(s)$ the multi-set of symbols generated by s:

$$\Sigma_\mu(s) = \{(a, \mu_s(a)) : a \in \Sigma(s)\}$$

Given a language $\mathcal{L}$ ($\mathcal{L} \subseteq \Sigma^*$) define the equivalence relation $\equiv_{\Sigma_\mu}$ as follows:

$$\forall s, s' \in \mathcal{L}, \quad s \equiv_{\Sigma_\mu} s' \quad \text{if} \quad \Sigma_\mu(s) = \Sigma_\mu(s')$$

Denote the equivalence class that includes s:

$$[s]_{\equiv_{\Sigma_\mu}} = \{s' \in \mathcal{L} : \Sigma_\mu(s) = \Sigma_\mu(s')\}$$

and the quotient language $\mathcal{L}_{/\equiv_{\Sigma_\mu}} = \Big\{[s]_{\equiv_{\Sigma_\mu}} \mid s \in \mathcal{L}\Big\}$.

Example 2. *Given the set of symbols $\Sigma = \{a, b, c\}$, denote $\mathcal{L} = \{aab, acc, ccb\}$ and $\mathcal{L}' = \{aa, bb\}$ two languages defined over the set of symbols Σ. Then we have:*

$$\mathcal{L}\mathcal{L}' = \{aabaa, accaa, ccbaa, aabbb, accbb, ccbcc\}$$

$$\overline{\mathcal{L}} = \{\epsilon, a, c, aa, ac, cc, aab, acc, ccb\}$$

Given $s = accaa$ we have $\Sigma(s) = \{a, c\}$ and $\Sigma_\mu(s) = \{(a, 3); (c, 2)\}$. Then for $s' = ccaaa$ we have that $\Sigma_\mu(s) = \Sigma_\mu(s')$ that is $s \equiv_{\Sigma_\mu} s'$.

Definition 2. *Consider a partially ordered set $(\Sigma, \preceq)$. Then the string $s = a_1 a_2 \ldots a_\upsilon$ is a linearization of $(\Sigma, \preceq)$ if $\upsilon = \mid \Sigma \mid$ and $\forall a_\iota, a_\lambda \in \Sigma$:*

i) $a_\iota = a_\lambda \Rightarrow \iota = \lambda$

ii) for $\iota \neq \lambda$, if $a_\iota \preceq a_\lambda$ then $\iota < \lambda$

In words s is a string obtained considering all the symbols of the set Σ, where each symbol appears only once in the string s and for any two different elements of Σ s.t. $a_\iota \preceq a_\lambda$ then a_ι is considered in s before a_λ.

Then denote by $\langle\Sigma\rangle_\preceq$ the set of all the strings s that are linearizations of $(\Sigma, \preceq)$. Notice that whenever clear from the context we drop the lower index $\preceq$ of $\langle\Sigma\rangle_\preceq$.

2.3 Discrete Event Systems

When the state of a system is described by a discrete set like $\{0, 1, 2, \ldots\}$, and state transitions are only observed at discrete points in time, we associate these state transitions with *events* and talk about *discrete event-systems*.

2.3.1 Automata

An automaton is a mathematical model of a sequential system. It provides a compact way of representing a language. Some well defined operations on automata allow for manipulation and analysis of large and complex languages.

Definition 3. *[CL99] A* Deterministic Automaton*, denoted G is a five-tuple:*

$$G = (X, E, f, g, x_0)$$

where:

X is a countable set of states

E is the set of events associated with the transitions in G

$f : X \times E \rightarrow X$ is the transition function: $f(x, e) = y$ represents that e is defined in state x and when e is executed in x then the state is transformed into y. Since for some state x' the transition e may not be allowed, $f(x, e)$ is a partial function on its domain

$g : X \rightarrow Pwr(E)$ is the active event function (or feasible event function); $g(x)$ is the set of all events e for which $f(x, e)$ is defined and it is called the active event set *(or feasible event set) of G at x.*

x_0 is the initial state

If X is finite then we call G a *deterministic finite state automaton*. G is said to be deterministic because f is defined from $X \times E$ onto X.

Remark 1. *If f is defined from $X \times E$ onto $Pwr(X)$ then G is a* nondeterministic automaton.

The automaton G defines a dynamical system with language $\mathcal{L}$ as follows. It starts in the initial state x_0 and upon the occurrence of an event $e_1 \in g(x_0) \subseteq E$ it will make a transition to state $x_1 = f(x_0, e_1)$. Then $e_2 \in g(x_1)$ can be executed leading to state $x_2 = f(x_1, e_2)$. The process continues executing each time a transitions e_v for which $f(x_v, e_v)$ is defined.

For the sake of convenience f is always extended from domain $X \times E$ to domain $X \times E^*$ in the following manner:

$f(x, \epsilon) \triangleq x$

$f(x, se) \triangleq f(f(x, s), e)$ for $s \in E^*$ and $e \in E$

Definition 4. *The* language generated *by $G = (X, E, f, g, x_0)$ is :*

$$\mathcal{L}_G \triangleq \{s \in E^* : f(x_0, s) \text{ is defined } \}$$

Definition 5. *Consider an automaton $G = (X, E, f, g, x_0)$ and a labeling function $l : E \rightarrow \Omega$ where Ω is a set of labels. Then the labeled language generated by G is:*

$$\mathcal{L}^l_G = \{l(s) : s \in \mathcal{L}_G\}$$

Example 3. *Consider the automaton $G = (X, E, f, g, x_0)$ displayed in Fig.2.1 where: $X = \{x_0, x_1, x_2, x_3, x_4\}$, $E = \{a, b, c\}$. Then we have that: $\mathcal{L}_G = \left\{\overline{(aba)^*cac^*}\right\}$.*

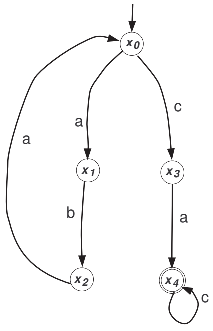


Figure 2.1:

2.3.2 Petri Nets

First developed by C.A. Petri in early 1960's, a Petri Net is a mathematical model for modeling and analyzing distributed systems comprising synchronous and asynchronous activities. A Petri Net is a way of graphically representing automata by using state representation in the form of an integer vector.

In Petri Nets, events in the corresponding automaton correspond to the execution of a transition in a Petri Net. A transition can occur when several conditions are satisfied. These (pre-) conditions are expressed as the places that are input to a transition. The occurrence of a transition affects the output places of a transition (i.e. the post-conditions). Transitions, places and certain relationships between them define the basic components of a Petri Net. Thus Petri Net is a bipartite graph comprising two types of nodes: places and transitions, and arcs connecting places to transitions and transitions to places.

2.3.3 Petri Net notation and definitions

Definition 6. *A Petri Net is a structure* $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ *where:*

$\mathcal{P}$ *denotes the finite set of places,*

$\mathcal{T}$ *denotes the finite set transitions, and*

$F \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ *is the incidence (flow) relation that specifies the arcs from places to transitions* (Pre) *and from transitions to places* $(Post)$*:* $F = Pre \cup Post$*,* $Pre : \mathcal{P} \times \mathcal{T} \to \mathbb{N}$ *and* $Post(t,p) : \mathcal{T} \times \mathcal{P} \to \mathbb{N}$ *where:*

- $Pre(p,t) \in \mathbb{N}$ *gives the weight that is associated with the arc directed from place* p *to transition* t

- $Post(t, p) \in \mathbb{N}$ *gives the weight that is associated with the arc directed from transition t to place p*

If all the arcs have the weight 1 $(\forall p \in \mathcal{P}, \forall t \in \mathcal{T}\ Pre(p, t) \leq 1$ *and* $Post(t, p) \leq 1)$ *then* $\mathcal{N}$ *is an ordinary PN. In this thesis all the PN are considered ordinary. The reason is that both ordinary and non-ordinary PNs have the same modeling power, the only difference is modeling efficiency or convenience [Mur89].*

The graphical representation of a Petri Net $\mathcal{N}$ uses the following convention: a place is represented by a circle and a transition is represented by a bar (box), and for each pair (p, t) that is defined in Pre there is an arc directed from p to t. Similarly for each pair (t, p) that is defined in $Post$ there is an arc directed from t to p.

Denote $\mathcal{X} = \mathcal{P} \cup \mathcal{T}$. Then for $x \in \mathcal{X}$ we use the standard notations $x^\bullet = \{y \in \mathcal{X} \mid xFy\}$, ${}^\bullet x = \{y \in \mathcal{X} \mid yFx\}$ and then for $\mathcal{X}' \subseteq \mathcal{X}$ $\mathcal{X}'^\bullet = \{y \in \mathcal{X} \mid \exists x \in \mathcal{X}' \text{ s.t. } y \in x^\bullet\}$ and ${}^\bullet\mathcal{X}' = \{y \in \mathcal{X} \mid \exists x \in \mathcal{X}' \text{ s.t. } y \in {}^\bullet x\}$.

Definition 7. *A* marking M *of a PN* $\mathcal{N}$ *is represented by a* $\mid \mathcal{P} \mid$*-vector that assigns to each place p of* $\mathcal{P}$ *a non-negative number of tokens* $M : \mathcal{P} \rightarrow \mathbb{N}$.

A PN system is a pair $\langle \mathcal{N}, M_0 \rangle$ *where* $\mathcal{N}$ *is a connected graph having at least one place and one transition and* M_0 *is a marking of* $\mathcal{N}$ *called the initial marking.*

The initial marking M_0 is represented graphically by tokens (dots) drawn inside the circle representing the place p i.e. in p draw $M_0(p)$ tokens. In the following we treat a marking also as a multi-set comprising the places that contain tokens. E.g. $M = \{(p, M(p)) \mid p \in \mathcal{P}$ and $M(p) \neq 0\}$ where $M(p)$ is the number of tokens present in p in the marking M ($M(p)$ stands for $\mu(p)$ when talking about a marking seen as a multi-set of tokens).

Definition 8. *Given a PN* $\mathcal{N}$ *and a marking M, a transition* $t \in \mathcal{T}$ *is enabled in M if* $\forall p \in {}^\bullet t$, $M(p) \geq Pre(p, t)$. *Denote by* $ENABLED(M)$ *the set of all the enabled transitions in the marking M. An enabled transition* $t \in ENABLED(M)$ *in a marking M fires in M and produces the marking M′ where:*

$$M' = M - Pre(\cdot, t) + Post(t, \cdot) \tag{2.1}$$

where abusing notation $Pre(\cdot, t)$ *and* $Post(t, \cdot)$ *are the* $\mid \mathcal{P} \mid$*-vectors whose element p is* $Pre(p, t)$ *respectively* $Post(t, p)$.

In the following we use the notation $M \xrightarrow{t} M'$ for the firing of a transition t transforming the marking (state) of the PN from M to the new marking M'.

Definition 9. *A legal trace* τ *in the PN system* $\langle \mathcal{N}, M_0 \rangle$ *is defined as:*

$$\tau = M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} \ldots \xrightarrow{t_\upsilon} M_\upsilon \tag{2.2}$$

where inductively for $\iota = 1, 2, \ldots, \upsilon$, $M_{\iota-1} \geq Pre(\cdot, t_\iota)$. $M_0 \xrightarrow{\tau} M_\upsilon$ *denotes that the enabling conditions are satisfied so that* τ *fires at* M_0 *yielding* M_υ.

Definition 10 (reachable set). *Given a PN system $\langle \mathcal{N}, M_0 \rangle$ the set of all legal traces in $\langle \mathcal{N}, M_0 \rangle$ is denoted by $\mathcal{L}_{\mathcal{N}}(M_0)$ while the set of reachable markings is:*

$$\mathcal{R}_{\mathcal{N}}(M_0) = \left\{ M \mid \exists \tau \in \mathcal{L}_{\mathcal{N}}(M_0) \; s.t. \; M_0 \xrightarrow{\tau} M \right\} \tag{2.3}$$

Definition 11 (reachability tree). *The set of reachable markings $\mathcal{R}_{\mathcal{N}}(M_0)$ can be represented as a tree $\mathcal{RT}_{\mathcal{N}}(M_0)$ where the set of nodes in $\mathcal{RT}_{\mathcal{N}}(M_0)$ is represented by the set of reachable markings $\mathcal{R}_{\mathcal{N}}(M_0)$; $node_\iota$ and $node_\lambda$ in $\mathcal{RT}_{\mathcal{N}}(M_0)$ (with corresponding markings M_ι and resp. M_λ) are connected by an edge (labeled $t \in \mathcal{T}$) that starts in $node_\iota$ and points to $node_\lambda$ if $M_\iota \xrightarrow{t} M_\lambda$.*

For each node $M_\iota \in \mathcal{RT}_{\mathcal{N}}(M_0)$ denote the set of successors respectively predecessors of M_ι in $\mathcal{RT}_{\mathcal{N}}(M_0)$ by:

$$\begin{aligned} Succ(M_\iota) &= \left\{ M_\lambda \in \mathcal{R}_{\mathcal{N}}(M_0) : \exists t \in \mathcal{T} \;\; s.t. \;\; M_\iota \xrightarrow{t} M_\lambda \right\} \\ Pred(M_\iota) &= \left\{ M_\lambda \in \mathcal{R}_{\mathcal{N}}(M_0) : \exists t \in \mathcal{T} \;\; s.t. \;\; M_\lambda \xrightarrow{t} M_\iota \right\} \end{aligned}$$

Then denote $Succ(M_\iota)^*$ resp. $Pred(M_\iota)^*$ the transitive closure of the successor-nodes resp. predecessor-nodes of a node M_ι:

$$\begin{aligned} Succ(M_\iota)^* &= Succ(M_\iota) \cup Succ(Succ(M_\iota)) \cup \ldots \cup Succ(\ldots Succ(M_\iota) \ldots) \\ Pred(M_\iota)^* &= Pred(M_\iota) \cup Pred(Pred(M_\iota)) \cup \ldots \cup Pred(\ldots Pred(M_\iota) \ldots) \end{aligned}$$

A node without predecessor nodes is called a root *of $\mathcal{RT}_{\mathcal{N}}(M_0)$ while a node that has no successor nodes is called a* terminal node.

In Section 7.1 in Appendix we provide the pseudo-code of the Carp-Miller algorithm for computing the reachability tree $\mathcal{RT}_{\mathcal{N}}(M_0)$ (Algorithm *Reach_Tree(M_0)*).

Consider a legal trace $\sigma \in \mathcal{L}_{\mathcal{N}}(M_0)$. The Parikh vector associated with σ is denoted $\overrightarrow{\sigma}$ and is a $\mid \mathcal{T} \mid$-vector whose element ι that corresponds with transition $t_\iota \in \mathcal{T}$ is given by $\mu_\sigma(t_\iota)$ that is the number of appearances of t_ι in the legal trace σ).

Lemma 1 (marking equation lemma). *If $M_0 \xrightarrow{\sigma} M$ then the following Marking Equation holds:*

$$M_0 + F \cdot \overrightarrow{\sigma} = M \tag{2.4}$$

(with the incidence relation F in a matrix representation)

Notice that in a general PN $\mathcal{N}$, Equation 2.4 is a necessary but not sufficient condition for checking if a marking M is reachable from M_0 by firing a trace σ.

However if $\mathcal{N}$ is acyclic then Equation 2.4 is a necessary and sufficient condition for the reachability problem [Mur89].

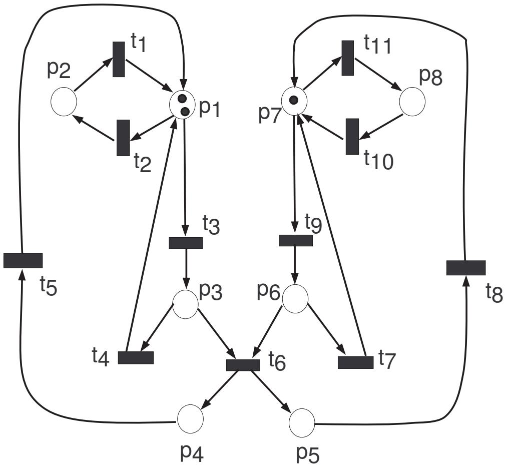


Figure 2.2:

Example 4. *Consider the PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ displayed in Fig. 2.2. The initial marking is represented by tokens (black dots) placed in some places. In this example there are two tokens in p_1 and one token in p_7. The set of enabled transitions in M_0 is $ENABLED(M_0) = \{t_2, t_3, t_9, t_{11}\}$. Let t_2 be executed first from M_0. A token is removed from p_1 and is added to p_2. The resulting marking is $M' = \{(p_1, 1); (p_2, 1); (p_7, 1)\}$ and the set of enabled transitions in this new state is $ENABLED(M') = \{t_1, t_2, t_3, t_9, t_{11}\}$.*

Let then t_{11} be executed first. A token is removed from p_7 and is added to p_8. The resulting marking is $M'' = \{(p_1, 2), (p_8, 1)\}$ and the set of enabled transitions in this new state is $ENABLED(M'') = \{t_2, t_3, t_{10}\}$.

Consider a feasible sequence of transitions $\sigma = t_2t_3t_1t_3$. The Parikh vector associated with σ is $\overrightarrow{\sigma} = [1, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0]^T$.

Then the marking M that results by firing σ from M_0 ($M_0 \xrightarrow{\sigma} M$) is calculated from the state equation:

$$M = M_0 + F \cdot \overrightarrow{\sigma}$$

where $M_0 = [2, 0, 0, 0, 0, 0, 1, 0]^T$,

$$F = \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

and $M = [0,0,2,0,0,0,1,0]^T$.

Definition 12. *Given a PN $\mathcal{N}$ then we say that:*

- *if* $\mid {}^\bullet t \mid > 1$ *then t is an input synchronizing transition*
- *if* $\mid t^\bullet \mid > 1$ *then t is an output synchronizing transition*
- *if* $\mid p^\bullet \mid > 1$ *then p is a forward choice place and $t \in p^\bullet$ is a forward choice transition of the place p*
- *if* $\mid {}^\bullet p \mid > 1$ *then p is a backward choice place and $t \in {}^\bullet p$ is a backward choice transition of the place p*
- *a* state machine *is a PN $\mathcal{N}$ such that each transition t has exactly one input place and exactly one output place, i.e.*

  $$\forall t \in \mathcal{T}, \mid {}^\bullet t \mid = 1 \textit{ and } \mid t^\bullet \mid = 1$$

  A state machine is a PN without synchronizing transition thus tokens in a state machine move independently; a state machine $\mathcal{N}$ with N tokens in a place compactly represents N copies of an automaton having the same structure with $\mathcal{N}$.

- *a* marked graph *is a PN $\mathcal{N}$ such that each place has at most one output transition, i.e.*

  $$\forall p \in \mathcal{P}, \mid p^\bullet \mid \leq 1$$

  A marked graph is a PN without choices; the temporal analysis of a marked graph can be made using $(max, +)$ algebra and is a lot a easier than for the general class of PNs.

- *a causal net (CN) is a PN $\mathcal{N}$ s.t. each place has at most one input transition and at most one output transition, i.e.*

  $$\forall p \in \mathcal{P}, \mid p^\bullet \mid \leq 1 \textit{ and } \mid {}^\bullet p \mid \leq 1$$

  In a causal net a token in a place p can be produced respectively removed from p by firing unique transitions.

Definition 13 ([DE95]). *A PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ is free-choice if $(p,t) \in F$ implies $^\bullet t \times p^\bullet \subseteq F$ for every place p and transition t.*

Equivalently $\mathcal{N}$ is a free-choice iff for every two transitions $t_1, t_2 \in \mathcal{T}$ either $^\bullet t_1 \cap {}^\bullet t_2 = \emptyset$ or $^\bullet t_1 = {}^\bullet t_2$.

Definition 14 ([DE95]). *Consider a PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ and a node $x \in \mathcal{X}$. The cluster of x, denoted $cluster(x)$, is the minimal set of nodes s.t.:*

$x \in cluster(x)$

if a place p belongs to $cluster(x)$ then $p^\bullet$ is included in $cluster(x)$

if a transition t belongs to $cluster(x)$ then $^\bullet t$ is included in $cluster(x)$

We have that if $x' \in cluster(x)$ then $cluster(x) = cluster(x')$. Thus the partition of a PN in clusters is unique.

Definition 15. *Consider two PNs $\mathcal{N}_1 = (\mathcal{P}_1, \mathcal{T}_1, F_1)$ and $\mathcal{N}_2 = (\mathcal{P}_2, \mathcal{T}_2, F_2)$. Then $\langle \mathcal{N}_1, M_{0_1} \rangle$ is a sub-net of $\langle \mathcal{N}_2, M_{0_2} \rangle$ if:*

i) $\mathcal{P}_1 \subseteq \mathcal{P}_2$

ii) $\mathcal{T}_1 \subseteq \mathcal{T}_2$

iii) $Pre_1 = Pre_2 \mid_{\mathcal{T}_1 \times \mathcal{P}_1}$

iv) $Post_1 = Post_2 \mid_{\mathcal{P}_1 \times \mathcal{T}_1}$

v) $M_{0_1} = M_{0_2} \mid_{\mathcal{P}_1}$

Conditions $i) - iv)$ state that $\mathcal{N}_1$ is a sub-graph of $\mathcal{N}_2$, where conditions $iii)$ and $iv)$ state that Pre_1 and $Post_1$ are the restriction of Pre_2, respectively $Post_2$ to the domains $\mathcal{T}_1 \times \mathcal{P}_1$, respectively $\mathcal{P}_1 \times \mathcal{T}_1$. Condition $v)$ states that the initial marking M_{0_1} is the restriction of the marking M_{0_2} to the places $\mathcal{P}_1$.

If $\langle \mathcal{N}_1, M_{0_1} \rangle$ is a sub-net of $\langle \mathcal{N}_2, M_{0_2} \rangle$ and in addition $\forall t \in \mathcal{T}_1$, all the input places and the output places of t in $\mathcal{N}_2$ are contained in $\mathcal{P}_1$ then we say that $\langle \mathcal{N}_1, M_{0_1} \rangle$ is a proper sub-net of $\langle \mathcal{N}_2, M_{0_2} \rangle$.

In the following, whenever we refer to $\mathcal{N}'$ as a subnet of a given PN $\mathcal{N}$ we implicitly assume that $\mathcal{N}'$ is a proper subnet of $\mathcal{N}$.

Definition 16 ([Mur89]). *Given a PN $\mathcal{N} = \langle \mathcal{P}, \mathcal{T}, F \rangle$ a subset of places $\mathcal{P}' \subseteq \mathcal{P}$ is a trap respectively a siphon if $\mathcal{P}'^\bullet \subseteq {}^\bullet\mathcal{P}'$, respectively $^\bullet\mathcal{P}' \subseteq \mathcal{P}'^\bullet$.*

A trap has the property that if it is marked (i.e. it has at least one token) under some marking, then it remains marked under the successor marking. A siphon has the property that if it is token-free (i.e. it has no token) under some marking, then it remains token free under each successor marking.

Definition 17. *A path of a PN $\mathcal{N}$ is a non-empty sequence $\wp = x_1 \ldots x_\upsilon$ of nodes that satisfies $(x_1, x_2), \ldots, (x_{\upsilon-1}, x_\upsilon) \in F$. A path $\wp = x_1 \ldots x_\upsilon$ is said to lead from x_1 to x_υ. A path $\wp$ leading from a node x to a node y is a circuit if no element occurs more than once in it and $(y, x) \in F$. Notice that a sequence containing one element is a path but not a circuit since $(x, x) \notin F$.*

Definition 18 ([Mur89]). *A PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ is called:*

- *connected if every two nodes x, y satisfy $(x, y) \in (F \cup F^{-1})^*$.*
- *acyclic if $\mathcal{N}$ has no circuits.*
- *trap-circuit PN if the set of places in every directed circuit is a trap.*
- *siphon-circuit PN if the set of places in every directed circuit is a siphon.*

Example 5. *Consider the PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ displayed in Fig. 2.2. $\wp = p_1 t_3 p_3 t_6$ is a path in $\mathcal{N}$ while $\zeta = p_1 t_3 p_3 t_6 p_4 t_5$ is a circuit. p_1 is a choice place, and t_2 and t_3 are the choice transitions of p_1. $\mathcal{N}$ is not a free-choice net because $^\bullet t_4 \cap {}^\bullet t_6 \neq \emptyset$ and $^\bullet t_4 \neq {}^\bullet t_6$. The clusters associated with t_4, t_6, t_7 are equal: $cluster(t_4) = cluster(t_6) = cluster(t_7) = \{p_3, p_6, t_4, t_6, t_7\}$.*

2.3.4 Properties of PNs

Lemma 2 (monotonicity). *Given $\langle \mathcal{N}, M_0 \rangle$ and $\langle \mathcal{N}, M_0' \rangle$ such that $M_0 \leq M_0'$ then:*

$$\mathcal{L}_{\mathcal{N}}(M_0) \subseteq \mathcal{L}_{\mathcal{N}}(M_0') \tag{2.5}$$

Definition 19. *A PN $\langle \mathcal{N}, M_0 \rangle$ is bounded if for every place $p \in \mathcal{P}$ there is a natural number k s.t. $M(p) \leq k$ for any $M \in \mathcal{R}_{\mathcal{N}}(M_0)$. $\langle \mathcal{N}, M_0 \rangle$ is* k-bounded *if no place has a bound greater than k. If the bound k is equal to 1 for all places $p \in \mathcal{P}$, then $\langle \mathcal{N}, M_0 \rangle$ is* 1-safe *PN.*

Lemma 3 (boundedness lemma). *Let $\langle \mathcal{N}, M_0 \rangle$ be bounded and $M \in \mathcal{R}_{\mathcal{N}}(M_0)$. If $M \leq M'$ and $M \xrightarrow{\tau} M'$ then $M = M'$.*

Definition 20. *Given $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ and $\mathcal{T}' \subseteq \mathcal{T}$ then $\mathcal{N}$ is structurally bounded w.r.t. $\mathcal{T}'$ iff for any initial marking $M_0 \in \mathbb{N}^{|\mathcal{P}|}$ and $\forall \tau \in \mathcal{T}'^*$ we have that:*

$$\textit{if } M_0 \xrightarrow{\tau} M \textit{ and } M_0 \leq M \textit{ then } M_0 = M$$

Definition 21. *Consider a PN $\langle \mathcal{N}, M_0 \rangle$ and a labeling function $l : \mathcal{T} \rightarrow \Omega$ where Ω is a set of labels. Then extend the definition of l to strings in the obvious manner i.e. for $\tau \in \mathcal{T}^*$, $\tau = t_1 t_2 \ldots t_\lambda$ we have $l(\tau) = l(t_1) l(t_2) \ldots l(t_\lambda)$.*

The labeled language generated by $\langle \mathcal{N}, M_0 \rangle$ is:

$$\mathcal{L}^l_{\mathcal{N}}(M_0) = \{ l(\tau) : \tau \in \mathcal{L}_{\mathcal{N}}(M_0) \}$$

Definition 22. *Denote by $\mathcal{T}^*$ the Kleene closure of the set $\mathcal{T}$. Then let $\sigma \in \mathcal{L}_{\mathcal{N}}(M_0) \subseteq \mathcal{T}^*$ and $\mathcal{T}' \subset \mathcal{T}$. The projection $\Pi_{\mathcal{T}'} : \mathcal{L}_{\mathcal{N}}(M_0) \rightarrow \mathcal{T}'^*$ is defined as:*

i) $\Pi_{\mathcal{T}'}(\epsilon) = \epsilon$*;*

ii) $\Pi_{\mathcal{T}'}(t) = t$ *if* $t \in \mathcal{T}'$*;*

iii) $\Pi_{\mathcal{T}'}(t) = \epsilon$ *if* $t \in \mathcal{T} \setminus \mathcal{T}'$*;*

iv) $\Pi_{\mathcal{T}'}(\sigma t) = \Pi_{\mathcal{T}'}(\sigma)\Pi_{\mathcal{T}'}(t)$ *for* $\sigma \in \mathcal{L}_{\mathcal{N}}(M_0)$ *and* $t \in \mathcal{T}$*.*

Example 6. *For the PN displayed in Fig. 2.2 consider a labeling function $l : \mathcal{T} \rightarrow \{a, b\}$ where $l(t_\iota) = a$ if ι is even and $l(t_\iota) = b$ if ι is odd. For $\sigma = t_9t_2t_3t_1t_3t_6$ we have $l(\sigma) = babbba$.*

Consider $\mathcal{T}' = \{t_1, t_2, t_3, t_4, t_5\}$. The projection of σ on to $\mathcal{T}'$ is $\Pi_{\mathcal{T}'}(\sigma) = t_2t_3t_1t_3$. Then for $\Omega' = \{a\}$, the projection of $l(\tau)$ on to Ω' is $\Pi_{\Omega'}(l(\tau)) = aa$.

Definition 23. *A* P-invariant *of a PN $\mathcal{N}$ is a rational-valued solution of the equation $\mathbf{X} \cdot F = \mathbf{0}$ where $\mathbf{X}$ is a $\mid \mathcal{T} \mid$-vector and $\mathbf{0}$ is a $\mid \mathcal{P} \mid$ vector that has all its elements* 0*. We call a* P-invariant $\mathbf{X}$ *semi-positive if $\forall t \in \mathcal{T}$, $\mathbf{X}(t) \geq \mathbf{0}$ and $\mathbf{X} \neq \mathbf{0}$ while a* P-invariant $\mathbf{X}$ *is called positive if $\mathbf{X}(p) > 0$ for all $p \in \mathcal{P}$.*

Proposition 1. *Given $\langle \mathcal{N}, M_0 \rangle$, let X be a* P-invariant *of $\mathcal{N}$. If $M_0 \xrightarrow{*} M$ then* $\mathbf{X} \cdot M_0 = \mathbf{X} \cdot M$

Definition 24. *A* T-invariant *of a net $\mathcal{N}$ is a rational-valued $\mid \mathcal{P} \mid$-vector solution of the equation $F \cdot \mathbf{X} = \mathbf{0}$.*

Proposition 2. *Let σ be a finite sequence of transitions of a PN $\mathcal{N}$ which is enabled at a marking M. Then the Parikh vector $\vec{\sigma}$ is a* P-invariant *if $M \xrightarrow{\sigma} M$ (i.e., if the occurrence of σ reproduces the marking M).*

Example 7. *Consider the PN $\langle \mathcal{N}, M_0 \rangle$ displayed in Fig. 2.2. The Parikh vectors $\vec{\sigma}_\iota$ ($\iota = 1, \ldots, 5$) of the following traces are* P-invariants *in $\langle \mathcal{N}, M_0 \rangle$: $\sigma_1 = t_1t_2$; $\sigma_2 = t_3t_4$; $\sigma_3 = t_{11}t_{10}$; $\sigma_4 = t_9t_7$; $\sigma_5 = t_3t_9t_6t_5t_8$. Notice that the addition vector $\vec{\sigma}_{\iota\lambda}$ of two vectors $\vec{\sigma}_\iota$ and $\vec{\sigma}_\lambda$ that are* P-invariants *($\vec{\sigma}_{\iota\lambda} = \vec{\sigma}_\iota + \vec{\sigma}_\lambda$) is also a* P-invariant.

2.4 The observations and the fault representation for PN models

As already mentioned in the introduction, our aim is to design on-line algorithms for fault detection and diagnosis. Generally speaking this task comprises two steps: i) to derive the plant behaviour that explains the plant observation and then ii) to check whether *"something wrong"* has happened in the plant or not. In this section we discuss how such faults are represented in the models and how the plant observation (sensor readings) can be taken into account in the on-line algorithms.

2.4.1 The plant observation

We consider in the following the most common way of representing the plant observations that is *event-observations* (the event occurrences are reported), although state-observation (the place marking is observed) or a combination of observation of both places and transitions is possible.

Consider a labeling function $l_o : \mathcal{T} \to \Omega$, with Ω the set of observation-labels $l_o(t)$ that are received by the monitoring systems when the event t is executed. Thus l_o may be interpreted as the observation mask under which the monitoring/supervisory systems observes the occurrences of the events in the plant. The observation and fault detection problem becomes interesting because in practice sensors are attached only to some events in the plant. Thus let $\mathcal{T}$ be partitioned into two disjoint sets $\mathcal{T} = \mathcal{T}_o \cup \mathcal{T}_{uo}$ where $\mathcal{T}_o$ denotes the set of observable events and $\mathcal{T}_{uo}$ denotes the set of unobservable events. Notice that the assumption we made implies that the occurrence of an unobservable event is silent (i.e. it is not at all notified to the monitoring system) since there is no sensor at all to do this.

Hence, to formally model this, let the observation labeling function be defined as follows (recall that ϵ is the empty string and $\epsilon a = a = a\epsilon$ and Ω_o does not include ϵ).

Definition 25. *Given a PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ and a partition $\mathcal{T} = \mathcal{T}_o \cup \mathcal{T}_{uo}$, then $l_o : \mathcal{T} \to \Omega_o \cup \{\epsilon\}$ is the observation labeling function of $\mathcal{N}$ where $l_o(t) \in \Omega_o$ if $t \in \mathcal{T}_o$ and $l_o(t) = \epsilon$ if $t \in \mathcal{T}_{uo}$.*

If l_o is s.t. $\forall\, t_\iota, t_\lambda \in \mathcal{T}_o$, $l_o(t_\iota) = l_o(t_\lambda) \Rightarrow t_\iota = t_\lambda$ we say that the observation labeling function l_o is deterministic (injective in $\mathcal{T}_o$) otherwise l_o is said to be a non-deterministic observation function.

To simplify the notation we present in this thesis only the design of the fault detection and diagnosis algorithms for a deterministic observation of the plant, and we only discuss briefly in some remarks what are the implications of a non-deterministic labeling on the construction of the algorithms. Thus unless otherwise stated, l_o is a deterministic observation and obviously we chose the label that is emitted by an observable transition to be the name of the transition.

Assumption 1. *Unless otherwise stated we also make the following important assumptions:*

1 *the observation is emitted and received correctly (no corrupted messages)*

2 *the observation is always received (no loss of observation)*

3 *the observation is totally ordered i.e. if $l_o(t_\iota)$ was received before $l_o(t_\lambda)$ then t_ι happened before t_λ*

Notice that item 2 and 3 in Assumption 1 above maybe relaxed if one assumes a maximal bound on the number of observed events that are lost [LZ02] respectively a maximum delay in receiving the plant observation [DLT03]. Notice that item 1 can also be relaxed but all three relaxations are beyond the scope of this thesis.

Formulation of the basic observation problem

Consider a PN model $\langle\mathcal{N}, M_0\rangle$ where $\mathcal{T} = \mathcal{T}_o \cup \mathcal{T}_{uo}$ and l_o is the observation labeling function. When the feasible string of transitions $\tau \in \mathcal{L}_{\mathcal{N}}(M_0)$, $\tau = t_1 t_2 \dots t_\lambda$ is executed by the plant, the observation that is received by the monitoring system (denoted $\mathcal{O}$) is $\mathcal{O} = l_o(\tau)$. If l_o is deterministic (i.e. $l_o(t_\iota) = t_\iota$ for $t_\iota^o \in \mathcal{T}_o$) we have that $\mathcal{O} = \Pi_{\mathcal{T}_o}(\tau)$.

Conversely given a received observation $\mathcal{O} = t_1^o t_2^o \dots t_n^o$ the set of plant behaviours that obey (and explain) the observation $\mathcal{O}$ is:

$$\mathcal{L}_{\mathcal{N}}(\mathcal{O}) = \{\tau \in \mathcal{L}_{\mathcal{N}}(M_0) \mid \Pi_{\mathcal{T}_o}(\tau) = \mathcal{O}\} \tag{2.6}$$

2.4.2 Fault representation

As stated in the introduction, the paradigm for fault detection and diagnosis that we choose is to explicitly represent all the faults that the model-based fault detector can and must detect.

As commonly accepted a fault is understood as any kind of malfunction in the plant, that leads to unacceptable future behaviour of the overall plant [Fra90].

In this work the faults in the PN models are represented as (fault) transitions whose occurrence indicates a malfunction in the plant behavior [SSL+95]. Obviously the set of fault transitions (denoted $\mathcal{T}_f$) is a subset of the set of unobservable transitions ($\mathcal{T}_f \subseteq \mathcal{T}_{uo}$) since otherwise the fault detection problem would be trivial.

The fault labeling function l_f is defined on $\mathcal{T}_f$ taking values on the set of fault-labels Ω_f ($l_f : \mathcal{T}_f \to \Omega_f$). By l_f we represent that a fault event $t \in \mathcal{T}_f$ is of kind $\mathtt{F}_\iota$ if $l_f(t) = \mathtt{F}_\iota$. Unless otherwise stated we assume in this thesis that l_f is injective and then choose as fault-label the name of the fault transition.

Beside the fact that a fault must be unobservable, it must also be unpredictable, i.e. for any state the plant can be in before the occurrence of a fault at least one no-fault event is legal according to the plant model; otherwise the fault would be imminent or predictable and, consequently, the model is regarded as incorrect (an earliest event should have been labeled as a fault). We formalize this as follows.

Assumption 2. *Given a PN model $\langle \mathcal{N}, M_0 \rangle$ and $\mathcal{T}_f$ ($\mathcal{T}_f \subseteq \mathcal{T}_{uo}$) the set of fault transitions, then for any reachable state $M \in \mathcal{R}_{\mathcal{N}}(M_0)$, at least one non-fault transition t, $t \in \mathcal{T} \setminus \mathcal{T}_f$ is enabled, that is:*

$$\forall M \in \mathcal{R}_{\mathcal{N}}(M_0), ENABLED(M) \nsubseteq \mathcal{T}_f$$

The straightforward implication of Assumption 2 is that:

$$\text{if } t \in \mathcal{T}_f \text{ then } \exists t' \in \mathcal{T} \setminus \mathcal{T}_f \text{ s.t. } {}^{\bullet}t \cap {}^{\bullet}t' \neq \emptyset$$

In words Assumption 2 says that: *"a fault is the choice of the plant of not respecting the good (designed) behavior".*

Remark 2. *Notice that there are some other representations of a fault for Petri Net models (and DES in general). Among others a fault may also be represented:*

- *by the* violation of the model dynamics *e.g. tokens may disappear from places or the execution of a transition violates the firing rule [HV99] (e.g. not producing tokens in the output places); similarly for automata models a fault may be represented by the execution of an event that leads to a state different than the one specified by the transition function [OW90].*

- *or by logical propositions defined over a set of logical variables that comprise both places and transitions [JK04]*

2.5 Occurrence Nets and Net Unfolding

Unfortunately the complexity of the PN reachability analysis has been proven to be EXPSPACE-hard in the general case. This is because in the standard reachability algorithm (the Carp-Miller algorithm) all the possible interleavings of the concurrent transitions are considered.

The methods that were proposed for reducing the state-space explosion problem are based on the observation that for reachability analysis not all interleavings of a given set of independent transitions need to be considered.

Among others (e.g. *stubborn sets* [Val90] and *persistent sets* [GW93]) a method that has received a lot of consideration over the past years is that of net unfoldings [Eng91], [McM93], [Esp94].

The unfolding of a PN is an occurrence net (i.e. an ordinary Petri Net without cycles) that is behaviorally equivalent with the original net. Unfoldings are usually *infinite* nets. However it is always possible to construct a finite *initial prefix* of the unfolding which captures its entire behavior [McM93]. The *initial prefix* of the unfolding has the property that it contains all the reachable states of the whole unfolding and being finite it can be stored in a computer. Besides initial prefixes can be constructed such that they are never larger and in general a lot smaller than the state space of the original PN [ERV96].

Thus the net unfolding technique represents a useful technique for attacking the state explosion problem since for PN models whose degree of concurrency is high compared to their degree of (forward) branching the unfolding method reduces the cost of the analysis from exponential in the size of parameter to polynomial.

The second advantage of unfoldings is that they contain information about causality, conflicts, and concurrency. As it will be presented later this is very useful in a distributed setting to derive the causality between the border-conditions, information that is required to be exchanged between the local agents for achieving the consistency of their local results.

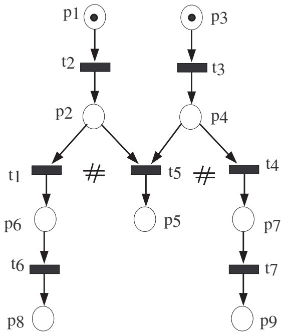


Figure 2.3:

Definition 26. *Given a PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ the immediate dependence relation $\preceq_1 \subset (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ is defined as:*

$$\forall (a, b) \in (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P}) : a \preceq_1 b \textit{ if } F(a, b) \neq 0$$

Then define $\preceq$ as the transitive closure of $\preceq_1$ ($\preceq = \preceq_1^*$).

Example 8. *Consider the PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ displayed in Fig. 2.3. We have that $p_1 \preceq t_2 \preceq p_2 \preceq t_1$.*

Definition 27. *Given a PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ the immediate conflict relation $\sharp_1 \subset \mathcal{T} \times \mathcal{T}$ is defined as:*

$$\forall (t_1, t_2) \in \mathcal{T} \times \mathcal{T} : t_1 \sharp_1 t_2 \textit{ if } {}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$$

Then define $\sharp \subset (\mathcal{P} \cup \mathcal{T}) \times (\mathcal{P} \cup \mathcal{T})$ as:

$$\forall (a,b) \in (\mathcal{P} \cup \mathcal{T}) \times (\mathcal{P} \cup \mathcal{T}) : a \sharp b \textit{ if } \exists t_1, t_2 \textit{ s.t. } t_1 \sharp_1 t_2 \textit{ and } t_1 \preceq a \textit{ and } t_2 \preceq b$$

Example 9. *Consider the PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ displayed in Fig. 2.3. We have that $t_1 \sharp_1 t_5$ and $t_4 \sharp_1 t_5$ are in immediate conflict. $t_6 \sharp t_5$ since $t_1 \preceq t_6$ and $t_1 \sharp_1 t_5$.*

Definition 28. *Given a PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ the independence relation $\| \subset (\mathcal{P} \cup \mathcal{T}) \times (\mathcal{P} \cup \mathcal{T})$ is defined as:*

$$\forall (a,b) \in (\mathcal{P} \cup \mathcal{T}) \times (\mathcal{P} \cup \mathcal{T}) : a \| b \Rightarrow \neg(a \sharp b) \wedge a \npreceq b \wedge b \npreceq a$$

Example 10. *Consider the PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ displayed in Fig. 2.3. We have that $t_2 \parallel t_3$, $p_6 \parallel t_4$, etc.*

Definition 29. *Given two PNs $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ and $\mathcal{N}' = (\mathcal{P}', \mathcal{T}', F')$, ϕ is a homomorphism from $\mathcal{N}$ to $\mathcal{N}'$, denoted $\phi : \mathcal{N} \rightarrow \mathcal{N}'$ where:*

1. *$\phi(\mathcal{P}) \subseteq \mathcal{P}'$ and $\phi(\mathcal{T}) \subseteq \mathcal{T}'$*
2. *$\forall t \in \mathcal{T}$, the restriction of ϕ to $^\bullet t$ is a bijection between $^\bullet t$ and $^\bullet \phi(t)$*
3. *$\forall t \in \mathcal{T}$, the restriction of ϕ to $t^\bullet$ is a bijection between $t^\bullet$ and $\phi(t)^\bullet$*

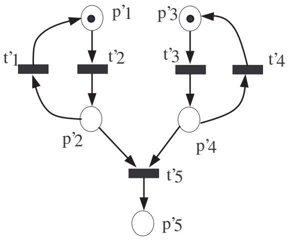


Figure 2.4:

Example 11. *Consider the PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ displayed in Fig. 2.3 and the PN $\mathcal{N}' = (\mathcal{P}', \mathcal{T}', F')$ displayed in Fig. 2.4. Then consider $\phi : \mathcal{N} \rightarrow \mathcal{N}'$ defined as:*
- $\phi(p_\iota) = p'_\iota$ for $\iota = 1, 2, 3, 4, 5$
- $\phi(p_6) = p'_1$; $\phi(p_7) = p'_3$; $\phi(p_8) = p'_2$; $\phi(p_9) = p'_4$
- $\phi(t_\iota) = t'_\iota$ for $\iota = 1, 2, 3, 4, 5$
- $\phi(t_6) = t'_2$; $\phi(t_7) = t'_3$;

Definition 30. *An occurrence net is a net $\mathfrak{O} = (B, E, \preceq_1)$ such that:*

i) $\forall a \in B \cup E : \neg(a \preceq a)$ (acyclic)

ii) $\forall a \in B \cup E : | \{b : a \preceq b\} | < \infty$ (well-formed) *iii)* $\forall b \in B : \mid {}^{\bullet}b \mid \leq 1$ *(no backward conflict)*

In the following B is referred as the set of conditions while E is the set of events. Denote by X^{con} a set of pairwise concurrent nodes, by X_E^{con} a set of concurrent events, and by X_B^{con} a set of concurrent conditions. A maximal (w.r.t. set inclusion) set of concurrent conditions is called a CUT.

Example 12. *The PN* $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ *displayed in Fig. 2.3 is an occurrence net where* $\mathcal{P}$ *stands for* B*,* $\mathcal{T}$ *stands for* E *and* $F = \preceq_1$*.* $CUT_1 = \{p_1, p_3\}$*;* $CUT_2 = \{p_2, p_3\}$*;* $CUT_3 = \{p_1, p_9\}$*, ...*

Remark 3. *The partial order relation* $\preceq$ *introduces the roughest notion of time. For instance* $a, b \in E$ *s.t.* $a \prec b$ *can be interpreted as "*a happens before b*".*

Definition 31. *A configuration* $C = (B_C, E_C, \preceq)$ *in the occurrence net* $\mathfrak{O}$ *is defined as follows:*

i) C *is a proper sub-net of* $\mathfrak{O}$ $(C \subseteq \mathfrak{O})$

ii) C *is conflict free, i.e.* $\forall a, b \in (B_C \cup E_C) \times (B_C \cup E_C) \Rightarrow \neg(a \sharp b)$

iii) C *is causally upward-closed, i.e.* $\forall b \in B_C \cup E_C : a \in B \cup E$ *and* $a \preceq_1 b \Rightarrow a \in B_C \cup E_C$

iv) $\mathtt{min}_{\preceq}(C) = \mathtt{min}_{\preceq}(\mathfrak{O})$

Denote by $\mathcal{C}$ *the set of all the configurations* C *of the occurrence net* $\mathfrak{O}$*.*

Definition 32. *Consider a PN* $\langle \mathcal{N}, M_0 \rangle$ *s.t.* $\forall p \in \mathcal{P} : M_0(p) \in \{0, 1\}$*. A branching process* $\mathfrak{B}$ *of a PN* $\langle \mathcal{N}, M_0 \rangle$ *is a pair* $\mathfrak{B} = (\mathfrak{O}, \phi)$ *where* $\mathfrak{O}$ *is an occurrence net and* ϕ *is a homomorphism* $\phi : \mathfrak{O} \rightarrow \mathcal{N}$ *s.t.:*

1. *the restriction of* ϕ *to* $\mathtt{min}_{\preceq}(\mathfrak{O})$ *is a bijection between* $\mathtt{min}_{\preceq}(\mathfrak{O})$ *and* M_0 *(the set of initially marked places)*
2. $\phi(B) \subseteq \mathcal{P}$ *and* $\phi(E) \subseteq \mathcal{T}$
3. $\forall a, b \in E : ({}^{\bullet}a = {}^{\bullet}b) \wedge (a^{\bullet} = b^{\bullet}) \Rightarrow a = b$

For a PN $\mathcal{N} = \langle \mathcal{P}, \mathcal{T}, F \rangle$ with an initial marking $M_0 \in \mathbb{N}^{|\mathcal{P}|}$ s.t. $\forall p \in \mathcal{P}$, $M_0(p) \leq 1$ the branching process $\mathfrak{B} = (\mathfrak{O}, \phi)$ of $\langle \mathcal{N}, M_0 \rangle$ is constructed in the following way (see [McM93]):

1. For each place p that contains a token in the initial marking M_0 make a condition-node b in the occurrence net and then label the condition-nodes appropriately ($\phi(b) = p$).
2. Choose a transition t from $\mathcal{T}$.

3. For each place p' in $\mathcal{N}$ that is an input place to t ($p' \in {}^{\bullet}t$) find a condition-node b' in the occurrence net whose label corresponds to p' ($\phi(b') = p'$) and mark it with a token (if you can't find a copy, go back to step 2). For a given t, do not choose the same subset of condition-nodes in the occurrence net twice.

4. If any of the places that are marked with tokens are *not concurrent*, go to step 2. Notice that two condition-nodes b, b' in the occurrence net are said to be *concurrent* if:

   (a) there is not an oriented path from b to b' or *vice-versa* (no causal relation between b and b')

   (b) and there is not third place b'' from which are distinct oriented paths that reach b and b' (no conflict between b and b')

5. Make an event-node e in the occurrence net and label it $\phi(e) = t$. Draw an arrow from every condition-node b' that was marked in the occurrence net to e. Erase the tokens.

6. For each place p'' that is an output place of t in $\mathcal{N}$ ($p'' \in t^{\bullet}$) make a condition-node b'' in the occurrence net and label it $\phi(b'') = p''$ and draw an arc from e to b''

7. Repeat the steps $2 - 6$

Denote by $C^{\perp} = (B_{C^{\perp}}, E_{C^{\perp}}, \preceq)$ the initial configuration of the occurrence net $\mathfrak{O}$ where $B_{C^{\perp}} = \{b \in B : {}^{\bullet}b = \emptyset\}$ is the set of condition-nodes in $\mathfrak{O}$ that correspond with the places that contain a token in initial marking ($B_{C^{\perp}} = \mathtt{min}_{\preceq}(\mathfrak{O})$) and $E_{C^{\perp}} = \emptyset$.

For a configuration C in $\mathfrak{O}$ denote by $CUT(C)$ the maximal (w.r.t. set inclusion) set of conditions in C that have no successors in C:

$$CUT(C) = \{e^{\bullet} \mid e \in E_C\} \cup \mathtt{min}_{\preceq}(\mathfrak{O}) \setminus \{\,{}^{\bullet}e \mid e \in E_C\}$$

Then denote by $mark(C)$ the marking in $\mathcal{N}$ that corresponds with $CUT(C)$:

$$mark(C) = \phi(CUT(C))$$

Obviously we have that $CUT(C^{\perp}) = B_{C^{\perp}} = \mathtt{min}_{\preceq}(\mathfrak{O})$ and $mark(C^{\perp}) = \phi(CUT(C^{\perp})) = M_0$ (where a marking is seen as a multi-set of tokens).

Then denote by $ENABLED(C)$ the set of transitions that are enabled in $\mathcal{N}$ from the marking $mark(C)$.

For an enabled transition t make an event e s.t. $t = \phi(e)$ as described by the steps 2-6 above. We say that C is extended by e and denote the configuration that is obtained by $C' = C \odot e$. We have that $C' = (B_{C'}, E_{C'}, \preceq)$ where $B_{C'} = B_C \cup \{e^{\bullet}\}$ and $E_{C'} = E_C \cup \{e\}$.

Consider two configurations C and C' s.t. C' is obtained from C by appending the events $e_1, \ldots, e_q$ ($C' = C \odot e_1 \odot \ldots \odot e_q$). We have that C is a proper sub-net of C' and we say that C is a prefix of C' and denote this as $C \sqsubset C'$.

For a PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ with a general initial marking $M_0 \in \mathbb{N}^{|\mathcal{P}|}$ the branching process $\mathfrak{O}$ of $\langle \mathcal{N}, M_0 \rangle$ is constructed as follows (see [Eng91]):

1. let $\mathcal{N}' = \langle \mathcal{P}', \mathcal{T}', F' \rangle$ where:
   1.1 $\mathcal{P}' = \mathcal{P} \cup \{\mathcal{P}_{start}\}$
   1.2 $\mathcal{T}' = \mathcal{T} \cup \{\mathcal{T}_{start}\}$
   1.3 $\forall (a, b) \in (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P}) : F'(a, b) = F(a, b)$
   1.4 for each marked place p and for each token in $M_0(p)$ we have p_{start}, t_{start} s.t. $p_{start}{}^{\bullet} = t_{start} \wedge t^{\bullet}_{start} = p$
   1.5 $\forall p_{start} \in \mathcal{P}_{start}$, ${}^{\bullet}p_{start} = \emptyset$
   1.6 let $M_0'(p) = 1$ for $p \in \mathcal{P}_{start}$ and $M_0'(p) = 0$ otherwise
2. construct $\mathfrak{O}'$
3. remove $\mathcal{P}_{start}, \mathcal{T}_{start}$ and their corresponding arcs

Definition 33. *Given a PN $\langle \mathcal{N}, M_0 \rangle$ and two branching processes $\mathfrak{B}, \mathfrak{B}'$ of PN $\langle \mathcal{N}, M_0 \rangle$ then $\mathfrak{B}' \sqsubseteq \mathfrak{B}$ if there exists an injective homomorphism $\psi : \mathfrak{B}' \to \mathfrak{B}$ s.t. $\varphi(\mathtt{min}(\mathfrak{B}')) = \mathtt{min}(\mathfrak{B})$ and $\phi \circ \varphi = \phi'$.*

There exists (up to an isomorphism) an unique maximum branching process (w.r.t. $\sqsubseteq$) that is the unfolding of $\langle \mathcal{N}, M_0 \rangle$ and is denoted $\mathcal{U}_{\mathcal{N}}(M_0)$ [McM93], [Esp94]. Then denote by $\mathcal{C}$ the set of all the configurations in $\mathcal{U}_{\mathcal{N}}(M_0)$.

Definition 34. *Given the unfolding $\mathcal{U}_{\mathcal{N}}(M_0)$ of a PN $\langle \mathcal{N}, M_0 \rangle$ and an event-node e then $\underline{C}(e) = (B_{\underline{C}(e)}, E_{\underline{C}(e)}, \preceq)$ is the minimal configuration that explains the execution of e where:*

1. $E_{\underline{C}(e)} = \{e' \in E : e' \preceq e\}$
2. $B_{\underline{C}(e)} = \big\{b \mid b \in \mathtt{min}_{\preceq}(\mathcal{U}_{\mathcal{N}}) \ \vee \ b \in {}^{\bullet}e'^{\bullet}$ *for some* $e' \in E_{\underline{C}(e)}\big\}$

As already mentioned unfoldings are usually *infinite* nets. As shown in [McM93] it is always possible to construct a finite *initial prefix* of the unfolding which captures its entire behavior by deriving the set of cut-off events. An event e is a cut-off event in the unfolding if there exists another event e' s.t. $i)$ $\phi(\underline{C}(e)) = \phi(\underline{C}(e'))$ and $ii)$ $\underline{C}(e') \sqsubset \underline{C}(e)$. The idea is that the continuations of $\mathcal{U}_{\mathcal{N}}(M_0)$ from $\underline{C}(e)$ and $\underline{C}(e')$ are isomorphic. Notice that the construction of the initial prefix as presented in [McM93] was improved in [KTKT95] and [ERV96] where is shown that a smaller initial prefix can be computed but this is beyond the purpose of this thesis.

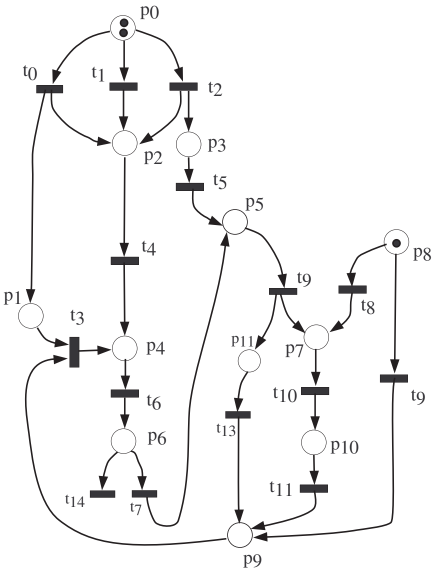


Figure 2.5:

Example 13. *Consider the PN displayed in Fig. 2.5. The unfolding net $\mathcal{U}_{\mathcal{N}}(M_0)$ is constructed as follows (see Fig. 2.6). First define the condition nodes that correspond to the tokens in M_0: b_8, b_0 and bb_0. Notice that for the sake of simplicity in notation we use for condition-nodes b_ι,bb_ι and for event nodes e_ι, ee_ι with upper indexes prime and double prime to represent conditions that correspond to a place p_ι respectively a transition t_ι.*

Then we have the initial configuration $C^\perp = (B_{C^\perp}, E_{C^\perp}, \preceq)$ where $B_{C^\perp} = \{b_8, b_0, bb_0\}$ and $E_{C^\perp} = \emptyset$.

For instance by appending e_8 to $C^\perp$ we obtain the configuration $C_1 = C^\perp \odot e_8$ where $C_1 = (B_{C_1}, E_{C_1}, \preceq)$ with $B_{C_1} = B_{C^\perp} \cup e_8^\bullet$ and $E_{C_1} = E_{C^\perp} \cup \{e_8\}$. For C_1 we have $CUT(C_1) = \{b_7, b_0, bb_0\}$ and $mark(C_1) = \phi(CUT(C_1))$, $mark(C_1) = \{(p_7, 1); (p_0, 2)\}$. Then in C_1 we have the set of enabled events:

$$ENABLED(C_1) = \{t_0, t_1, t_2, t_{10}\}$$

that is the set of transitions that are enabled in $\mathcal{N}$ in the marking $mark(C_1)$.

If e_0 is appended to $C^\perp$ we obtain the configuration $C_2 = C^\perp \odot e_0$ where $C_2 = (B_{C_2}, E_{C_2}, \preceq)$ with $B_{C_2} = B_{C^\perp} \cup e_0^\bullet$ and $E_{C_2} = E_{C^\perp} \cup \{e_0\}$. For C_2 we have $CUT(C_2) = \{b_8, b_1, b_2, bb_0\}$ and $mark(C_2) = \phi(CUT(C_2))$, $mark(C_2) = \{(p_8, 1); (p_1, 1); (p_2, 1); (p_0, 1)\}$. Then in C_2 we have the set of enabled events:

$$ENABLED(C_2) = \{t_0, t_1, t_2, t_4, t_8, t_9\}$$

that is the set of enabled transitions in the marking $mark(C_2)$ *in* $\mathcal{N}$.

From $CUT(C^{\perp}) = \{b_8, b_0, bb_0\}$ *all the enabled transitions are appended, in this example* $e_8, e_9, e_0, e_1, e_2, ee_0, ee_1, ee_2$ *as well as their output conditions.*

Notice that $e_8 \sharp e_9$, $e_0 \sharp e_1$, $e_1 \sharp e_2$, $e_0 \sharp e_2$, *and* $ee_0 \sharp ee_1$, $ee_1 \sharp ee_2$, $ee_0 \sharp ee_2$.

A prefix of the unfolding $\mathcal{U}_{\mathcal{N}}(M_0)$ *is displayed in Fig. 2.6 where the dotted lines emphasize this. Consider the node* e_6 *in* $\mathcal{U}_{\mathcal{N}}(M_0)$. *We have that* $\underline{C}(e_6) = (B_{\underline{C}(e_6)}, E_{\underline{C}(e_6)}, \preceq)$ *is the minimal configuration that explains the execution of* e_6 *where:*

1. $E_{\underline{C}(e_6)} = \{e_6, e_3, e_{11}, e_{10}, e_8, e_0\}$
2. $E_{B(e_6)} = \{b_8, b_0, bb_0, b_7, b_{10}, b_9, b_1, b_2, b_4, b_6\}$

The set of linearizations for $(E_{\underline{C}(e_6)}, \preceq)$ *is:*

$$\langle E_{\underline{C}(e_6)} \rangle_{\preceq} = \{e_0 e_8 e_{10} e_{11} e_3 e_6;\ \ e_8 e_0 e_{10} e_{11} e_3 e_6;\ \ e_8 e_{10} e_0 e_{11} e_3 e_6;\ \ e_8 e_{10} e_{11} e_0 e_3 e_6\}$$

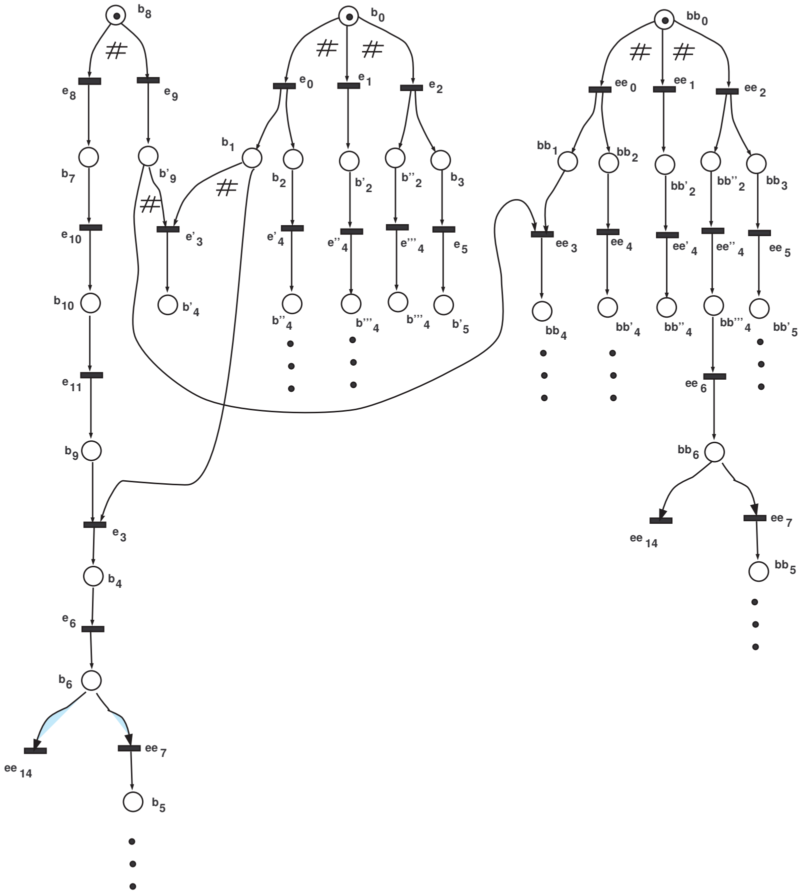

b_8
b_0
bb_0
#
#
#
#
#
e_8
e_9
e_0
e_1
e_2
ee_0
ee_1
ee_2
b_1
#
bb_1
bb_2
b_7
b'_9
#
b_2
b'_2
b''_2
b_3
bb'_2
bb''_2
bb_3
e_{10}
e'_3
e'_4
e''_4
e'''_4
e_5
ee_3
ee_4
ee'_4
ee''_4
ee_5
b_{10}
b'_4
b''_4
b'''_4
b'''_4
b'_5
bb_4
bb'_4
bb''_4
bb'''_4
bb'_5
e_{11}
ee_6
b_9
bb_6
e_3
ee_{14}
ee_7
b_4
bb_5
e_6
b_6
ee_{14}
ee_7
b_5

Figure 2.6:

2.6 Basics on modeling with DES

Generally speaking a DES is a system whose state changes with the occurrence of an event and not by the time flow. A DES model can easily be understood considering as an example the chess game. The initial state for this example is represented by the initial position of the pieces on the board. According to the chess game rules we have a set of legal moves that can be executed. Executing a legal move (event) from a position (state) results in a new position (state). In the new position the distribution of the pieces on the board allows in general for some new moves that were not legal previously while some moves that could have been executed in the previous position may become illegal in the new position.

The DES models arise from two different categories of systems. The first category includes the sequential concurrent processes (e.g. networks of computers, communication protocols, FMS, etc.) where the DES paradigm suits perfectly with the discrete nature of the dynamics of the process under investigation. The second category is represented by hybrid systems (that cover most of the real-world systems) where the behavior of a component is described both by continuous variables and by discrete variables. Abstract DES models (quantizing continuous variables to discrete variables [Lun00]) of these hybrid systems are often appropriate for designing efficient algorithms for solving fault diagnosis, control or optimization problems.

To illustrate the first category of DES models consider a device Dev that performs the following activities:

- initially Dev is $IDLE$ (ready for operation)
- at any time while Dev is $IDLE$ an unexpected $reset$ event may occur that makes Dev $UNAVAILABLE$ for operation
- when Dev is $UNAVAILABLE$ some repairing actions are taken and an event $recover$ brings back Dev in the $IDLE$ state
- when Dev is $IDLE$, some conditions that are not modeled in the DES model may provoke Dev to execute the event $alert$ that moves the device in the $ACTIVE$ state (the event *alert* will typically be synchronized with an event in the model of another component of the plant)
- from the $ACTIVE$ state, the device can $return$ to the $IDLE$ state (if some conditions are fulfilled by some other plant component) or it can operate by performing a certain $action$
- after operating (executing the *action* event), Dev moves to the $PAUSE$ state from where it $returns$ to the $IDLE$ state

The cyclic behavior of the above specified device can be represented by the deterministic automaton $G = (X, E, f, g, x_0)$ displayed in Fig. 2.7 where:

1. $X = \{UNAVAILABLE, IDLE, ACTIVE, PAUSE\}$
2. $E = \{reset, recover, return1, alert, action, return2\}$
3. the transition function f is a partial function defined as follows:

   $$\begin{aligned}&f(IDLE, reset) = UNAVAILABLE\\&f(IDLE, alert) = ACTIVE\\&f(UNAVAILABLE, recover) = IDLE\\&f(ACTIVE, return1) = IDLE\\&f(ACTIVE, action) = PAUSE\\&f(PAUSE, return2) = IDLE\end{aligned}$$

   not defined for the rest of domain

4. the active event function is:
   (a) $g(IDLE) = \{reset, alert\}$
   (b) $g(UNAVAILABLE) = \{recover\}$
   (c) $g(ACTIVE) = \{return1, action\}$
   (d) $g(PAUSE) = \{return2\}$
5. the initial state is $x_0 = IDLE$

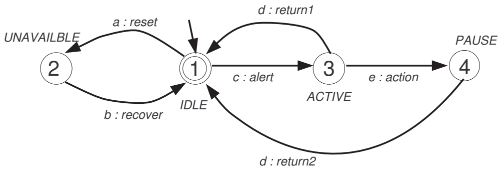


Figure 2.7:

Consider a labeling function $l : E \rightarrow \{a, b, c, d\}$. The events in G are deterministically labeled except for the two events $return1$ and $return2$ that have the same label d.

Now consider the case of two identical devices, modeled by two automata G_1 and G_2 having the same description as the device above discussed except the fact that when the event $action$ is executed, it is executed but only if it is simultaneously legal in both devices. To model this we bind up the two events named $action$ by a bar as displayed in Fig. 2.8. It means that the event $operation$ is executed jointly only if the state of both devices is

$ACTIVE$. The event $action$ is a synchronizing event and it defines an important operation on automata namely the synchronous product that allows for modeling and analysis of large and complex systems [CL99].

Taking a better look at the example presented in Fig. 2.8 one may see that the composition of the two automata G_1 and G_2 via the synchronizing event is structurally identical with the PN model displayed in Fig. 2.2.

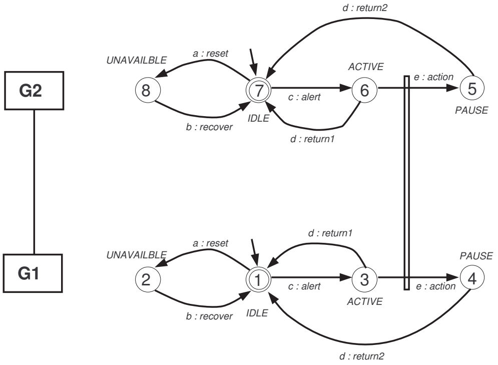


Figure 2.8:

This is natural since an automaton may be seen as a PN state-machine whose marking comprises a single place marked with one token. Thus behaviorally speaking the synchronous composition of G_1, G_2 is equivalent with the PN $\mathcal{N}$ displayed in Fig. 2.2 if the initial marking would be $M_0' = \{(p_1, 1), (p_7, 1)\}$ instead of $M_0 = \{(p_1, 2), (p_7, 1)\}$.

Notice that the presence of the second token in p_1 (see Fig. 2.2) makes the representation in terms of composition of automata more difficult. The main advantage of PNs is that it provides a more compact representation of the state space comparing with a model that results as a composition of automata. Notice moreover that not all Petri Nets can be translated in *finite-state* automata. Some PN of interest for applications may have unbounded markings.

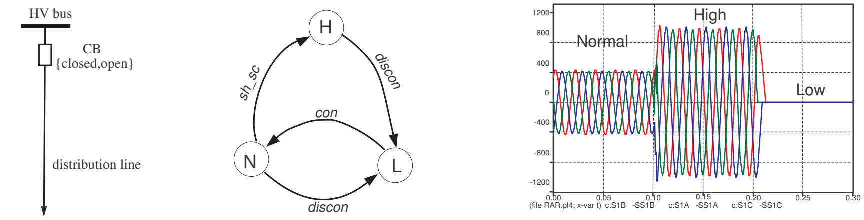


Figure 2.9:

DES models may also be obtained as abstractions of hybrid systems involving both continuous and discrete variables. To illustrate the modeling of a system with a hybrid dynamic consider the case of a power distribution line where at the supplying end of the line there is a circuit-breaker (CB) that connects and disconnects the line in case of emergency or in case that this is required by the plant operation. Assume that the current flow is the variable of interest. Even though the current flow is a continuous variable, for some purposes (say diagnosis, optimization) it can be discretized into a finite number of discrete-variable values e.g. as follows (see Fig. 2.9):

1. $NORMAL$ - when the r.m.s value of the current is in the normal range

2. $HIGH$ - when the r.m.s value of the current exceeds a threshold value, that jeopardizes the safe operation of the line

3. LOW - when the line is disconnected from the network and there is no power flow

The current flow state of the line changes discretely because of the occurrence of some events. E.g. when the state of the line is $NORMAL$ the occurrence of a short circuit (sh_sc) increases the current through the line and the new state of the line is $HIGH$. Then by disconnecting the circuit-breaker the line state changes to LOW as it is displayed in Fig. 2.9.

Notice first that in the above described example the time is not specifically considered. There is no indication how long an operation takes (e.g. which is the time required for the CB to open). The (untimed) model provides a high level (abstract) description of the logic plant behavior (e.g. the plant evolution can be described by a sequence of events but not the exact time of their occurrence).

Moreover the transient phenomena that accompany an abrupt change in the plant (a switch, a lightning strike, a short circuit, etc.) are not explicitly considered. This is because the plant behavior that is not relevant for the problem under consideration is not modeled. Moreover there is the assumption that the model is causally closed (closed world assumption) i.e. the occurrence of an event may have some unknown or irrelevant causes that are not modeled. For instance the unpredictable occurrence of a lightning strike that provokes a short-circuit is determined by the environment and not by the line model.

2.7 The distributed/modular analysis of the plant

For large systems the analysis cannot practically be performed monolithically. The reason is that that overall plant calculation becomes intractable because of the state space explosion. If we consider the plant composed by different components (sites, sub-systems), the overall plant behaviour (history) may be represented by a collection of individual histories and the overall plant state space analysis decomposes in the analysis of (consistent) local state spaces.

However the advantage of calculating local state spaces (of size a lot smaller then the size of the overall plant) may be *"compensated"* by a huge amount of information that has to be exchanged for checking the consistency of the local results.

Let $\mathcal{N}$ be the PN model of a large system. The flat overall plant model may be seen as composed by several components where each component is modeled by a PN $\mathcal{N}_i$, $i \in J$.

There are several composition rules for assembling the PN models $\mathcal{N}_i$ ($i \in J$):

- composition via common transitions [SR92]
- composition via common places [Val94], [GL03], [BFHJ03], [BJ04], [FBHJ05], [GL05], [JB05a]
- composition via guards (logical propositions) attached to some transitions, where the guards are defined over the marking of some places on the overall plant [BJ03a], [JB06].

Consider two local PN models $\mathcal{N}_i = (\mathcal{P}_i, \mathcal{T}_i, F_i)$ and $\mathcal{N}_j = (\mathcal{P}_j, \mathcal{T}_j, F_j)$. The composed PN model of $\mathcal{N}_i$ and $\mathcal{N}_j$ is $\mathcal{N} = (\mathcal{P}_i \cup \mathcal{P}_j, \mathcal{T}_i \cup \mathcal{T}_j, F_i \cup F_j)$ where $F_i \mid_{(\mathcal{P}_i \cap \mathcal{P}_j) \times (\mathcal{T}_i \cap \mathcal{T}_j)} = F_j \mid_{(\mathcal{P}_i \cap \mathcal{P}_j) \times (\mathcal{T}_i \cap \mathcal{T}_j)}$.

We say that $\mathcal{N}_i$ and $\mathcal{N}_j$ are:

transition-bordered if $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ and $\mathcal{T}_i \cap \mathcal{T}_j \neq \emptyset$

place-bordered if $\mathcal{P}_i \cap \mathcal{P}_j \neq \emptyset$ and $\mathcal{T}_i \cap \mathcal{T}_j = \emptyset$

The intuitive interpretation of the shared (common) part of two PN models is that it represents the interactions between two components of the overall plant.

For transition-bordered nets, the shared transitions *synchronize* parts of the components i.e. the execution of a shared transition takes place simultaneously in both components.

On the other hand, in *place-bordered* nets the interaction is different since for untimed models it is required that the number of tokens that enter a local model (say $\mathcal{N}_i$) via a border place p_{IN_i} *is smaller* than the number of tokens that could leave the corresponding local PN model that has p_{IN_i} as a border place.

Consider again the case of two PN models $\langle \mathcal{N}_i, M_{0_i} \rangle$ and $\langle \mathcal{N}_j, M_{0_j} \rangle$ and their composed PN model $\langle \mathcal{N}, M_0 \rangle$, i.e. $\mathcal{N} = \mathcal{N}_i \cup \mathcal{N}_j$ and $M_0 = M_{0_i} \uplus M_{0_j}$ (where $\uplus$ denotes the union with addition of two multi-sets and a marking is considered a multi-set of tokens). Then we have that:

> a feasible trace $\tau \in \mathcal{L}_{\mathcal{N}}(M_0)$ in $\langle \mathcal{N}, M_0 \rangle$ is a pair $\tau = (\tau_i, \tau_j)$ of local traces τ_i resp. τ_j, where τ_i resp. τ_j are the projection of τ on to $\mathcal{T}_i$ respectively $\mathcal{T}_j$ ($\Pi_{\mathcal{T}_i}(\tau) = \tau_i$ and $\Pi_{\mathcal{T}_j}(\tau) = \tau_j$)
>
> a reachable marking $M \in \mathcal{R}_{\mathcal{N}}(M_0)$ in $\langle \mathcal{N}, M_0 \rangle$ is a pair $M = (M_i, M_j)$ of local markings M_i resp. M_j, where M_i resp. M_j are the subvectors of M that correspond with the set of places $\mathcal{P}_i$ respectively $\mathcal{P}_j$.

The key issue of a modular/distributed algorithm is that instead of a monolithic analysis of the plant, firstly each component is analyzed in *isolation* and then it is checked to have a consistent behaviour with all the other components.

The analysis of each component in *isolation* gives rise to a major difficulty namely the analysis of a PN (the PN model of the component) with incomplete knowledge on the marking of the border places (i.e. the number of the tokens that enter a component via the common places is unknown).

For instance consider two PNs displayed in Fig. 2.10 where $\mathcal{N}_i$ and $\mathcal{N}_j$ are place-bordered nets that model two components of a plant. The interactions between the components are represented by the common places $\mathcal{P}_{IN_i} = \mathcal{P}_{OUT_j}$ and $\mathcal{P}_{IN_j} = \mathcal{P}_{OUT_i}$.

Thus we have that tokens from $\mathcal{N}_i$ can leave via the output places $\mathcal{P}_{OUT_i}$ and enter $\mathcal{N}_j$ via the input place $\mathcal{P}_{IN_j}$ and similarly tokens from $\mathcal{N}_j$ can leave via the output place $\mathcal{P}_{OUT_j}$ and enter $\mathcal{N}_i$ via the input places $\mathcal{P}_{IN_i}$.

Consider the analysis of component i in *isolation*. The main difficulty is that *via* the input places $\mathcal{P}_{IN_i} = \{p^1_{IN_i}, \ldots, p^m_{IN_i}\}$ tokens can enter component i from component j.

In this case we are faced with the analysis of a PN model whose initial marking is partially unknown i.e. it is known the number of tokens that are in the places that are in $\mathcal{N}_i$ and are marked in the overall initial marking M_0 but *via* $\mathcal{P}_{IN_i}$ an arbitrary number of tokens can enter $\mathcal{N}_i$.

Consider for $\mathcal{N}_i$ two initial markings M'_{0_i} and M''_{0_i} s.t. $\forall p \in \mathcal{P}_i \setminus \{\mathcal{P}_{IN_i}\}$

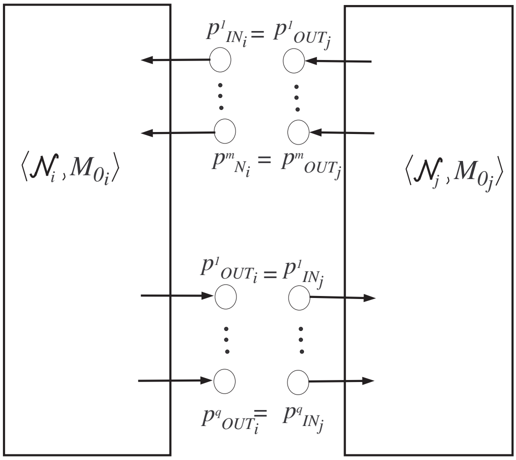

$p^{1}_{IN_i} = p^{1}_{OUT_j}$
$p^{m}_{N_i} = p^{m}_{OUT_j}$
$\langle \mathcal{N}_i, M_{0i} \rangle$
$\langle \mathcal{N}_j, M_{0j} \rangle$
$p^{1}_{OUT_i} = p^{1}_{IN_j}$
$p^{q}_{OUT_i} = p^{q}_{IN_j}$

Figure 2.10:

we have that $M'_{0_i}(p) = M''_{0_i}(p)$ and $\forall p_{IN_i} \in \mathcal{P}_{IN_i}$, $M'_{0_i}(p_{IN_i}) \leq M''_{0_i}(p_{IN_i})$.

By Proposition 2 we have that $\mathcal{L}_{\mathcal{N}_i}(M'_{0_i}) \subseteq \mathcal{L}_{\mathcal{N}_i}(M''_{0_i})$ that intuitively may be understood as: *the more tokens enter component i PN model $\mathcal{N}_i$, the more local traces in $\mathcal{N}_i$ are possible.*

Then denote by $\mathcal{M}_{\mathcal{N}_i}(M'_{0_i})$ and $\mathcal{M}_{\mathcal{N}_i}(M''_{0_i})$ the set of markings that can be obtained by firing the traces considered by $\mathcal{L}_{\mathcal{N}_i}(M'_{0_i})$ and $\mathcal{L}_{\mathcal{N}_i}(M''_{0_i})$ respectively:

$$\mathcal{M}_{\mathcal{N}_i}(M'_{0_i}) = \left\{ M'_i \mid \exists \tau'_i \in \mathcal{L}_{\mathcal{N}_i}(M'_{0_i}) \ \text{ s.t. } \ M'_{0_i} \xrightarrow{\tau'_i} M'_i \right\}$$

$$\mathcal{M}_{\mathcal{N}_i}(M''_{0_i}) = \left\{ M''_i \mid \exists \tau''_i \in \mathcal{L}_{\mathcal{N}_i}(M''_{0_i}) \ \text{ s.t. } \ M''_{0_i} \xrightarrow{\tau''_i} M''_i \right\}$$

Since $\mathcal{L}_{\mathcal{N}_i}(M'_{0_i}) \subseteq \mathcal{L}_{\mathcal{N}_i}(M''_{0_i})$ we have that $\forall M' \in \mathcal{M}_{\mathcal{N}_i}(M'_{0_i})$, $\exists M'' \in \mathcal{M}_{\mathcal{N}_i}(M''_{0_i})$ s.t. $M' \leq M''$. This result can be intuitively understood as: *the more tokens enter $\mathcal{N}_i$ via the input border places $\mathcal{P}_{IN_i}$, the more tokens can leave $\mathcal{N}_i$ via the output border places $\mathcal{P}_{IN_i}$.*

For M'_{0_i} and M''_{0_i} defined above denote $\mathcal{U}'_{\mathcal{N}_i}(M'_{0_i})$ and $\mathcal{U}''_{\mathcal{N}_i}(M''_{0_i})$ the net unfolding of $\langle \mathcal{N}_i, M'_{0_i} \rangle$ respectively the net unfolding of $\langle \mathcal{N}_i, M''_{0_i} \rangle$. Denote $\Delta M_{0_i} = M''_{0_i} - M'_{0_i}$ the difference between the two initial markings. Notice that $\Delta M_{0_i}(p) \neq 0 \Rightarrow p \in \mathcal{P}_{IN_i}$.

Under a proper labeling we have that $\mathcal{U}''_{\mathcal{N}_i}(M''_{0_i}) \sqsubseteq \mathcal{U}'_{\mathcal{N}_i}(M'_{0_i})$. Denote by $\Delta \mathcal{U}_{\mathcal{N}_i}(\Delta M_{0_i})$ the sub-net of $\mathcal{U}''_{\mathcal{N}_i}(M''_{0_i})$ that comprises events that are not included in $\mathcal{U}'_{\mathcal{N}_i}(M'_{0_i})$:

$$\Delta\mathcal{U}_{\mathcal{N}_i}(\Delta M_{0_i}) = \mathcal{U}''_{\mathcal{N}_i}(M''_{0_i}) \setminus \mathcal{U}'_{\mathcal{N}_i}(M'_{0_i})$$

Denote by $\Delta B(IN_i)$ and $\Delta B(OUT_i)$ the set of input-border conditions respectively the set of output-border conditions in $\Delta\mathcal{U}_{\mathcal{N}_i}(\Delta M_{0_i})$:

$$\Delta B(IN_i) = \{b \in \Delta\mathcal{U}_{\mathcal{N}_i}(\Delta M_{0_i}) : \phi(b) \in \mathcal{P}_{IN_i}\}$$

$$\Delta B(OUT_i) = \{b \in \Delta\mathcal{U}_{\mathcal{N}_i}(\Delta M_{0_i}) : \phi(b) \in \mathcal{P}_{OUT_i}\}$$

We have that $\forall b_{OUT_i} \in \Delta B(OUT_i) \Rightarrow \exists b_{IN_i} \in \Delta B(IN_i)$ s.t. $b_{IN_i} \preceq b_{OUT_i}$ that simply says that the extra-tokens that can be produced at the output border places $\mathcal{P}_{OUT_i}$ are consequences (predecessors) of the extra-tokens (ΔM_{0_i}) that are assumed in the initial marking at the input border places $\mathcal{P}_{IN_i}$ of component i.

Consider in the overall model $\mathcal{N}$ an initial marking M_0 and assume for simplicity that the border places are not marked in M_0 i.e. $\forall p \in \mathcal{P}_{OUT_i} \cup \mathcal{P}_{IN_i}$, $M_0(p) = 0$. Denote by M_{0_i} and M_{0_j} the local initial markings of component i, respectively component j ($M_{0_i}(p) = M_0(p)$ for $p \in \mathcal{P}_i$ and $M_{0_j}(p) = M_0(p)$ for $p \in \mathcal{P}_j$). Then consider M'_{0_i} s.t. $M_{0_i} < M'_{0_i}$ and $M_{0_i}(p) \neq M'_{0_i}(p) \Rightarrow p \in \mathcal{P}_{IN_i}$ and denote by $\mathcal{C}_i(M'_{0_i})$ the set of configurations in the net unfolding $\mathcal{U}_{\mathcal{N}_i}(M'_{0_i})$. Similarly, for component j denote by $\mathcal{C}_j(M'_{0_j})$ the set of configurations in the net unfolding $\mathcal{U}_{\mathcal{N}_j}(M'_{0_j})$

For a local configuration $C'_{\nu_i} \in \mathcal{C}_i(M'_{0_i})$ denote by $B'_{C'_{\nu_i}}(IN_i)$ the set of input-border conditions of configuration C'_{ν_i}, and denote by $B'_{C'_{\nu_i}}(OUT_i)$ the set of output-border conditions of configuration C'_{ν_i}:

$$B'_{C'_{\nu_i}}(IN_i) = \left\{b_{\nu_i} \in B_{C'_{\nu_i}} : \phi(b_{\nu_i}) \in \mathcal{P}_{IN_i}\right\}$$

$$B'_{C'_{\nu_i}}(OUT_i) = \left\{b_{\nu_i} \in B_{C'_{\nu_i}} : \phi(b_{\nu_i}) \in \mathcal{P}_{OUT_i}\right\}$$

Similarly for a local configurations $C'_{\nu_j} \in \mathcal{C}_j(M'_{0_j})$ denote by $B'_{C'_{\nu_j}}(IN_j)$ and $B'_{C'_{\nu_j}}(OUT_j)$ the set of input-border conditions, respectively the set of output-border conditions of configuration C'_{ν_j}:

$$B'_{C'_{\nu_j}}(IN_j) = \left\{b_{\nu_j} \in B_{C'_{\nu_j}} : \phi(b_{\nu_j}) \in \mathcal{P}_{IN_j}\right\}$$

$$B'_{C'_{\nu_j}}(OUT_j) = \left\{b_{\nu_j} \in B_{C'_{\nu_j}} : \phi(b_{\nu_j}) \in \mathcal{P}_{OUT_j}\right\}$$

For simplicity consider that the set of input-border places $\mathcal{P}_{IN_i}$ and the set of output border places $\mathcal{P}_{OUT_i}$ are disjoint ($\mathcal{P}_{IN_i} \cap \mathcal{P}_{OUT_i} = \emptyset$).

Given two local configurations C'_{ν_i} and C'_{ν_j} we say that C'_{ν_i} and C'_{ν_j} are *possibly consistent* if:

$$\begin{aligned} &\forall p \in \mathcal{P}_{IN_i} \quad \mid B_{C'_{\nu_i}}(p) \mid \;\leq\; \mid B_{C'_{\nu_j}}(p) \mid \quad \text{and} \\ &\forall p \in \mathcal{P}_{IN_j} \quad \mid B_{C'_{\nu_j}}(p) \mid \;\leq\; \mid B_{C'_{\nu_i}}(p) \mid \end{aligned} \tag{2.7}$$

where $B_{C'_{\nu_i}}(p) = \left\{ b_\nu \in B_{C'_{\nu_i}} : \phi(b_{\nu_i}) = p \right\}$ and $\mid B_{C'_{\nu_i}}(p) \mid$ denotes the cardinality of the set $\mid B_{C'_{\nu_i}}(p) \mid$.

In words C'_{ν_i} and C'_{ν_j} are possibly consistent (and thus their composition may result in a global configuration) if for any place of the common border the number of input-border conditions is smaller than or equal to the number of output border conditions, i.e. the number of tokens that enter a component is smaller than or equal with the number of tokens that left the component j.

Notice that we say that C'_{ν_i} and C'_{ν_j} are *possibly consistent* since the inequality 2.7 is a necessary but not sufficient condition for C'_{ν_i} and C'_{ν_j} to make a global configuration.

Given a pair of local configurations (C'_{ν_i}, C'_{ν_j}) that are *possibly consistent* denote by $\psi_{l_{\nu_i \nu_j}}$ the interpretation function of the border conditions of C'_{ν_i} and C'_{ν_j}:

$$\psi_{l_{\nu_i \nu_j}} : B_{C'_{\nu_i}}(IN_i) \cup B_{C'_{\nu_j}}(IN_j) \rightarrow B_{C'_{\nu_j}}(OUT_j) \cup B_{C'_{\nu_i}}(OUT_i)$$

where:

1. $\psi_{l_{\nu_i \nu_j}}$ is injective
2. $\psi_{l_{\nu_i \nu_j}}(b_{IN_i}) \in B_{C'_{\nu_j}}(OUT_j)$
3. $\psi_{l_{\nu_i \nu_j}}(b_{IN_j}) \in B_{C'_{\nu_i}}(OUT_i)$

Denote by $C'_{l_{\nu_i \nu_j}}$ the PN that is obtained from the local configurations C'_{ν_i},C'_{ν_j} by merging their border conditions according with the interpretation function $\psi_{l_{\nu_i \nu_j}}$.

We have that if $C'_{l_{\nu_i \nu_j}}$ is acyclic and every input border-conditions in C'_{ν_i} and C'_{ν_j} has one input event in $C'_{l_{\nu_i \nu_j}}$ then C'_{ν_i} and C'_{ν_j} under the interpretation $\psi_{C'_{l_{\nu_i \nu_j}}}$ (denoted $(C'_{\nu_i}, C'_{\nu_j}, \psi_{\nu_i \nu_j})$) is a global configuration in the unfolding $\mathcal{U}_{\mathcal{N}}(M_0)$ of the overall PN $\langle \mathcal{N}, M_0 \rangle$.

Consider a global configuration $C_v \in \mathcal{C}$, $C_v = (B_{C_v}, E_{C_v}, \preceq)$ in the net unfolding $\mathcal{U}_{\mathcal{N}}(M_0)$ of the global net $\langle \mathcal{N}, M_0 \rangle$. Denote by $B^i_{C_v}$ respectively $E^i_{C_v}$ the set of conditions respectively the set of events of C_ν that corresponds with $\mathcal{N}_i$:

$$B^i_{C_v} = \left\{ b \in B^i_{C_v} : \phi(b) \in \mathcal{P}_i \right\}$$

$$E^i_{C_v} = \{e \in E^i_{C_v} : \phi(e) \in \mathcal{T}_i\}$$

and denote by C^i_v the subnet of C_v that correspond with the set of conditions $B^i_{C_v}$ and the set of events $E^i_{C_v}$.

We have that if $(C'_{\nu_i}, C'_{\nu_j}, \psi_{l_{\nu_i \nu_j}})$ is consistent then there exists a global configuration $C_v \in \mathcal{C}$ s.t. $C^i_v = C'_{\nu_i}$ and $C^j_v = C'_{\nu_j}$.

Thus given two arbitrary initial markings M'_{0_i} and M'_{0_j} we can derive global configurations as pairs of local configurations under some interpretations.

Notice that to check the consistency of a pair of local configurations requires not only the information about how many input and output border-conditions are considered in a local configuration but also how the input border-conditions and the output border-conditions are related in a local configuration.

Remark 4. *Beside the computational advantage of using the net unfolding technique for making the local calculations there is also the advantage that in a configuration the causality relations between the tokens and the transitions are explicitly represented. This allows for an easy extraction of the causality between the tokens that enter and leave a local model, information that is required for checking the consistency between the local configurations. Notice that if we derive the local plant behavior as a set of local traces (and not local configurations) then there is a major difficulty to check the consistency between the local traces since inequality 2.7 (with $B_{C'_{\nu_i}}(p)$ understood as the marking $M_{0_{\nu_i}}(p)$) is a necessary but not sufficient condition for checking the consistency of two local traces.*

Consider again the analysis of each component (local site) in isolation. Intuitively to derive the set of all global configurations as pairs of local configurations one should consider for a component i an initial marking M'_{0_i} s.t. M'_{0_i} considers for each input place $p_{IN_i} \in \mathcal{P}_{IN_i}$ the maximum number of tokens that can enter to $\mathcal{N}_i$ via p_{IN_i}. But this requires the calculation of all the legal traces of the global model something we want to avoid by designing a modular/distributed algorithm. Moreover the preliminary calculation of a component may have the same computational complexity as the overall plant calculation [Val94] that further requires a large amount of information to be exchanged for discarding the local configurations that are not consistent.

Notice that even if some *a priori* knowledge allows for deriving the maximum number of tokens that can enter to $\mathcal{N}_i$ via p_{IN_i} without making the calculation of the global net this may not be of any help if the plant structure changes often in the sense that some components are plugged in and removed.

Even though this method is hardly feasible for real applications it has some advantages. In the first place the calculations are simple and require only once to check the consistency of the local calculations that in other words means a single communication round between two components. Secondly before checking the global consistency the preliminary calculation of a component is an over-approximation of its behaviour.

The second solution is to incrementally construct the overall net unfolding as a composition of local net unfoldings as follows:

1. for each component i consider the known initial marking M_{0_i} and calculate $\mathcal{U}_{\mathcal{N}_i}(M_{0_i})$ and then repeat the following steps:
   (a) derive the set all global consistent pairs (C_{ν_i}, C_{ν_j})
   (b) then for each global consistent pair $C_\nu = (C_{\nu_i}, C_{\nu_j}, \psi_{l_{\nu_i \nu_j}})$ calculate the extensions of C_ν in $\mathcal{N}_i$ and $\mathcal{N}_j$
   (c) repeat (a) and (b)

This method looks more feasible than the previous method since it does not require *a priori* knowledge of the plant behaviour, basically the overall plant unfolding is incrementally constructed by extending in each local model the local configurations. Moreover the information that is exchanged is minimal in the sense that each *piece* of information that is exchanged is used for deriving global consistent configurations (i.e. the agents do not exchange information for discarding some local configurations but to extend a global configuration derived at the previous step).

As already mentioned one of the objectives of this thesis is to design online distributed algorithms for fault detection and diagnosis. Anticipating the presentation that follows, consider that for each component i ($i \in J$) there is an agent Ag_i that makes the local calculations and there is a communication channel between the agents that allow to exchange limited information.

Assume that each model of a component includes a non-empty set of observable transitions and consider that the local agents construct the overall plant unfolding in an incremental manner as presented above. Moreover consider that all the places that correspond with the PN model of component i are not marked in the initial marking of the overall plant and that the communication between the local sites is unavailable. In this case the local agent Ag_i does not compute anything since the initial marking of the local site i is $M_{0_i} = 0$.

Then consider that Ag_i observes that an (observable) event (e.g. $t_i^o \in \mathcal{T}_{o_i}$) was executed in component i. Since the observation is correct it means that some tokens entered component i via the the input places $\mathcal{P}_{IN_i}$ so that the event that was observed has become possible to be executed.

Thus it looks absolutely reasonable to design a distributed algorithm s.t. the local agent makes some calculations even though it has no knowledge about the number of the tokens that could have entered in $\mathcal{N}_i$ via the input places $\mathcal{P}_{IN_i}$. The idea is simple. For the sake of presentation consider that the observation is deterministic (i.e. each observable transition emits a distinct label). Since the observation of a component is correct Ag_i is sure that the input places $p \in {}^\bullet t_i^o$ of the observed transition t_i^o have been marked before the transition t_i^o was executed. Then by inferring backwards agent Ag_i can derive the unobservable events that must have been executed before providing the tokens in $p \in {}^\bullet t_i^o$ that made the observation of t_i^o possible. Repeating the same backward inference Ag_i can derive a set of minimal (preliminary) explanations in the model of its component that explain the local observation. A local (preliminary) explanation includes the unobservable events whose occurrence made possible the observation as well as the assumption that (a minimal) number of tokens have entered from the neighboring component(s).

The main advantage of this method is that a local agent can derive a preliminary local calculation that under some technical conditions allows for taking local control actions when the communication between the local agents is not possible. When the communication between the agents is allowed they exchange limited information to check the consistency of their preliminary results and also to generate the set of complete explanations.

Chapter 3

The analysis of PN models under partial observation

In this chapter we present two methods for the state estimation of untimed Petri Net models under partial observation.

First we present in Section 3.1 the standard algorithm for constructing a classical observer-automaton of a given Petri Net model. The method is similar to the construction of a classical observer for DES modeled as automata [OW90]. When analyzing large systems under partial observation the use of a classical observer-automaton is hardly possible because of its high spatial complexity (exponential in the number of places [OW90]). Basically a state of an observer-automaton includes all the possible states (markings) the plant can be in after observing a string of observable events.

To overcome this limitation we propose in Section 3.2 the construction of a reduced (resource-aware) observer automaton that contains in a given state fewer markings than the classical observer-automaton.

3.1 Classical observer

Consider a PN model $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ with $\mathcal{T} = \mathcal{T}_o \cup \mathcal{T}_{uo}$. Then given an arbitrary marking M denote by $UR_{\mathcal{N}}(M)$ the unobservable reach of M that is the set of markings that can be obtained starting from M and firing only strings of unobservable transitions:

$$UR_{\mathcal{N}}(M) = \left\{ M' \mid \exists \sigma_{uo} \in \mathcal{T}_{uo}^{*} \text{ s.t. } M \xrightarrow{\sigma_{uo}} M' \right\} \tag{3.1}$$

For a set of markings $\mathcal{M}$, define:

$$UR_{\mathcal{N}}(\mathcal{M}) = \bigcup_{M \in \mathcal{M}} UR_{\mathcal{N}}(M) \tag{3.2}$$

Recall that $\mathcal{R}_{\mathcal{N}}(M_0)$ is the set of the reachable markings of $\langle \mathcal{N}, M_0 \rangle$ and let $Pwr(\mathcal{R}_{\mathcal{N}}(M_0))$ be the set of all the subsets of $\mathcal{R}_{\mathcal{N}}(M_0)$.

Consider a PN $\mathcal{N}$ whose initial marking M_0 and the partition of the transition set $\mathcal{T}$ into the disjoint sub-sets of observable transitions $\mathcal{T}_o$ and unobservable transitions $\mathcal{T}_{uo}$ are completely known.

The classical observer-automaton $\texttt{CO}(\langle \mathcal{N}, M_0 \rangle)$ of the partial observable PN model $\langle \mathcal{N}, M_0 \rangle$, $\mathcal{T} = \mathcal{T}_o \cup \mathcal{T}_{uo}$ is:

$$\texttt{CO}(\langle \mathcal{N}, M_0 \rangle) = (X_{co}, E_{co}, f_{co}, x_0^{co}, \varrho_{co})$$

where:

- X_{co} is the set of states of $\texttt{CO}(\langle \mathcal{N}, M_0 \rangle)$
- $\varrho_{co} : X_{co} \to Pwr(\mathcal{R}_{\mathcal{N}}(M_0))$ is a function that associates to each state $x_{co} \in X_{co}$ a set of reachable markings $\varrho_{co}(x_{co}) \in Pwr(\mathcal{R}_{\mathcal{N}}(M_0))$
- E_{co} is the set of events of the classical observer $\texttt{CO}(\langle \mathcal{N}, M_0 \rangle)$. $E_{co} = \Omega_o$ if the observation is non-deterministic and $E_{co} = \mathcal{T}_o$ if the observation is deterministic.
- $\varrho_{co}(x_0^{co}) = UR_{\mathcal{N}}(M_0)$ is the set of markings in $\langle \mathcal{N}, M_0 \rangle$ estimated in the initial state of the classical observer $\texttt{CO}(\langle \mathcal{N}, M_0 \rangle)$
- $f_{co} : X_{co} \times E_{co}^* \to X_{co}$ is the transition function of $\texttt{CO}(\langle \mathcal{N}, M_0 \rangle)$ that is defined as follows:

  for $x_\iota^{co} \in X_{co}$ a state of $\texttt{CO}(\langle \mathcal{N}, M_0 \rangle)$ and a string of observable transitions $\sigma \in E_{co}^*$ we have: $f_{co}(x_0^{co}, \sigma) = x_\iota^{co}$ if $\varrho_{co}(x_\iota^{co}) \neq \emptyset$ where:

  $$\varrho_{co}(x_\iota^{co}) = \left\{ M_\iota : M_0 \xrightarrow{\tau} M_\iota \wedge \Pi_{\mathcal{T}_o}(\tau) = \sigma \right\}$$

Given the PN $\langle \mathcal{N}, M_0 \rangle$ and $\texttt{CO}(\langle \mathcal{N}, M_0 \rangle)$ its classical observer-automaton, we have that:

$$\mathcal{L}_{\texttt{CO}} = \Pi_{\mathcal{T}_o}(\mathcal{L}_{\mathcal{N}}(M_0)) \tag{3.3}$$

if the observation is deterministic or $\mathcal{L}_{\texttt{CO}} = \mathcal{L}_{\mathcal{N}}^{\ell}(M_0)$ otherwise.

Example 14. *Consider the PN $\langle \mathcal{N}, M_0 \rangle$ displayed in Fig. 3.1-left. The initial marking is $M_0 = \{M(p_1) = 2; M(p_2) = 2;\}$ and the only observable transition is t_3 ($\mathcal{T}_o = \{t_3\}$).*

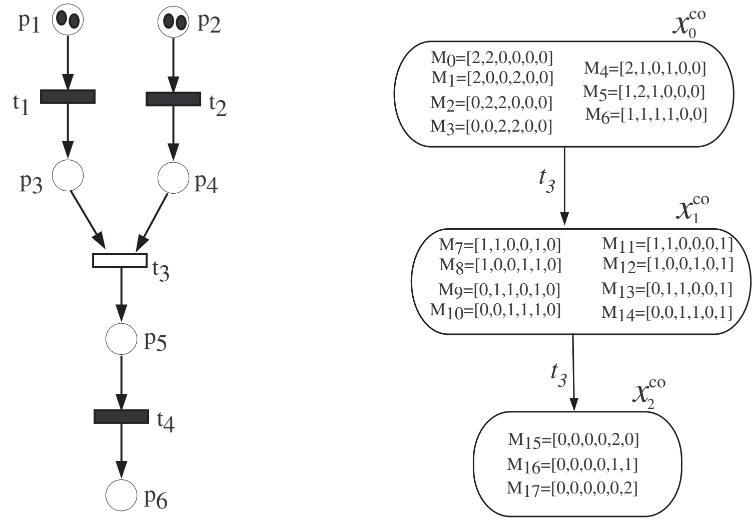


Figure 3.1:

For this very simple example we have in Fig. 3.1-right the classical observer automaton $\mathtt{CO}(\langle\mathcal{N}, M_0\rangle)$ *that comprises three states* $X_{co} = \{x_0^{co}, x_1^{co}, x_2^{co}\}$ *and two events both corresponding with the execution of the only observable transition* t_3*.*

Then $\varrho_{co}(x_0^{co}) = UR_{\mathcal{N}}(M_0)$ *comprises 7 markings* $\{M_\iota \mid \iota = 0, \dots 6\}$, $\varrho_{co}(x_1^{co})$ *comprises 8 markings* $\{M_\iota \mid \iota = 7, \dots 14\}$ *while* $\varrho_{co}(x_2^{co})$ *comprises 3 markings* $\{M_\iota \mid \iota = 15, \dots 17\}$*.*

We have that:

1. $M \in \varrho_{co}(x_0^{co}) \Leftrightarrow (M_0 \xrightarrow{\sigma} M) \wedge (\Pi_{\mathcal{T}_o}(\sigma') = \epsilon)$
2. $M' \in \varrho_{co}(x_1^{co}) \Leftrightarrow (M_0 \xrightarrow{\sigma'} M') \wedge (\Pi_{\mathcal{T}_o}(\sigma') = t_3)$
3. $M'' \in \varrho_{co}(x_2^{co}) \Leftrightarrow (M_0 \xrightarrow{\sigma'} M'') \wedge (\Pi_{\mathcal{T}_o}(\sigma'') = t_3 t_3)$

3.2 Reduced observer

As mentioned above the use of the classical observer-automaton becomes computationally unfeasible because its size grows exponentially in the number of places $\mid \mathcal{P} \mid$. In order to overcome this limitation we propose in the following the construction of a reduced observer-automaton that represents the same language as the classical observer-automaton but includes in a given state fewer markings then the classical one.

To illustrate the rationale behind the construction of a reduced observer-automaton ($\mathtt{RO}$) consider a state $x_\iota^{co} \in X_{co}$ of the classical observer-automaton

$\mathtt{CO}(\langle\mathcal{N}, M_0\rangle)$ and then let $\mathcal{M}'(x_\iota^{co})$ be a subset of the set of markings $\varrho_{co}(x_\iota^{co})$ that is estimated by the classical observer-automaton for the state x_ι^{co} such that $UR_\mathcal{N}(\mathcal{M}'(x_\iota^{co})) = \varrho_{co}(x_\iota^{co})$.

We call $\mathcal{M}'(x_\iota^{co})$ a base for $\varrho_{co}(x_\iota^{co})$ if $UR_\mathcal{N}(\mathcal{M}'(x_\iota^{co})) = \varrho_{co}(x_\iota^{co})$ and say that $\mathcal{M}'(x_\iota^{co})$ is a minimal base of $\varrho_{co}(x_\iota^{co})$ if:

$$\forall \mathcal{M}''(x_\iota^{co}) \subseteq \mathcal{M}'(x_\iota^{co}), UR(\mathcal{M}''(x_\iota^{co})) = \mathcal{M}'(x_\iota^{co}) \Rightarrow \mathcal{M}''(x_\iota^{co}) = \mathcal{M}'(x_\iota^{co}).$$

Example 15. *Consider the classical observer automaton derived in Example 14 and consider the initial state x_0^{co} in $\mathtt{CO}(\langle\mathcal{N}, M_0\rangle)$. We have that M_0 is the minimal base for $\varrho_{co}(x_0^{co})$ since $UR_\mathcal{N}(M_0) = \varrho_{co}(x_0^{co})$. For x_1^{co} we have that M_7 is the minimal base for $\varrho_{co}(x_1^{co})$ since $UR_\mathcal{N}(M_7) = \varrho(x_1^{co})$ and then for x_2^{co} we have that M_{15} is the minimal base for $\varrho_{co}(x_2^{co})$ since $UR_\mathcal{N}(M_{15}) = \varrho_{co}(x_2^{co})$.*

Definition 35. $\mathtt{RO}(\langle\mathcal{N}, M_0\rangle) = (X_{ro}, E_{ro}, f_{ro}, x_{ro_0}, \varrho_{ro})$ *is a reduced observer-automaton of the PN $\langle\mathcal{N}, M_0\rangle$ if:*

$X_{ro} = X_{co}$

$E_{ro} = E_{co}$

$f_{ro} = f_{co}$

$\forall x_\iota \in X_{ro}, \varrho_{ro}(x_\iota) \subseteq \varrho_{co}(x_\iota) \wedge UR_\mathcal{N}(\varrho_{ro}(x_\iota)) = \varrho_{co}(x_\iota)$

$f_{ro}(x_0^{ro}, \sigma) = x_\iota^{ro} \Rightarrow \forall M_\iota \in \varrho_{ro}(x_\iota^{ro}), \exists \tau \in \mathcal{L}_\mathcal{N}(M_0)$ *s.t.* $M_0 \xrightarrow{\sigma} M_\iota$ *and* $\Pi_{\mathcal{T}_o}(\tau) = \sigma$

Thus $\mathtt{RO}(\langle\mathcal{N}, M_0\rangle)$ is an observer-automaton having the same structure as $\mathtt{CO}(\langle\mathcal{N}, M_0\rangle)$ the only difference being that a state x_ι in the reduced observer-automaton contains a set of markings $\varrho_{ro}(x_\iota)$ that is a base of the set of markings $\varrho_{co}(x_\iota^{co})$ that is considered by the corresponding state x_ι^{co} in the classical observer-automaton $\mathtt{CO}(\langle\mathcal{N}, M_0\rangle)$.

Example 16. *For the PN in Fig. 3.1 the (minimal) reduced observer-automaton is displayed in Fig. 3.2.*

Assume in the following that the size of the plant under investigation is large so that the classical observer would have almost the same size as the PN model. Moreover assume that changes of the plant structure (i.e. changes in $\mathcal{T}_o$) are possible to take place with some regularity. It means that for such plant the use of off-line derived observers is hardly possible. On the other hand the current computational capabilities allow for fast on-line calculations.

Consider then that the on-line observer (classical or reduced) operates as follows:

1. the on-line (classical or reduced) observer is in the initial state x_0.

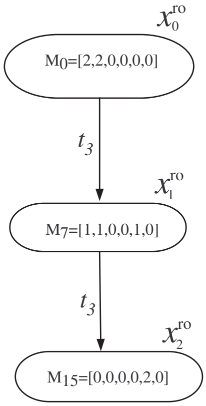


Figure 3.2:

2. one observable event (transition t^o) is executed in the plant (we assume that no two observable events are executed exactly at the same time) and the sensor associated with t^o immediately informs the supervisory system (we assume the sensor output never lost and never delayed).

3. a new state of the observer is calculated by enumerating the set of possible markings the plant can be in after observing the execution of t^o.

4. the on-line observer restarts at 1 with its new state (the set of new markings) as the initial state (as the initial marking).

Basically an on-line observer is obtained by deriving only the branch of the off-line observer-automaton that explains the on-line plant observation.

In the following $\mathcal{O}_n = t_1^o \dots t_n^o$ denotes a string of n observable events ($\mathcal{O}_n \in \mathcal{T}_o^*$) that are known to have happened in the plant.

The classical on-line observer $\texttt{CO}(\langle \mathcal{N}, M_0 \rangle)$ considers for its initial state x_0^{co} the set of markings given by $\varrho_{co}(x_0^{co})$ where:

$$\varrho_{co}(x_0^{co}) = \left\{ M : M_0 \xrightarrow{\sigma_{uo}} M \wedge \sigma_{uo} \in \mathcal{T}_{uo}^* \right\}$$

Then inductively for $\mathcal{O}_k = t_1^o \dots t_k$, we have that the x_k^{co} state of the on-line classical observer considers the set of markings given by:

$$\varrho_{co}(x_k^{co}) = \left\{ M : M_0 \xrightarrow{\tau} M \wedge \Pi_{\mathcal{T}_o}(\tau) = \mathcal{O}_k \right\}$$

The on-line computation of a (minimal) reduced observer $\texttt{RO}(\langle \mathcal{N}, M_0 \rangle)$ can be performed backwards by calculating the minimal explanations of the received observation where a minimal explanation is a trace that considers only transitions that must have happened *prior to* the execution of the received observation.

3.2.1 Synthesis of an on-line reduced observer for PN models using minimal explanations

Consider the PN model $\langle \mathcal{N}, M_0 \rangle$ and its net unfolding $\mathcal{U}_{\mathcal{N}}(M_0)$ as defined in Section 2.5.

Then use the definition of a minimal configuration $\underline{C}(e)$ (Definition 34) where the event e is s.t. $\phi(e) = t_1^o$ and all the other event-nodes of $\underline{C}(e)$ except e are images of unobservable transitions.

$(E_{\underline{C}(t_1^o)}, \preceq)$ is the partial order relation between the events in $\underline{C}(t_1^o)$ and $\langle E_{\underline{C}(t_1^o)} \rangle$ is the set of all the linearization of $(E_{\underline{C}(t_1^o)}, \preceq)$ (see Definition 2).

Denote by $\phi(\langle E_{\underline{C}(t_1^o)} \rangle)$ the set of all traces in $\langle \mathcal{N}, M_0 \rangle$ that correspond *via* ϕ to strings of $\langle E_{\underline{C}(t_1^o)} \rangle$:

$$\phi(\langle E_{\underline{C}(t_1^o)} \rangle) = \{\tau = \phi(e_1) \dots \phi(e_\nu) \mid \sigma = e_1 \dots e_\nu, \sigma \in \langle E_{\underline{C}(t_1^o)} \rangle\}$$

We say that $\tau \in \phi(\langle E_{\underline{C}(t_1^o)} \rangle)$ is a minimal explanation of t_1^o since all the events (transitions) that are considered in τ *must necessarily have happened* before t_1^o can be executed. Formally we have:

Definition 36. *Given the unfolding $\mathcal{U}_{\mathcal{N}}(M_0)$ of a PN $\langle \mathcal{N}, M_0 \rangle$ and the first observed event $\mathcal{O}_1 = t_1^o$ then $\underline{C}(t_1^o) = (B_{\underline{C}(t_1^o)}, E_{\underline{C}(t_1^o)}, \preceq)$ is a minimal configuration that allows for the execution of t_1^o if:*

i) $\underline{C}(t_1^o)$ *is a configuration in* $\mathcal{U}_{\mathcal{N}}(M_0)$

ii) $\phi(t_1^o) = e_1^o$ *and* $\forall e \in E$ *if* $e \| e_1^o$ *then* $e \notin E_{\underline{C}(t_1^o)}$

iii) $\forall e \in E_{\underline{C}(t_1^o)}$, *if* $e \neq e_1^o$ *then* $\phi(e) \in \mathcal{T}_{uo}$

iv) $B_{\underline{C}(t_1^o)} = \{b \mid (b \in \mathtt{min}_{\preceq}(\mathcal{U}_{\mathcal{N}})) \;\vee\; (b \in {}^\bullet e^\bullet \text{ for some } e \in E_{\underline{C}(t_1^o)})\}$

Denote by $\underline{\mathcal{C}}(\mathcal{O}_1)$ the set of all minimal configurations that satisfy Definition 36 for observation $\mathcal{O}_1 = t_1^o$ and denote by $\underline{\mathcal{E}}(\mathcal{O}_1)$ the set of all minimal explanations of $\mathcal{O}_1$:

$$\underline{\mathcal{E}}(\mathcal{O}_1) = \{\sigma \mid \sigma \in \langle E_{\underline{C}(t_1^o)} \rangle \wedge \underline{C}(\mathcal{O}_1) \in \underline{\mathcal{C}}(\mathcal{O}_1)\}$$

Denote by $\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_1)$ the set of traces in $\langle \mathcal{N}, M_0 \rangle$ that correspond to the minimal explanations $\underline{\mathcal{E}}(\mathcal{O}_1)$:

$$\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_1) = \{\tau \mid \tau = \phi(\sigma) \wedge \sigma \in \underline{\mathcal{E}}(\mathcal{O}_1)\}$$

Then Definition 36 can be extended as follows. For a given a sequence of observed events $\mathcal{O}_n = t_1^o \dots t_n^o$, the minimal configuration that explains the observation $\mathcal{O}_n$ is:

Definition 37. *Given the unfolding $\mathcal{U}_{\mathcal{N}}(M_0)$ of a PN $\langle \mathcal{N}, M_0 \rangle$ and a sequence of observed events $\mathcal{O}_n = t_1^o \dots t_n^o$ then $\underline{C}(\mathcal{O}_n) = (B_{\underline{C}(\mathcal{O}_n)}, E_{\underline{C}(\mathcal{O}_n)}, \preceq)$ is a minimal configuration that allows for the execution of $\mathcal{O}_n$ if:*

1. *$\underline{C}(\mathcal{O}_n)$ is a configuration in $\mathcal{U}_{\mathcal{N}}(M_0)$*
2. *there are n events in $E_{\underline{C}(\mathcal{O}_n)}$ that have images via ϕ observable transitions*
3. *$\forall k, 1 \leq k \leq n$, there exists an unique $e_k^o \in E_{\underline{C}(\mathcal{O}_n)}$ s.t. $\phi(e_k^o) = t_k^o$*
4. *($\forall q, k \quad 1 \leq q < k \leq n$) $\Rightarrow$ ($e_q^o \prec e_k^o$ or $e_q^o \| e_k^o$)*
5. *$\forall e \in E$, if ($e \| e_k^o, \forall k, 1 \leq k \leq n$) then $e \notin E_{\underline{C}(\mathcal{O}_n)}$*
6. *$B_{\underline{C}(\mathcal{O}_n)} = \{ b \in B \mid (b \in \mathtt{min}_{\preceq}(U_{\mathcal{N}})) \vee (b \in {}^\bullet e^\bullet \text{ for some } e \in E_{\underline{C}(\mathcal{O}_n)}) \}$*

Denote by $\underline{\mathcal{C}}(\mathcal{O}_n)$ the set of all minimal configurations that minimally explain $\mathcal{O}_n$ and let $\underline{\mathcal{E}}(\mathcal{O}_n)$, respectively $\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_n)$ be defined as above.

The computation of a reduced observer

The backward computation of the minimal explanations can be seen as a forward search in the reverse net $\overleftarrow{\mathcal{N}}$ (obtained from $\mathcal{N}$ by reversing the direction of all the arcs) using modified firing and enabling rules.

The backward search algorithm that we use for deriving the reduced observer is an adaptation of the algorithm presented in [AIN00], [FRSB02] for checking the coverability property. The problem in [AIN00] is to check (backwards) if, given a bad marking M_{bad}, there is a trace allowable from the initial marking M_0 that leads to a marking greater than M_{bad} or equal. The difference is that here we must calculate all the minimal traces whereas in checking the coverability property there it suffices to prove the existence of one single trace.

Formally we have the following way of defining the reverse net dynamics. Define $a \ominus b = a - b$ if $a \geq b$, and $a \ominus b = 0$ otherwise and extend " $\ominus$ " to multisets in the natural manner [AIN00].

Definition 38. Backwards enabling rule: *A transition t is backward enabled in a marking $M \in \mathbb{N}^{|\mathcal{P}|}$ if $\exists p \in t^\bullet$ s.t. $M(p) \geq 1$.* Backwards firing rule: *A backward enabled transition t in a marking $M \in \mathbb{N}^{|\mathcal{P}|}$ fires backwards from M producing M' (denoted $M \overset{t}{\rightsquigarrow} M'$) where $M' = M \ominus Post(t, \cdot) + Pre(\cdot, t)$.*

A sequence of transitions $\tau = t_\nu \ldots t_1$ is backward allowable from M_ν (denoted $M_\nu \overset{\tau}{\rightsquigarrow} M_0$) if for $\upsilon = \nu, \ldots, 0$, $\tau_\upsilon = t_\nu \ldots t_{\upsilon+1}$, and t_υ is backward enabled in M_υ where $M_\nu \overset{\tau_\upsilon}{\rightsquigarrow} M_\upsilon$ i.e. $\exists M_{\nu-1}, \ldots M_{\upsilon+1}$ s.t.:

$$M_\nu \overset{t_\nu}{\rightsquigarrow} M_{\nu-1} \overset{t_{\nu-1}}{\rightsquigarrow} M_{\nu-2} \ldots \overset{t_{\upsilon+1}}{\rightsquigarrow} M_\upsilon$$

Definition 39. *Given a PN $\langle \mathcal{N}, M_0 \rangle$ and a marking M, then M is covered by M_0 if $\exists M' \leq M_0$ s.t $M \overset{\sigma}{\rightsquigarrow} M'$.*

Definition 40. *Consider a PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ and a partition $\mathcal{T} = \mathcal{T}_o \cup \mathcal{T}_{uo}$. Then given an initial marking M_0 and a final marking M_{fin} denote by $\mathcal{UC}_{\mathcal{N}}(M_{fin}, M_0)$ the set of all markings $M \leq M_0$ that cover M_{fin} by finite unobservable strings:*

$$\mathcal{UC}_{\mathcal{N}}(M_{fin}, M_0) = \left\{ M \leq M_0 \mid M_{fin} \overset{\sigma_{uo}}{\rightsquigarrow} M \wedge \sigma_{uo} \in \mathcal{T}_{uo}^* \right\}$$

Let $\mathcal{UL}_{\mathcal{N}}(M_{fin}, M_0)$ be the set of unobservable strings that are backwards feasible from M_{fin} and lead to a marking $M \leq M_0$:

$$\mathcal{UL}_{\mathcal{N}}(M_{fin}, M_0) = \left\{ \sigma_{uo} \in \mathcal{T}_{uo}^* \mid \exists M \in \mathcal{UC}_{\mathcal{N}}(M_{fin}, M_0) \;\; s.t. \;\; M_{fin} \overset{\sigma_{uo}}{\rightsquigarrow} M \right\}$$

Proposition 3. *We have that:*

(a) Given a PN $\langle \mathcal{N}, M_0 \rangle$ and a marking M that is not covered by M_0 then $\forall M' > M$, M' is not covered by M_0.

(b) Given a PN $\mathcal{N}$, a partition $\mathcal{T} = \mathcal{T}_o \cup \mathcal{T}_{uo}$, a final marking M_{fin}, and an initial marking M_{ini} then:

$$\mathcal{UC}_{\mathcal{N}}(M_{fin}, M_0) \neq \emptyset \text{ if } \forall M'_{fin} < M_{fin}, \mathcal{UC}_{\mathcal{N}}(M'_{fin}, M_0) \neq \emptyset \tag{3.4}$$

Proof. The proof is straightforward. □

For a bounded net $\langle \mathcal{N}, M_0 \rangle$ the computation of $\mathcal{UC}_{\mathcal{N}}(M_{fin}, M_0)$ and $\mathcal{UL}_{\mathcal{N}}(M_{fin}, M_0)$ is performed similarly as the forward computation of the reachability tree. The pseudo-code for this algorithm is provided in Section 7.2 in Appendix. The main features are:

- SET is the set of markings (nodes in the tree) that have to be processed and $sort(SET)$ means that the elements of SET are arranged from $HEAD$ to $TAIL$ such that if $mark(node) \leq mark(node')$ then $node$ is closer to $HEAD$ than $node'$.

- in Procedure $choose\_node\_cur$ the node (marking) $node\_cur$ is chosen from SET if:

    - there are no nodes with unknown status $node \in VISIT\_UKW$ s.t. $mark(node) \leq mark(node\_cur)$ (Procedure $condition$)

- if the selection fails then the algorithm terminates since no solution will be found
- if a node is chosen then for $node\_cur$ the algorithm checks whether it provides a solution or not (Procedure $check\_sol$)
- if $node\_cur$ provides a solution then all the predecessors of $node\_cur$ are removed from $VISIT\_UKW$ and added to $VISIT\_SOL$ (Procedure $propagate\_sol$)
- if $node\_cur$ does not provide a solution then:
    - if there are unobservable transitions that are backwards enabled in $mark(node\_cur)$ then $node\_cur$ is added to $VISIT\_UKW$
    - else $node\_cur$ is declared no solution and all the nodes $node'$ s.t. $mark(node') > mark(node\_cur)$ are removed as well as the nodes that remain unconnected (Procedure $propagate\_no\_sol$).
- in Procedure $make\_new\_node$, the current node is processed and new nodes are added to SET but this is only executed if the marking of a new node is not greater than the marking of a node that was found without solution. Notice that two nodes with the same marking are merged into a single one.

The condition checked by Procedure $condition$ is based on Proposition 3, that is a $node$ is chosen from SET only if all the visited nodes that have a marking smaller than $mark(node)$ are in $VISIT\_SOL$.

Thus a node is processed only if its marking is either $i)$ different than the markings of all the nodes that were visited or $ii)$ smaller than the markings that were visited or $iii)$ if any marking that was visited and is smaller is considered in backward trace that leads to a marking smaller than M_0 (is part of a solution). This guarantees that the backward search algorithm terminates. The proof is simple. Since $\langle \mathcal{N}, M_0 \rangle$ is bounded we have that the set of markings $VISIT\_SOL$ that are considered in the unobservable strings that are backwards feasible from M_{fin} and lead to a marking $M \leq M_0$ is finite.

$$\mathcal{M}_{sol} = \Big\{ M' \mid \exists \sigma_{uo} \in \mathcal{UL}_{\mathcal{N}}(M_{fin}, M_0) \text{ s.t. } \sigma_{uo} = \sigma_{uo_1}\sigma_{uo_2} \wedge \\ M_{fin} \overset{\sigma_{uo_1}}{\rightsquigarrow} M' \overset{\sigma_{uo_1}}{\rightsquigarrow} M \leq M_0 \Big\}$$

When $VISIT\_SOL = \mathcal{M}_{sol}$, that is all the markings that are part of a solution have been derived, a $node$ is chosen to be processed by Procedure $choose\_node\_cur$ only if $mark(node)$ is different from all the markings that were already visited. The algorithm terminates because one cannot generate an infinite sequence of different markings that are neither smaller nor bigger than the other markings that are considered in the sequence.

Given the received observation $\mathcal{O}_n = t_1^o \ldots t_n^o$ the computation of a reduced observer $\mathtt{RO}(\mathcal{O}_n)$ is performed recursively as follows:

1. initialize first $\underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_0) = \{M_0\}$
2. initialize the initial state in the reduced observer automaton $\underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_0) = \{M_0\}$ ($\varrho(x_0^{ro}) = \underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_0)$)
3. then for $k = 1, \ldots, n$
   (a) $M_{fin_k} = Pre(\cdot, t_k^o)$
   (b) for all $M_{k-1} \in \underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_{k-1})$
      i. compute $\mathcal{UC}_{\mathcal{N}}(M_{fin_k}, M_{ini_{k-1}})$ that is the set of markings that cover unobservably M_{fin_k} considering as initial marking $M_{ini_{k-1}} = M_{k-1}$
      ii. derive $\mathcal{UL}_{\mathcal{N}}(M_{fin_k}, M_{ini_{k-1}})$ that is the set of minimal unobservable traces that can be executed from $M_{ini_{k-1}}$ s.t. the resulting marking covers M_{fin_k}
      iii. derive the set of minimal explanations $\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_k)$ and the set of markings $\underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_k)$:
$$\begin{aligned}\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_k) = \{\tau_k \mid \tau_k = \tau_{k-1}\sigma_{uo}t_k^o, \tau_{k-1} \in \underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_k) \\ \wedge \sigma_{uo} \in \mathcal{UL}_{\mathcal{N}}(M_{fin_k}, M_{ini_{k-1}})\}\end{aligned}$$
$$\underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_k) = \left\{M_k \mid M_0 \xrightarrow{\tau_k} M_k \wedge \tau_k \in \underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_k)\right\}$$
4. create a new state x_k^{ro} in $\mathtt{RO}(\mathcal{O}_k)$ and draw an arc from x_{k-1}^{ro} to x_k^{ro} labeled t_k^o
5. $\varrho(x_k^{ro}) = \underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_k)$

Algorithm 1 *Reduced_Obs*

```
Require: $\langle \mathcal{N}, M_0 \rangle$, $\mathcal{T}_o$, $\mathcal{T}_{uo}$, $\mathcal{O}_n$
Ensure: $\mathtt{RO}(\mathcal{O}_n)$
$k = 1$; $\underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_0) = \{M_0\}$; $\varrho(x_0^{ro}) = \{M_0\}$
while $k \leq n$ do
    $M_{fin_k} = Pre(\cdot, t_k^o)$
    for all $M_{k-1} \in \underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_{k-1})$ do
        $M_{ini_{k-1}} = M_{k-1}$
        compute $\mathcal{UC}_{\mathcal{N}}(M_{fin_k}, M_{ini_{k-1}})$
        compute $\mathcal{UL}_{\mathcal{N}}(M_{fin_k}, M_{ini_{k-1}})$
        compute $\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_k)$
        compute $\underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_k)$
    end for
    $x_k^{ro} = \underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_k)$
    $e_k = t_k^o$
    $f(x_{k-1}^{ro}, e_k) = x_k^{ro}$
    $k = k + 1$
end while
```

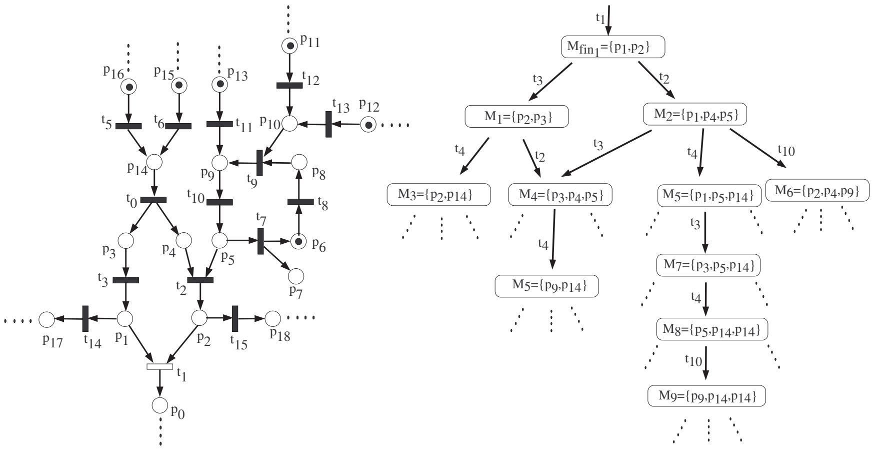


Figure 3.3:

Notice that the reduced observer $\mathtt{RO}(\mathcal{O}_n)$ derived by running Algorithm 1 is unique since $\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_n)$ and respectively $\underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_n)$ are uniquely defined. Notice moreover that in general $\mathtt{RO}(\mathcal{O}_n)$ is not a minimal reduced observer.

However this is not a drawback since in general the set of markings $\underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_k)$ considered in a state x_k^{ro} of the reduced observed derived with Algorithm 1 is a lot smaller than the set of markings considered by the state x_k^{co} of the classical observer.

$$\underline{\mathcal{M}}_{\mathcal{N}}(\mathcal{O}_k) \Subset \mathcal{M}_{\mathcal{N}}(\mathcal{O}_k)$$

The same remark holds also in general for the set of minimal explanations $\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_k)$ and the set of explanations $\mathcal{L}_{\mathcal{N}}(\mathcal{O}_k)$:

$$\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_k) \Subset \mathcal{L}_{\mathcal{N}}(\mathcal{O}_k)$$

The main drawback of the backward search methods is that unreachable states are visited during computation. As shown in [FRSB02], the backward search can be driven by using place invariants (i.e. the visited markings must not violate the P-invariants) or other heuristics. Moreover for real-life applications, the size of the unobservable sub-net that is processed is in general small, so that the calculation is efficient.

Remark 5. *Above we have assumed that the observation is deterministic. If this does not hold the algorithm applies to all the observable sequences τ_k^o s.t. $l_o(\tau_k^o) = \mathcal{O}_k$. The implication of a non-deterministic observation can be intuitively understood as a forward search algorithm where the initial state is given as a set of possible initial markings.*

Example 17. *To illustrate the backward calculation of minimal explanations of an observed event consider the PN $\mathcal{N}'$ displayed in Fig. 3.3-left as a part of a PN model $\mathcal{N}$ of a large plant (the dotted lines emerging out from p_0, p_7, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{17}, p_{18} indicate connections with the remaining of $\mathcal{N}$).*

Let t_1 be the only observable transition in this proper subnet $\mathcal{N}'$ of $\mathcal{N}$. In Fig. 3.3 we present a part of the tree that is derived backward considering that t_1 is observed. The nodes of the tree represent the markings that cover unobservably the marking $M_{fin} = Pre(\cdot, t_1) = \{p_1, p_2\}$.

Notice that the concurrency is not filtered out (e.g. t_3t_2 and t_2t_3 both occurring in separate branches) but the markings are checked whether they are equal and merged into a single one (e.g. $M_4 = \{p_3, p_4, p_5\}$). The node $M_9 = \{p_9, p_{14}, p_{14}\}$ is processed only after the node $M_5 = \{p_9, p_{14}\}$ is found that is part of a solution. If $M_5 = \{p_9, p_{14}\}$ is not part of a solution then M_9 is deleted and also the part that remains unconnected to the root node after eliminating M_9.

$\mathcal{N}'$ contains an unobservable circuit ζ_{uo} with choice places where $\Sigma_{\zeta_{uo}} = t_{10}t_7t_8t_9$. Consider the node $M_5 = \{p_9, p_{14}\}$. We have t_{11}, t_9 among the unobservable transitions that are backward enabled in M_5. If we choose t_9 and then t_8 a token is found in p_6. Unfortunately the computation must be continued to check how many times ζ_{uo} could have been fired because the number of full executions of ζ_{uo} give the number of tokens in p_7.

Thus after backfiring from M_5 the sequence of transition t_9, t_8, t_7, and t_{10} we will obtain $M' = \{p_{14}, p_9, p_{10}\}$ ($M_5 \overset{\zeta_{uo}}{\rightsquigarrow} M'$). Because $M' > M_5$ the backward computation from M' is stopped until M_5 is found part of a backward trace that is solution.

3.2.2 Backward Unfoldings

As for the forward search algorithms, the backward unfolding technique increases the computational efficiency especially for PN models whose degree of concurrency is high compared to their degree of (backward) branching.

In this section we present an algorithm that allows us to derive the set of minimal configurations $\underline{\mathcal{C}}(\mathcal{O}_n)$ for a sequence of observed events $\mathcal{O}_n$ without deriving first the unfolding of the net $\mathcal{U}_{\mathcal{N}}(M_0)$.

Definition 41. *A reverse occurrence net (RON) $\overleftarrow{\mathfrak{O}}$ is a net $\overleftarrow{\mathfrak{O}} = (\overleftarrow{B}, \overleftarrow{E}, \preceq_1)$ s.t.:*

i) $\forall a \in \overleftarrow{B} \cup \overleftarrow{E} : \neg(a \preceq a)$ *(acyclic)*

ii) $\forall b \in \overleftarrow{B} \cup \overleftarrow{E} : \mid \{a : a \preceq b\} \mid < \infty$ *(well-formed)*

iii) $\forall a \in \overleftarrow{B} : \mid a^\bullet \mid \leq 1$ *(no-forward conflict)*

iv) $\mathtt{max}(\overleftarrow{\mathfrak{O}}) \subseteq \overleftarrow{B}$

Definition 42. *Given a PN $\mathcal{N}$ and a final marking M_{fin}, the reverse branching process of $\langle \mathcal{N}, M_{fin} \rangle$ is $\overleftarrow{\mathfrak{B}} = (\overleftarrow{\mathfrak{O}}, \phi)$ s.t.:*

i) $\phi(\overleftarrow{B}) \subseteq \mathcal{P}$ *and* $\phi(\overleftarrow{E}) \subseteq \mathcal{T}$

ii) $M_{fin} \subseteq \phi(\mathtt{max}_{\preceq}(\overleftarrow{\mathfrak{O}}))$

iii) $\forall a, b \in \overleftarrow{E} : ({}^{\bullet}a = {}^{\bullet}b) \wedge (a^{\bullet} = b^{\bullet}) \Rightarrow a = b$

Remark 6. *Notice that condition $ii)$ above requires that $M_{fin} \subseteq \phi(\mathtt{max}(\overleftarrow{\mathfrak{O}}))$ and not $M_{fin} = \phi(\mathtt{max}_{\preceq}(\overleftarrow{\mathfrak{O}}))$ since our aim is to compute markings that cover M_{fin} and not markings that reach M_{fin}.*

Definition 43. *Given a PN $\mathcal{N}$ the immediate backward conflict relation $\overleftarrow{\sharp}_1 \subseteq \mathcal{T} \times \mathcal{T}$ is defined as follows:*

$$\forall (t_1, t_2) \in \mathcal{T} \times \mathcal{T} : t_1 \overleftarrow{\sharp}_1 t_2 \textit{ if } t_1^{\bullet} \cap t_2^{\bullet} \neq \emptyset$$

Then define $\overleftarrow{\sharp} \subseteq (\mathcal{P} \cup \mathcal{T}) \times (\mathcal{P} \cup \mathcal{T})$ as:

$$\forall (a, b) \in (\mathcal{P} \cup \mathcal{T}) \times (\mathcal{P} \cup \mathcal{T}) : a \overleftarrow{\sharp} b \Rightarrow \exists (t_1, t_2) \in \overleftarrow{\sharp}_1 \textit{ s.t. } a \preceq t_1 \textit{ and } b \preceq t_2.$$

In words, two transitions t_1, t_2 are in immediate backward conflict if their occurrence produces tokens in the same place e.g. $t_1 \overleftarrow{\sharp}_1 t_2 \Leftrightarrow t_1^{\bullet} \cap t_2^{\bullet} \neq \emptyset$. The idea is that if we have a place p that contains a token and there are two transitions t_1 and t_2 s.t. $p \in t_1^{\bullet} \cap t_2^{\bullet}$ then to explain how the token was produced in p we have that either t_1 or t_2 have fired before but not both of them.

Notice that our goal is to derive in the backward unfolding configurations that are minimal configurations in the forward unfolding. The problem that occurs is that a transition t is enabled backwards in the marking M if at least one of its output places is marked in M. Thus a transition t may be executed backwards considering that it is enabled by any subset of its output places that are marked in M. These different execution-modes of a transition are in auto-conflict and we represent this by the auto-conflict relation as defined bellow:

Definition 44. *Given a PN $\mathcal{N}$, a final marking M_{fin}, and a reverse branching process $\overleftarrow{\mathfrak{B}} = (\overleftarrow{\mathfrak{O}}, \phi)$, the immediate auto-conflict relation $\overleftarrow{\sharp}_{ac_1} \subseteq \overleftarrow{E} \times \overleftarrow{E}$ is defined as follows:*

$$\forall (e_1, e_2) \in \overleftarrow{E} \times \overleftarrow{E}, \phi(e_1) = \phi(e_2) : e_1 \overleftarrow{\sharp}_{ac_1} e_2 \textit{ if } e_1^{\bullet} \cap e_2^{\bullet} \neq \emptyset \wedge e_1^{\bullet} \neq e_2^{\bullet}$$

Then define the auto-conflict relation $\overleftarrow{\sharp}_{ac} \subseteq (\overleftarrow{B} \cup \overleftarrow{E}) \times (\overleftarrow{B} \cup \overleftarrow{E})$ as:

$\forall(a,b) \in (\overleftarrow{B} \cup \overleftarrow{E}) \times (\overleftarrow{B} \cup \overleftarrow{E}) : a \overleftarrow{\sharp}_{ac} b \Rightarrow \exists(e_1, e_2) \in \overleftarrow{\sharp}_{ac_1}$ *s.t.* $a \preceq e_1$ *and* $b \preceq e_2$.

Definition 45. *A configuration* $\overleftarrow{C} = (\overleftarrow{B}_{\overleftarrow{C}}, \overleftarrow{E}_{\overleftarrow{C}}, \preceq_1)$ *in a reverse occurrence net* $\overleftarrow{\mathfrak{O}}$ *is defined as follows:*

i) $\overleftarrow{C}$ *is a sub-net of* $\overleftarrow{\mathfrak{O}}$

ii) $\overleftarrow{C}$ *is causally downward-closed -* $\forall a \in \overleftarrow{B} \cup \overleftarrow{E}$, $\forall b \in \overleftarrow{B}_{\overleftarrow{C}} \cup \overleftarrow{E}_{\overleftarrow{C}} : b \preceq a \Rightarrow a \in \overleftarrow{B}_{\overleftarrow{C}} \cup \overleftarrow{E}_{\overleftarrow{C}}$

iii) $\overleftarrow{C}$ *is backward conflict free -* $\forall(a,b) \in (\overleftarrow{B}_{\overleftarrow{C}} \cup \overleftarrow{E}_{\overleftarrow{C}}) \times (\overleftarrow{B}_{\overleftarrow{C}} \cup \overleftarrow{E}_{\overleftarrow{C}}) \Rightarrow \neg(a \overleftarrow{\sharp} b)$

iv) $\overleftarrow{C}$ *is auto-conflict free -* $\forall(a,b) \in (\overleftarrow{B}_{\overleftarrow{C}} \cup \overleftarrow{E}_{\overleftarrow{C}}) \times (\overleftarrow{B}_{\overleftarrow{C}} \cup \overleftarrow{E}_{\overleftarrow{C}}) \Rightarrow \neg(a \sharp_{ac} b)$

v) $\mathtt{max}_{\preceq}(\overleftarrow{C}) \subseteq \mathtt{max}_{\preceq}(\overleftarrow{\mathfrak{O}})$ *and* $M_{fin} \subseteq \phi(\mathtt{max}_{\preceq}(\overleftarrow{C}))$

Denote by $\overleftarrow{\mathcal{C}}$ *the set of all the configurations of a reverse occurrence net* $\overleftarrow{\mathfrak{O}}$.

Definition 46. *Given a PN* $\mathcal{N}$ *and a final marking* M_{fin} *and two reverse branching processes* $\overleftarrow{\mathfrak{B}}, \overleftarrow{\mathfrak{B}}'$ *then* $\overleftarrow{\mathfrak{B}}' \sqsubseteq \overleftarrow{\mathfrak{B}}$ *if there exists an injective homomorphism* $\overleftarrow{\psi} : \overleftarrow{\mathfrak{B}}' \rightarrow \overleftarrow{\mathfrak{B}}$ *s.t.* $\overleftarrow{\psi}(\mathtt{min}(\overleftarrow{\mathfrak{B}}')) = \mathtt{min}(\overleftarrow{\mathfrak{B}})$ *and* $\overleftarrow{\phi} \circ \overleftarrow{\varphi} = \overleftarrow{\phi}'$.

Definition 47. *Given a bounded PN* $\mathcal{N}$ *with an initial marking* M_0 *and a final marking* M_{fin} *denote by* $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(M_{fin}, M_0)$ *the maximal branching process w.r.t. set inclusion s.t.* $\forall \overleftarrow{C} \in \overleftarrow{\mathcal{C}}$, $\exists \overleftarrow{C}' \in \overleftarrow{\mathcal{C}}$ *s.t. i)* $\overleftarrow{C} \sqsubseteq \overleftarrow{C}'$, *ii)* $\phi(\mathtt{min}_{\preceq}(\overleftarrow{C}')) \subseteq M_0$ *and iii)* $M_{fin} \subseteq \phi(\mathtt{max}_{\preceq}(\overleftarrow{C}'))$.

Proposition 4. $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(M_{fin}, M_0)$ *is unique up to isomorphism.*

Proof. Consider $\sigma_{uo} \in \mathcal{UL}_{\mathcal{N}}(M_{fin}, M_0)$ an unobservable string that is backwards feasible from M_{fin} and leads to a marking $M \leq M_0$. Since $\langle \mathcal{N}, M_0 \rangle$ is bounded then any (forward) execution of a cycle ζ does not increase the marking. Thus if the (forward) execution of a cycle ζ decreases the marking there are a finite number of executions of the cycles ζ otherwise the marking remains the same after executing the cycle ζ and then ζ is not fired the second time. Thus $\mathcal{UL}_{\mathcal{N}}(M_{fin}, M_0)$ and $\mathcal{UC}_{\mathcal{N}}(M_{fin}, M_0)$ are finite sets that are uniquely represented by $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(M_{fin}, M_0)$.

□

Denote by $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(t_1^o)$ the backward unfolding calculated for the first observed event t_1^o where $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(t_1^o)$ is obtained by appending the event e_1^o ($\phi(e_1^o) = t_1^o$) to the unfolding $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(M_{fin}, M_0)$ where $M_{fin} = Pre(\cdot, t_1^o)$.

Notice that a backward configuration $\overleftarrow{C}$ in the backward unfolding $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(t_1^o)$ is such that $\phi(\mathtt{min}_{\preceq}(\overleftarrow{C})) \subseteq M_0$. Denote in what follows by $\overleftarrow{\underline{C}}$ the configuration obtained from $\overleftarrow{C}$ by adding for each token in a place p from the initial marking that was not *"consumed"* a condition b s.t. ${}^\bullet b = \emptyset$ and $\phi(b) = p$.

Denote by $\overleftarrow{\underline{\mathcal{C}}}(t_1^o)$ the set of backward configurations that are derived for the first observed event t_1^o.

Proposition 5. *Given a PN $\langle \mathcal{N}, M_0 \rangle$ and the first observed event in the plant t_1^o we have that:*

1. $\forall \overleftarrow{\underline{C}}(t_1^o) \in \overleftarrow{\underline{\mathcal{C}}}(t_1^o) \Rightarrow \exists \underline{C}(t_1^o) \in \underline{\mathcal{C}}(t_1^o)$ *such that* $\overleftarrow{\underline{C}}(t_1^o)$ *and* $\underline{C}(t_1^o)$ *are isomorphic configurations.*
2. $\forall \underline{C}(t_1^o) \in \underline{\mathcal{C}}(t_1^o) \Rightarrow \exists \overleftarrow{\underline{C}}(t_1^o) \in \overleftarrow{\underline{\mathcal{C}}}(t_1^o)$ *such that* $\underline{C}(t_1^o)$ *and* $\overleftarrow{\underline{C}}(t_1^o)$ *are isomorphic configurations.*

Proof. The proof of $i)$ is as follows. By definition we have that any configuration in the backward unfolding $\forall \overleftarrow{C}(t_1^o) \in \overleftarrow{\mathcal{C}}(t_1^o)$, $\overleftarrow{C}(t_1^o) = (\overleftarrow{B}_{\overleftarrow{C}(t_1^o)}, \overleftarrow{E}_{\overleftarrow{C}(t_1^o)}, \preceq)$ is a causal net that is $\forall b \in \overleftarrow{B}_{\overleftarrow{C}(t_1^o)} \Rightarrow \mid b^\bullet \mid \leq 1$ and $\mid {}^\bullet b \mid \leq 1$ (all the conditions nodes have at most one input respectively at most one output event node). Then by adding for each token from the initial marking that was not consumed $M_0 \setminus \phi(\mathtt{min}_{\preceq}(\overleftarrow{C}))$ we have that under an adequate labeling $\overleftarrow{\underline{C}}(t_1^o)$ is a minimal configuration in $\mathcal{U}_{\mathcal{N}}(M_0)$. Then the proof of $ii)$ is straightforward since the backward search terminates by a fix-point when no more traces (configurations) can be generated. □

Algorithm to construct $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(M_{fin}, M_0)$

Given a configuration $\overleftarrow{C} = (\overleftarrow{B}_{\overleftarrow{C}}, \overleftarrow{E}_{\overleftarrow{C}}, \preceq_1)$ in the backward net unfolding $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(M_{fin}, M_0)$ denote by $CUT(\overleftarrow{C})$ the maximal (w.r.t. set inclusion) set of concurrent conditions and then denote by $mark(\overleftarrow{C})$ the marking that corresponds to $CUT(\overleftarrow{C})$.

$$CUT(\overleftarrow{C}) = \left\{ {}^\bullet e \mid e \in \overleftarrow{E}_{\overleftarrow{C}} \right\} \cup \mathtt{max}(\overleftarrow{C}) \setminus \left\{ e^\bullet \mid e \in \overleftarrow{E}_{\overleftarrow{C}} \right\}$$

Denote in the following by $\overleftarrow{X}^{con}_{\overleftarrow{B}}$ a set of concurrent conditions in $\overleftarrow{C}$. A transition t is backward enabled in $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(M_{fin}, M_0)$ by a configuration $\overleftarrow{C}$ if

$\overleftarrow{X}^{con}_{B} \subseteq CUT(\overleftarrow{C})$ and $0 < \phi(\overleftarrow{X}^{con}_{B}) \leq Post(t, \cdot)$.

Denote by $B\_ENABLED(\overleftarrow{C})$ the set of all backwards enabled transitions.

$$B\_ENABLED(\overleftarrow{C}) = \left\{ (\overleftarrow{X}^{con}_{\overleftarrow{B}}, t) \mid (\overleftarrow{X}^{con}_{\overleftarrow{B}}, t) \ \text{– backwards enabled} \ \right\}$$

The backward unfolding $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(M_{fin})$ is constructed as follows: The initial configuration $\overleftarrow{C}^{\perp} = (\overleftarrow{B}_{\overleftarrow{C}^{\perp}}, \overleftarrow{E}_{\overleftarrow{C}^{\perp}}, \preceq))$ is obtained considering a condition b in $\overleftarrow{B}_{\overleftarrow{C}^{\perp}}$ for each token in a marked place in M_{fin}.

Then $\overleftarrow{C}^{\perp}$ is recursively extended by appending (backwards) a transitions $(\overleftarrow{X}^{con}_{\overleftarrow{B}}, t) \in B\_ENABLED(\overleftarrow{C})$ that is backwards enabled in the following way:

i) create an event-node e and label it $\phi(e) = t$

ii) add arcs from e to each $b \in \overleftarrow{X}^{con}_{\overleftarrow{B}}$

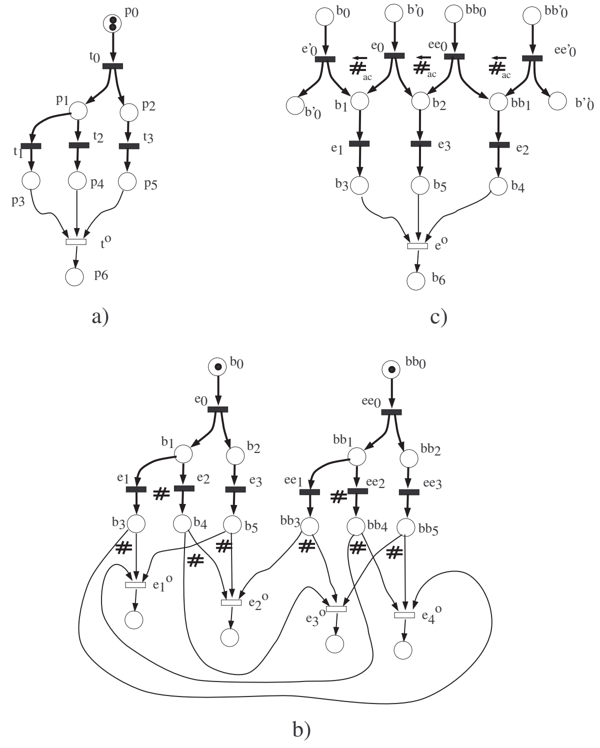
iv) add conditions b s.t. $\phi(b) = p \wedge p \in t^{\bullet} \setminus \phi(\overleftarrow{X}^{con}_{\overleftarrow{B}})$ and draw arcs from e to each b

iii) add conditions b s.t. $\phi(b) = p \wedge p \in {}^{\bullet}t$ and draw arcs from each b to e

$\overleftarrow{\mathcal{U}}_{\mathcal{N}}(M_{fin}, M_0)$ is generated extending each configuration by enabled transitions the only requirement being that $\phi(e_1) = \phi(e_2) \wedge e_1^{\bullet} = e_2^{\bullet} \Rightarrow e_1 = e_2$ (no redundancy).

Throughout the remaining of this paper we use the notation $C \odot e$ and $e \odot \overleftarrow{C}$ to indicate that a configuration C resp. $\overleftarrow{C}$ is extended forward, resp. backwards by appending an event e.

Example 18. *To illustrate the computation of* $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(t^o)$ *consider for the PN* $\mathcal{N}$ *displayed in Fig. 3.4.a that* t^o *was executed once. Fig. 3.4.b displays* $\mathcal{U}_{\mathcal{N}}(M_0)$ *while in Fig. 3.4.c* $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(t^o)$ *is displayed. We have that* $\overleftarrow{\mathcal{C}} = \left\{\overleftarrow{C}_1, \overleftarrow{C}_2\right\}$ *where for* $\overleftarrow{C}_1$ *and* $\overleftarrow{C}_2$ *we have* $E_{\overleftarrow{C}_1} = \{e'_0, ee_0, e_1, e_2, e_3, e^o\}$ *and* $E_{\overleftarrow{C}_2} = \{e_0, ee'_0, e_1, e_2, e_3, e^o\}$ *respectively.*



a)

c)

b)

Figure 3.4:

Consider the case of the first observed event in the plant $\mathcal{O}_1 = t^o_1$. The reverse occurrence net $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(t^o_1)$ is calculated by Algorithm 2.

Algorithm 2 B_Unfold(t^o)

```
Require: $t^o$, $\langle \mathcal{N}, M_0 \rangle$, $\mathcal{T}_o$, $\mathcal{T}_{uo}$
Ensure: $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(t^o)$
$\overleftarrow{\mathcal{U}}_{\mathcal{N}}(t^o) = {}^\bullet e \cup e \cup e^\bullet$ where $\phi(e) = t^o$
$\overleftarrow{C}_0 = \overleftarrow{\mathcal{U}}_{\mathcal{N}}(t^o)$; $\overleftarrow{\mathcal{C}} = \left\{ \overleftarrow{C}_0 \right\}$
B_ENABLE $= \bigcup_{\overleftarrow{C} \in \overleftarrow{\mathcal{C}}}$ B_ENABLED$(\overleftarrow{C}) \cap \mathcal{T}_{uo}$
while B_ENABLE $\neq 0$ do
    pick and delete $e = (X^{\overleftarrow{con}}_{\overleftarrow{B}}, t) \in$ B_ENABLED
    $\overleftarrow{C}_{new} = e \odot \overleftarrow{C}$ {extend $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(t^o)$}
    $\overleftarrow{\mathcal{C}} = \overleftarrow{\mathcal{C}} \cup \overleftarrow{C}_{new}$
    B_ENABLED := B_ENABLED $\cup$ B_ENABLED$(\overleftarrow{C}_{new})$
end while
```

Remark 7. *Notice that the same technical conditions as presented for the standard backward search guarantee that the backward unfolding algorithm B_Unfold(t^o) terminates.*

For a sequence of observed transitions $\mathcal{O}_n = t^o_1 \ldots t^o_n$ the computation of $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(\mathcal{O}_n)$ and $\overleftarrow{\mathcal{C}}(\mathcal{O}_n)$ is carried out recursively as follows.

Algorithm 3 B_Unfold($\mathcal{O}_n$)

```
Require: $\mathcal{O}_{\theta_c}$, $\langle \mathcal{N}, M_0 \rangle$, $\mathcal{T}_o$, $\mathcal{T}_{uo}$
Ensure: $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(\mathcal{O}_n)$
$k := 1$; $M_{fin_1} = Pre(\cdot, t^o_1)$; $M_{ini_1} = M_0$
B_Unfold$(M_{fin_1}, M_{ini_1})$ { compute $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(\mathcal{O}_1)$}
$\overleftarrow{\mathcal{C}}(\mathcal{O}_1) \rightarrow \overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_1)$
while $k \leq n$ do
    while $\overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_k) \neq \emptyset$ do
        pick and delete $\overleftarrow{\underline{C}} \in \overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_k)$
        $M_{fin_{k+1}} = Pre(\cdot, t^o_{k+1})$; $M_{ini_{k+1}} = mark(\overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_{\theta_k}))$
        B_Unfold$(M_{fin_{k+1}}, M_{ini_k})$
        $\overleftarrow{\mathcal{C}}(\mathcal{O}_{k+1}, \overleftarrow{\underline{C}}) \rightarrow \overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_{k+1}, \overleftarrow{\underline{C}})$
        $\overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_{k+1}) = \overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_{k+1}) \cup \overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_{k+1}, \overleftarrow{\underline{C}})$
    end while
    $k := k + 1$
end while
```

Proposition 6. *Given a PN $\langle \mathcal{N}, M_0 \rangle$ and a sequence of observed events $\mathcal{O}_n = t_1^o, \ldots, t_n^o$ we have that:*

1. $\forall \underline{C}(\mathcal{O}_n) \in \overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_n) \Rightarrow \exists \underline{C}(\mathcal{O}_n) \in \underline{\mathcal{C}}(\mathcal{O}_n)$ *such that* $\overleftarrow{\underline{C}}(\mathcal{O}_n)$ *and* $\underline{C}(\mathcal{O}_n)$ *are isomorphic configurations.*

2. $\forall \underline{C}(\mathcal{O}_n) \in \underline{\mathcal{C}}(\mathcal{O}_n) \Rightarrow \exists \overleftarrow{\underline{C}}(\mathcal{O}_n) \in \overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_n)$ *such that* $\underline{C}(\mathcal{O}_n)$ *and* $\overleftarrow{\underline{C}}(\mathcal{O}_n)$ *are isomorphic configurations.*

Proof. We prove this result by induction as follows.
$\mathcal{O}_1 = t_1^o$ (base) for $\mathcal{O}_1 = t_1^o$ we have Proposition 5.

$\mathcal{O}_k$ (induction base) $\underline{\mathcal{C}}(\mathcal{O}_k)$ and $\overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_k)$ contain isomorphic configurations.

$\mathcal{O}_{k+1} = \mathcal{O}_k t_{k+1}^o$ (induction step) $\overleftarrow{\underline{C}}(\mathcal{O}_{k+1}) \in \overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_{k+1})$ is obtained running B_Unfold(t_{k+1}^o) considering the of initial markings $mark(\overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_{\theta_c}^k))$ for $\overleftarrow{\underline{C}}(\mathcal{O}_k) \in \overleftarrow{\underline{\mathcal{C}}}(\mathcal{O}_k)$, so the proof is straightforward. □

Example 19. *The backward unfolding $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(t_6)$ is presented in Fig. 3.5. Consider a configuration in the backward unfolding $\overleftarrow{C}_1(t_6) = (\overleftarrow{B}_{\overleftarrow{C}(t_6)}, \overleftarrow{E}_{\overleftarrow{C}(t_6)}, \preceq)$ where:*

- $\overleftarrow{B}_{\overleftarrow{C}(t_6)} = \{b_6, b_4, b_9, b_1', b_2', b_8, b_0\}$
- $\overleftarrow{E}_{\overleftarrow{C}(t_6)} = \{e_6, e_3, e_9, e_0'\}$

$\overleftarrow{\underline{C}}_1(t_6)$ *is obtained from* $\overleftarrow{C}_1(t_6)$ *by adding to* $\overleftarrow{B}_{\overleftarrow{C}(t_6)}$ *a condition node that correspond with the second token that is present in* M_0 *(*$\overleftarrow{B}_{\overleftarrow{\underline{C}}(t_6)} = \overleftarrow{B}_{\overleftarrow{C}(t_6)} \cup \{bb_0'\}$*).*

For $\overleftarrow{\underline{C}}_1(t_6)$ *we have:*

$$\phi(\langle \overleftarrow{E}_{\overleftarrow{C}(t_6)} \rangle) = \{t_9 t_0 t_3 t_6, t_0 t_9 t_3 t_6\}$$

and $\underline{\mathcal{L}}_{\mathcal{N}}(t_6) = \{t_0 t_4 t_6; t_1 t_4 t_6; t_2 t_4 t_6; t_0 t_9 t_3 t_6; t_9 t_0 t_3 t_6; t_0 t_2 t_5 t_9 t_{13} t_3 t_6\}$

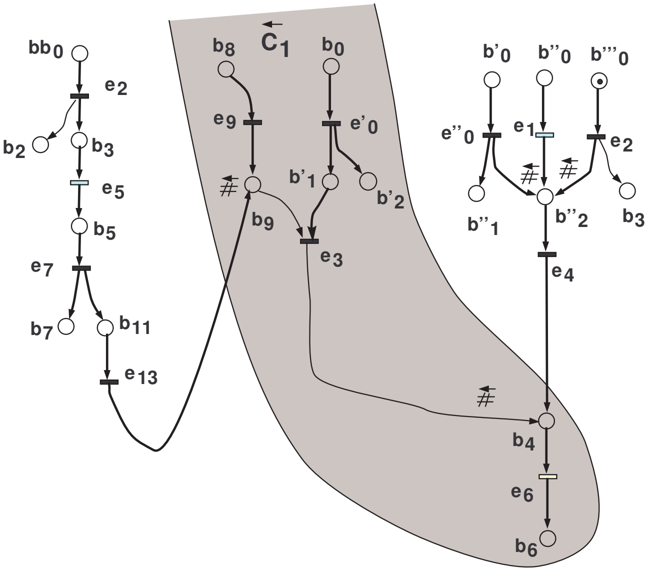

bb_0
e_2
b_2
b_3
e_5
b_5
e_7
b_7
b_{11}
e_{13}
$\overleftarrow{C}_1$
b_8
b_0
e_9
e'_0
$\overleftarrow{\#}$
b'_1
b'_2
b_9
e_3
b'_0
b''_0
b'''_0
e''_0
e_1
e_2
$\overleftarrow{\#}$
$\overleftarrow{\#}$
b''_1
b''_2
b_3
e_4
$\overleftarrow{\#}$
b_4
e_6
b_6

Figure 3.5:

Chapter 4

Diagnosis of PN models

4.1 Introduction

This chapter is devoted to presenting the diagnosis of a plant modeled as a PN. We consider two settings namely in Section 4.2 we consider the case of a plant monitored by a centralized agent while Section 4.3 presents a distributed diagnosis algorithm. In the distributed case the plant is modeled by a collection of different interacting components that are locally monitored by local diagnoser-agents. The local agents can interact by communicating among each other. Each component is modeled by a PN model while the interactions between the components (local-sites) are represented by tokens that can pass from one PN model to another via common border places ([Val94], [BFHJ03] , [GL03], [BJ04], [FBHJ05], [GL05]).

The observation in the plant detects the occurrence of some events that are reported to the central agent in centralized setting respectively to the local agent in distributed case. In the PN model some transitions can be observed when they are executed i.e. whenever an observable event fires, this event is available for use by the algorithm running in the centralized agent respectively local agent.

The plant model represents the normal plant behavior as well as the abnormal usually undesirable behavior that can occur after a fault has occurred. The abnormal behavior is initiated by a subset of unobservable (silent) transitions that represent the fault events that may happen in the plant. The diagnoser(s) must use the plant model, the plant observation, and the information exchange in the distributed setting in order to ask the following questions: *"Did a fault happen or not ?"*(fault detection), *"Which kind of fault happened if any ?"* (fault isolation) and *"How did it happen ?"*(explanations [McI98]).

The diagnosis task should be seen in the following as part of a central-

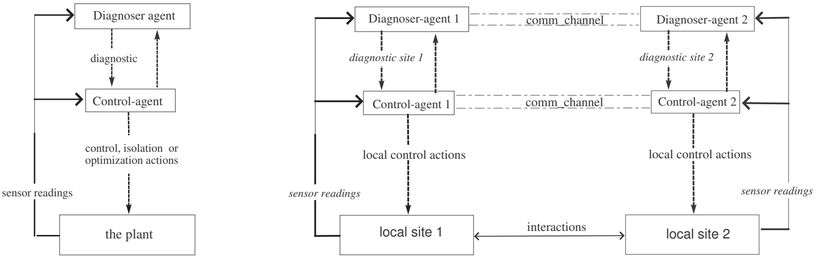


Figure 4.1:

ized resp. decentralized supervisory architecture where the diagnosis result is used on-line for taking some control action that are mandatory for maintaining the safe operation of the plant (see Fig. 4.1).

In this respect and taking into account that the plant under investigation is assumed to have a large size it is important to have specified before designing the algorithms what is desired to include in the plant diagnostic. For example, whether the diagnostic is concerned with finding all the fault-events that *"could have happened in the plant without contradicting the plant observation"* or only the fault events that *"necessarily must have happened for explaining the received observation"*.

Problem statement

We consider the following structural and functional assumptions:

- the overall plant PN model $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ is bounded and ordinary (the capacity of all the arcs is 1)

- the initial marking M_0 is precisely known

- the plant observation is represented by a subset of observable transitions $\mathcal{T}_o \subseteq \mathcal{T}$

- the occurrence of an observable transition $t \in \mathcal{T}_o$ is always reported correctly and without delays

- the faults are represented by a subset $\mathcal{T}_f$ of unobservable (silent) transitions ($\mathcal{T}_f \subseteq \mathcal{T}_{uo}$)

- *no-fault-masking* i.e. the occurrence of a fault transition must have effects on the resulting marking and consequently on the future plant behavior

- *no-design error assumption*

- *no-hidden interactions* i.e. no unrepresented interactions (the closed world assumption)

4.2 Centralized diagnosis

In this section we present two algorithms for the centralized diagnosis of a large plant. We present first in Section 4.2.1 the classical diagnosis algorithm based on the calculation of the complete explanations of the received observation. We call it classical since the diagnosis is performed based on the calculations derived by a classical observer as presented in Section 3.1.

Then in Section 4.2.2 we propose a diagnosis algorithm based on the calculations of the minimal explanations of the received observation (see Section 3.2). We show that the diagnosis result based on minimal explanations is sufficient for detecting the faults that happened for sure in the plant.

4.2.1 Centralized diagnosis based on complete explanations

Consider the plant model given as a PN $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ with given initial marking M_0. Then consider the partition of the transition set $\mathcal{T}$ in two disjunct subsets $\mathcal{T}_o$ observable and respectively $\mathcal{T}_{uo}$ unobservable transitions and let $\mathcal{T}_f \subset \mathcal{T}_{uo}$ be the subset of the unobservable transitions that model the faults. Consider the plant observation given by $\mathcal{O}_n = t_1^o \dots t_n^o$.

Since $\mathcal{O}_n$ is correct and there are no delays in receiving the observation $\mathcal{O}_n$, the possible plant evolutions are given by the set of all the possible traces in the PN model $\mathcal{N}$ that start from the known initial marking M_0 and that obey the observation $\mathcal{O}_n$:

$$\mathcal{L}_{\mathcal{N}}(\mathcal{O}_n) = \{\tau \in \mathcal{L}_{\mathcal{N}}(M_0) \mid \Pi_{\mathcal{T}_o}(\tau) = \mathcal{O}_n\}$$

The set of the possible states (markings) the plant can be in is:

$$\mathcal{M}_{\mathcal{N}}(\mathcal{O}_n) = \left\{M \mid \exists \tau \in \mathcal{L}_{\mathcal{N}}(\mathcal{O}_n) \text{ s.t. } M_0 \xrightarrow{\tau} M\right\}$$

Consequently the plant diagnosis after observing $\mathcal{O}_n$ is obtained by projecting the set of possible evolutions onto the set of fault events $\mathcal{T}_f$:

$$\mathcal{D}_{\mathcal{N}}(\mathcal{O}_n) = \left\{\sigma_f \mid \sigma_f = \Pi_{\mathcal{T}_f}(\tau) \wedge \tau \in \mathcal{L}_{\mathcal{N}}(\mathcal{O}_n)\right\} \tag{4.1}$$

The centralized diagnosis result is:

$$\mathcal{DR}_{\mathcal{N}}(\mathcal{O}_n) = \begin{cases} \texttt{N} & \text{if } \mathcal{D}_{\mathcal{N}}(\mathcal{O}_n) = \{\epsilon\} \\ \texttt{F} & \text{if } \epsilon \notin \mathcal{D}_{\mathcal{N}}(\mathcal{O}_n) \\ \texttt{UF} & \text{if } \{\epsilon\} \subsetneq \mathcal{D}_{\mathcal{N}}(\mathcal{O}_n) \end{cases} \tag{4.2}$$

where N, F and UF are the diagnoser state *normal* (no fault has happened), *fault* (a fault of kind F has happened *for sure*) and respectively *uncertain* (a fault may have happened) [SSL$^+$95].

4.2.2 Centralized diagnosis based on minimal explanations

The rationale behind deriving a diagnosis algorithm based on a subset of the possible evolutions of the plant is as follows.

Assume a plant having the PN model $\langle \mathcal{N}, M_0 \rangle$ that generates the observation $\mathcal{O}_n$. For large plants there often exists a relevant (and in general small) part of the plant (say the one that corresponds with a proper subnet $\mathcal{N}' \Subset \mathcal{N}$) s.t. it is sufficient to analyze $\langle \mathcal{N}', M_0' \rangle$ for obtaining the desired result.

For instance (see Fig. 4.2) $\mathcal{N}'$ is such that analyzing $\langle \mathcal{N}', M_0' \rangle$ (where $M_0'(p) = M_0(p)$ for $p \in \mathcal{P}'$) one can derive the set of feasible traces $\mathcal{L}_{\mathcal{N}'}(\mathcal{O}_n)$ whose use for fault isolation actions (say under the control function $\Phi(\mathcal{L}_{\mathcal{N}'}(\mathcal{O}_n)))$ gives the same result as taking control actions using the set of feasible traces $\mathcal{L}_{\mathcal{N}'}(\mathcal{O}_n)$ derived considered the overall plant model $\mathcal{N}$ that is $\Phi(\mathcal{L}_{\mathcal{N}'}(\mathcal{O}_n)) = \Phi(\mathcal{L}_{\mathcal{N}}(\mathcal{O}_n))$.

Notice that since $\mathcal{N}'$ is a proper subnet of $\mathcal{N}$, any trace in $\langle \mathcal{N}', M_0' \rangle$ is also a trace in $\langle \mathcal{N}, M_0 \rangle$.

Denote by $\mathcal{M}'_{\mathcal{N}}(\mathcal{O}_n)$ the set of estimated states in $\mathcal{N}$ obtained based on the traces $\mathcal{L}_{\mathcal{N}'}(\mathcal{O}_n)$ derived by analyzing $\langle \mathcal{N}', M_0' \rangle$ and recall that $\mathcal{M}_{\mathcal{N}}(\mathcal{O}_n)$ is the set of estimated markings obtained by analyzing the entire plant PN model $\langle \mathcal{N}, M_0 \rangle$.

If we have $\mathcal{M}'_{\mathcal{N}}(\mathcal{O}_n)$ such that:

$$UR_{\mathcal{N}}(\mathcal{M}'(\mathcal{O}_n)) = \mathcal{M}_{\mathcal{N}}(\mathcal{O}_n)$$

it means that we have not included in our calculation some unobservable events that could have happened concurrently with the last observed event in $\mathcal{O}_n$.

In the following, based on the concept of minimal explanations, we identify what the relevant part of the plant is when the plant is subject to the diagnosis.

Consider what follows that $\Phi(\cdot)$ gives the decisions taken when the centralized diagnoser agent is sure that a fault happened (the state {F}) and let the received observation be $\mathcal{O}_n$ as above.

Moreover let the set of minimal explanations $\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_n)$ and the set of estimated markings of $\underline{\mathcal{M}}(\mathcal{O}_n)$ be derived as presented in Section 3.2.

The minimal plant diagnosis after observing $\mathcal{O}_n$ (denoted $\underline{\mathcal{D}}_{\mathcal{N}}(\mathcal{O}_n)$) is obtained by projecting the set of minimal explanations on the set of fault

$$\begin{tikzcd}
 & \Phi(\mathcal{L}_{\mathcal{N}'}(\mathcal{O}_n)) & = & \Phi(\mathcal{L}_{\mathcal{N}}(\mathcal{O}_n)) & \\
\mathcal{L}_{\mathcal{N}'}(\mathcal{O}_n) \arrow[ur] \arrow[ddr] & \mathcal{L}_{\mathcal{N}'}(M'_0) \arrow[l] & \subseteq & \mathcal{L}_{\mathcal{N}}(M_0) \arrow[r] & \mathcal{L}_{\mathcal{N}}(\mathcal{O}_n) \arrow[ul] \arrow[ddl] \\
 & \langle \mathcal{N}', M'_0 \rangle \arrow[u] & \sqsubseteq & \langle \mathcal{N}, M_0 \rangle \arrow[u] \arrow[dll] \arrow[d] & \\
 & \mathcal{M}'(\mathcal{O}_n) \arrow[d] & \subseteq & \mathcal{M}_{\mathcal{N}}(\mathcal{O}_n) \arrow[d] & \\
 & UR_{\mathcal{N}}(\mathcal{M}'(\mathcal{O}_n)) & = & \mathcal{M}_{\mathcal{N}}(\mathcal{O}_n) &
\end{tikzcd}$$

Figure 4.2:

events $\mathcal{T}_f$:

$$\underline{\mathcal{D}}_{\mathcal{N}}(\mathcal{O}_n) = \big\{\sigma_f \mid \sigma_f = \Pi_{\mathcal{T}_f}(\tau) \wedge \tau \in \underline{\mathcal{L}}_{\mathcal{N}}(O_n)\big\} \tag{4.3}$$

Then the diagnosis result based on the set of minimal explanations is:

$$\underline{\mathcal{DR}}_{\mathcal{N}}(\mathcal{O}_n) = \begin{cases} \texttt{N} & \text{if } \underline{\mathcal{D}}_{\mathcal{N}}(\mathcal{O}_n) = \{\epsilon\} \\ \texttt{F} & \text{if } \epsilon \notin \underline{\mathcal{D}}_{\mathcal{N}}(\mathcal{O}_n) \\ \texttt{UF} & \text{if } \epsilon \subsetneq \underline{\mathcal{D}}_{\mathcal{N}}(\mathcal{O}_n) \end{cases} \tag{4.4}$$

Proposition 7. *If the plant PN model $\mathcal{N}$ obeys Assumption 2 then we have the following relationship between the diagnosis result $\mathcal{DR}_{\mathcal{N}}(\mathcal{O})$ derived based on the set of complete explanations and the diagnosis result $\underline{\mathcal{DR}}_{\mathcal{N}}(\mathcal{O})$ derived based on the set of minimal explanations:*

$$\begin{tikzcd}
\mathcal{L}_{\mathcal{N}}(\mathcal{O}_n) \arrow[r, phantom, "\supseteq"] \arrow[d] & \underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_n) \arrow[d] \\
\Pi_{\mathcal{T}_f}(\mathcal{L}_{\mathcal{N}}(\mathcal{O}_n)) \arrow[r, phantom, "\supseteq"] \arrow[d] & \Pi_{\mathcal{T}_f}(\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_n)) \arrow[d] \\
\mathcal{DR}_{\mathcal{N}}(\mathcal{O}_n) \arrow[r, phantom, "\sim_F"] & \underline{\mathcal{DR}}_{\mathcal{N}}(\mathcal{O}_n)
\end{tikzcd}$$

$\mathcal{DR}_{\mathcal{N}}(\mathcal{O}_n)$	$\sim_{\texttt{F}}$	$\underline{\mathcal{DR}}_{\mathcal{N}}(\mathcal{O}_n)$
$\{\texttt{N}\}$	$\Rightarrow$	$\{\texttt{N}\}$
$\{\texttt{N},\texttt{UF}\}$	$\Leftarrow$	$\{\texttt{N}\}$
$\{\texttt{UF}\}$	$\Rightarrow$	$\{\texttt{N},\texttt{UF}\}$
$\{\texttt{UF}\}$	$\Leftarrow$	$\{\texttt{UF}\}$
$\{\texttt{F}\}$	$\Leftrightarrow$	$\{\texttt{F}\}$

Figure 4.3:

Proof. Consider the observation event $\mathcal{O}_1 = t_1^o \dots t_n^o$, and consider then the set of configurations $\mathcal{C}(t^o)$ in $\mathcal{U}_{\mathcal{N}}(\mathcal{O}_n)$. We have that a fault t_f is diagnosed that for sure happened based on the received observation $\mathcal{O}_n$ iff $\forall C \in \mathcal{C}(t^o)$, $\exists e \in E_C$ s.t. $\phi(e) = t_f$ and $e \preceq e_q^o$ for some event $e_q^o \in E_C$ that corresponds with an event that was observed ($\phi(e_q^o) = t_q^o$, $1 \leq q \leq n$).

This is because by Assumption 2 in any reachable marking at least a non-fault event is enabled thus the necessary condition for a fault event t_f to be diagnosed that for sure happened is that for every configuration $C \in \mathcal{C}(t^o)$ there exists at least an event e that is the image of t_f ($\phi(e) = t_f$) that is a predecessor ($e \preceq e_q^o$) of an observed event e_q^o.

Hence by deriving only the set of minimal configuration (explanations) $\underline{\mathcal{C}}(t^o)$ all the faults that can be diagnosed that for sure have happened are also detected. Thus $\mathcal{DR}_{\mathcal{N}}\{\texttt{F}\} \Leftrightarrow \underline{\mathcal{DR}}_{\mathcal{N}}\{\texttt{F}\}$. The other relations between $\mathcal{DR}_{\mathcal{N}}$ and $\underline{\mathcal{DR}}_{\mathcal{N}}$ are trivial. □

Thus the computation of $\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_n)$ is sufficient for taking the same control decisions (via $\Phi(\cdot)$) as the ones that would have been taken considering $\mathcal{L}_{\mathcal{N}}(\mathcal{O}_n)$ provided one takes actions only when the fault is certain.

$\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_n)$ is in general a lot smaller than $\mathcal{L}_{\mathcal{N}}(\mathcal{O}_n)$, thus the efficiency relies on the computational effort for enumerating backwards the set of minimal explanations.

Even though the computational effort for deriving $\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_n)$ is not comparable with the computational effort for deriving $\mathcal{L}_{\mathcal{N}}(\mathcal{O}_n)$ (since the forward respectively the backward search explore different state spaces), the efficiency of the diagnosis algorithm based on the (backward) calculation of the minimal explanations of the plant observation can be further improved if:

- there is *a priori* knowledge of plant dynamics that allows the use of some heuristics to drive the backward search [FRSB02]

- there are no subnets $\mathcal{N}''$ of the PN model $\mathcal{N}$ having a large size and comprising only unobservable events ($\forall \mathcal{N}'' \subseteq \mathcal{N}, \mathcal{T}'' \subset \mathcal{T}_{uo}$, then $\mid \mathcal{T}' \mid$ is not big)
- the observation is deterministic or the maximal degree of nondeterminism of the observation is not very high (say there are at most m transitions that share the same label via l and m is small)

4.2.3 The case of PNs with unobservable trap circuits

In this section we treat the case when all the unobservable circuits in the PN model are traps (see Def. 18) showing that this class of PNs allows for fast computations.

Theorem 1. *Consider a trap circuit PN $\langle \mathcal{N}, M_0 \rangle$. Then, given a trace σ that is legal from the initial marking M_0, $\sigma \in \mathcal{L}_{\mathcal{N}}(M_0)$ we have that:*

$$\sigma' \in \mathcal{L}_{\mathcal{N}}(M_0) \textit{ and } \overrightarrow{\sigma'} < \overrightarrow{\sigma} \textit{ together imply that}$$

$$\exists\, \sigma'' \textit{ s.t. } i)\ \sigma'\sigma'' \in \mathcal{L}_{\mathcal{N}}(M_0) \textit{ and } ii)\ \overrightarrow{\sigma'} + \overrightarrow{\sigma''} = \overrightarrow{\sigma}$$

(where $\sigma'\sigma''$ is the trace obtained by catenation of σ' and σ'').

To prove Theorem 1 we need the following result that can be found as Theorem 17 in [Mur89].

Theorem 2. *([Mur89]) In a trap-circuit net $\mathcal{N}$, M_d is reachable from M_0 iff:*

$i)$ there exists $\overrightarrow{\sigma}$ a non-negative integer solution of the state-equation Eq. 2.1

$ii)$ and $\langle \mathcal{N}_{\overrightarrow{\sigma}}, M_{0_{\overrightarrow{\sigma}}} \rangle$ has no token-free siphons

where $\mathcal{N}_{\overrightarrow{\sigma}}$ denotes the sub-net of $\mathcal{N}$ consisting of transitions t s.t. $\overrightarrow{\sigma}(t) > 0$ together with their input and output places and $M_{0_{\overrightarrow{\sigma}}}$ denotes the sub-vector of M_0 for places in $\mathcal{N}_{\overrightarrow{\sigma}}$.

Proof. [Theorem 1] Since $\sigma \in \mathcal{L}_{\mathcal{N}}(M_0)$ denote by M_d the marking obtained firing σ from M_0 ($M_0 \xrightarrow{\sigma} M_d$). Then we have that $\exists \overrightarrow{\sigma''}$ s.t.:

$$M_0 + F \cdot \overrightarrow{\sigma'} + F \cdot \overrightarrow{\sigma''} = M_d$$

$\sigma' \in \mathcal{L}_{\mathcal{N}}(M_0)$ and $M_0 \xrightarrow{\sigma'} M'$ imply that:

$$M' + F \cdot \overrightarrow{\sigma''} = M_d$$

To prove that exists a legal trace σ'' that can be executed from M' we need to prove that $\langle \mathcal{N}_{\vec{\sigma}''}, M'_{\vec{\sigma}''} \rangle$ has no token-free siphons where $\mathcal{N}_{\vec{\sigma}''}$ is the sub-net of $\mathcal{N}$ consisting of transitions that are executed in σ'' together with their input and output places and $M'_{\vec{\sigma}''}$ is the sub-vector marking of M' for places in $\mathcal{N}_{\vec{\sigma}''}$.

Consider the set of transitions $\Sigma(\sigma)$ that appear at least once in σ and let $\Sigma(\sigma)$ be partitioned in disjoint subsets as follows:

- T_1 is the set of transitions that appear in σ' and do not appear in σ'', i.e. $t \in T_1 \Rightarrow \vec{\sigma}'(t) > 0$ and $\vec{\sigma}''(t) = 0$

- T_{12} is the set of transitions that appear in both σ' and σ'', i.e. $t \in T_{12} \Rightarrow \vec{\sigma}'(t) > 0$ and $\vec{\sigma}''(t) > 0$

- T_2 is the set of transitions that appear in σ'' and do not appear in σ', i.e. $t \in T_2 \Rightarrow \vec{\sigma}''(t) > 0$ and $\vec{\sigma}'(t) = 0$.

(obviously $\Sigma(\sigma) = T_1 \cup T_{12} \cup T_2$ and $T_1 \cap T_2 = T_{12} \cap T_1 = T_{12} \cap T_2 = \emptyset$)

Consider a set of places Q in the sub-net $\langle \mathcal{N}_{\vec{\sigma}''}, M'_{\vec{\sigma}''} \rangle$ s.t. Q is a siphon in $\langle \mathcal{N}_{\vec{\sigma}''}, M'_{\vec{\sigma}''} \rangle$, i.e. ${}^\bullet Q \cap \{T_2 \cup T_{12}\} \subseteq Q^\bullet \cap \{T_2 \cup T_{12}\}$.

Assume that Q is token-free in the marking that results after firing σ' from M_0, i.e. $M'_{\vec{\sigma}'}(Q) = 0$. Then consider that there exists a place p in Q s.t. p belongs also to $\langle \mathcal{N}_{\vec{\sigma}'}, M_{0_{\vec{\sigma}'}} \rangle$ that is $p \in Q$ s.t. $\exists t \in T_1 \cup T_{12}$ s.t. $p \in {}^\bullet t$ or $p \in t^\bullet$.

If such a place does not exist then Q is a siphon in $\langle \mathcal{N}_{\vec{\sigma}}, M_{0_{\vec{\sigma}}} \rangle$ that means that Q is marked in the initial marking M_0 otherwise it contradicts the fact that $\sigma \in \mathcal{L}_{\mathcal{N}}(M_0)$ is legal. Since the transitions that are executed in σ' have no input places in Q the marking $M'_{\vec{\sigma}'}(Q)$ can not become 0, hence it contradicts the assumption that Q is token-free siphon in the marking that results after firing σ' from M_0, i.e. $M'_{\vec{\sigma}'}(Q) = 0$.

It means that there exists a place in p and a transition $t \in T_1 \cup T_{12}$ such that $p \in {}^\bullet t$ or $p \in t^\bullet$.

We have that $M'_{\vec{\sigma}''}(p) = 0$ and σ' is legal, thus:

$$\sum_{t \in {}^\bullet p} \vec{\sigma}'(t) + M_0(p) = \sum_{t \in p^\bullet} \vec{\sigma}'(t) \tag{4.5}$$

that in words means that the number of executions of the transitions that remove tokens from p in σ' is equal with the number of tokens plus the number of executions of transitions in σ' that add tokens in p.

Similarly for $M_0 \xrightarrow{\sigma} M_d$ we have that $M_d(p) \geq 0$ and:

$$\sum_{t \in {}^{\bullet}p} \vec{\sigma}(t) + M_0(p) \geq \sum_{t \in p^{\bullet}} \vec{\sigma}(t) \tag{4.6}$$

We have that:

1. if $t \in T_1$ then $\sigma(t) = \sigma'(t)$ and $\sigma''(t) = 0$
2. if $t \in T_{12}$ then $\sigma(t) = \sigma'(t) + \sigma''(t)$, $\sigma'(t) > 0$ and $\sigma''(t) > 0$
3. if $t \in T_2$ then $\sigma(t) = \sigma''(t)$ and $\sigma'(t) = 0$

From 4.5 and 4.6 we obtain:

$$\sum_{t \in {}^{\bullet}p} \overrightarrow{\sigma''}(t) \geq \sum_{t \in p^{\bullet}} \overrightarrow{\sigma''}(t) \tag{4.7}$$

Since p is in $\langle \mathcal{N}_{\vec{\sigma'}}, M'_{\vec{\sigma'}} \rangle$ and Q is a siphon (${}^{\bullet}Q \cap \{T_2 \cup T_{12}\} \subseteq Q^{\bullet} \cap \{T_2 \cup T_{12}\}$) we have that there exists at least a transition t in $T_2 \cup T_{12}$ s.t. $t \in {}^{\bullet}p$. Since Q is siphon and $T_2 \cup T_{12}$ then the input places of t are in Q.

We have that 4.7 holds for any place of the sihpon $p' \in Q$ and thus in particular for the input places of transition t. Then inductively we have that p is a part of a circuit in Q and in $\mathcal{N}$ we have that any circuit is a trap.

Thus if $M_0(p) \neq 0$ or $\sigma'(t) > 0$ for some transition $t \in {}^{\bullet}p$ then the statement is proved straightforward since a trap once is marked cannot become token-free in a successor marking.

Notice that if $M_0(p) = 0$ and $\sigma'(t) = 0$ for any transition $t \in {}^{\bullet}p$ implies that $\sigma'(t') = 0$ for any transition $t' \in p^{\bullet}$ that means that p does not belong to $\mathcal{N}_{\vec{\sigma'}}$ that contradicts the assumption.

Thus Q contains tokens in $M'_{\vec{\sigma''}}$ that means that the siphon Q is not token-free. Hence $M_{\vec{\sigma}}$ is reachable from $M'_{\vec{\sigma''}}$. This means that exists a trace σ'' that can be executed after σ' and this completes the proof. □

Then we have the following corollary:

Corollary 1. *Consider a trap circuit PN $\langle \mathcal{N}, M_0 \rangle$. Then, given two traces σ_1 and σ_2 that are legal from the initial marking M_0, $\sigma_1 \in \mathcal{L}_{\mathcal{N}}(M_0)$ and $\sigma_2 \in \mathcal{L}_{\mathcal{N}}(M_0)$ we have that:*

$\sigma_1\sigma_2 \in \mathcal{L}_{\mathcal{N}}(M_0)$ *implies that*

$\exists \sigma'_1$ *s.t.* $\sigma_2\sigma'_1 \in \mathcal{L}_{\mathcal{N}}(M_0)$ *and* $\overrightarrow{\sigma'_1} = \overrightarrow{\sigma_1}$

Proof. Straightforward applying Theorem 1. □

Assumption 3. *All the unobservable circuits in the PN model of the plant are trap circuits.*

Based on Assumption 3 and Theorem 1 we have the following result:

Proposition 8. *Consider a PN $\langle \mathcal{N}, M_0 \rangle$ satisfying Assumption 3. Then, given two unobservable strings $\sigma_{uo_1}, \sigma_{uo_2} \in \mathcal{T}_{uo}^*$ that are both legal from the initial marking M_0 ($\sigma_{uo_1}, \sigma_{uo_2} \in \mathcal{L}_{\mathcal{N}}(M_0)$), if $\sigma_{uo_1}\sigma_{uo_2} \in \mathcal{L}_{\mathcal{N}}(M_0)$ then $\exists \sigma_{uo_2}\sigma'_{uo_1} \in \mathcal{L}_{\mathcal{N}}(M_0)$ such that $\vec{\sigma}_{uo_1} = \vec{\sigma'}_{uo_1}$.*

Proof. Straightforward applying Corollary 1 to $\langle \mathcal{N}_{uo}, M_0^{uo} \rangle$ where $\mathcal{N}_{uo}$ denotes the sub-net of $\mathcal{N}$ comprising the unobservable transitions $\mathcal{T}_{uo}$ and M_0^{uo} denotes the sub-vector of M_0 for places in $\mathcal{N}_{uo}$. □

The idea behind developing an efficient algorithm is to convert the initial problem P of finding the set of minimal explanations of t^o into a conjunction of subproblems of the initial problem e.g. $P \longrightarrow \bigwedge_{v \in \mathcal{V}} SP_v$ where a subproblem SP_v, $v \in \mathcal{V}$ is to find a minimal explanation of how a token gets to an input place p of t^o.

Consider that $^\bullet t^o = \{p_1, p_2\}$ and denote by $\overleftarrow{\mathcal{C}}(p_1)$ the set of backward configurations derived considering the final marking $M_{fin} = p_1$ where a backward configuration $\overleftarrow{C}(p_1) \in \overleftarrow{\mathcal{C}}(p_1)$ contains only unobservable transitions.

Then denote by $\underline{\mathcal{C}}(p_1)$ the set of minimal configurations that explain the presence of one token on p_1 where a minimal configuration $\underline{C}(p_1) \in \underline{\mathcal{C}}(p_1)$ is obtained from a backward configuration $\overleftarrow{C}(p_1) \in \overleftarrow{\mathcal{C}}(p_1)$ by adding the conditions that correspond with the tokens from the initial marking that were not used.

Denote by $\overleftarrow{\mathcal{C}}(p_2, \underline{C}(p_1))$ the set of backward configurations that explain the presence of one token in p_2 considering as initial marking $mark(\underline{C}(p_1))$ and then denote by $\underline{\mathcal{C}}(p_2, \underline{C}(p_1))$ the set of minimal configurations that explain one token p_2 provided that one token in p_1 was explained by the minimal configuration $\underline{C}(p_1)$.

Denote by $\underline{\mathcal{C}}(p_1, p_2)$ the set of all minimal configurations that explain first one token in p_1 and then one token in p_2:

$$\underline{\mathcal{C}}(p_1, p_2) = \{\underline{\mathcal{C}}(p_2, \underline{C}(p_1)) \mid \underline{C}(p_1) \in \underline{\mathcal{C}}(p_1)\}$$

and then denote by $\underline{\mathcal{L}}_{\mathcal{N}}(p_1, p_2)$ the set of minimal traces that explain one token in p_1 and then one token in p_2:

$$\underline{\mathcal{L}}_{\mathcal{N}}(p_1, p_2) = \{\tau \mid \tau = \phi(\sigma) \wedge \sigma \in \underline{\mathcal{E}}(p_1 p_2)\}$$

where:

$$\underline{\mathcal{E}}(p_1, p_2) = \left\{\sigma \in \langle E_{\underline{C}(p_1,p_2)} \rangle \mid \underline{C}(p_1, p_2) \in \underline{\mathcal{C}}(p_1, p_2)\right\}$$

Similarly denote by $\underline{\mathcal{C}}(p_2, p_1)$ the set of all minimal configurations that explain first one token in p_2 and then one token in p_1:

$$\underline{\mathcal{C}}(p_2, p_1) = \{\underline{\mathcal{C}}(p_1, \underline{C}(p_2)) \mid \underline{C}(p_2) \in \underline{\mathcal{C}}(p_2)\}$$

and then denote by $\underline{\mathcal{L}}_{\mathcal{N}}(p_2, p_1)$ the set of minimal traces that explain one token in p_2 and then one token in p_1:

$$\underline{\mathcal{L}}_{\mathcal{N}}(p_2, p_1) = \{\tau \mid \tau = \phi(\sigma) \wedge \sigma \in \underline{\mathcal{E}}(p_2 p_1)\}$$

where:

$$\underline{\mathcal{E}}(p_2, p_1) = \left\{\sigma \in \langle E_{\underline{C}(p_2,p_1)}\rangle \mid \underline{C}(p_2, p_1) \in \underline{\mathcal{C}}(p_1, p_2)\right\}$$

Based on Proposition 8 we have that:

$$\underline{\mathcal{L}}_{\mathcal{N}}(p_1, p_2) \equiv_{\Sigma_\mu} \underline{\mathcal{L}}_{\mathcal{N}}(p_2, p_1)$$

that is:

1. $\forall \tau \in \underline{\mathcal{L}}_{\mathcal{N}}(p_1, p_2) \Rightarrow \exists \tau' \in \underline{\mathcal{L}}_{\mathcal{N}}(p_2, p_1)$ s.t. $\Sigma_\mu(\tau) = \Sigma_\mu(\tau')$
2. and $\forall \tau' \in \underline{\mathcal{L}}_{\mathcal{N}}(p_2, p_1) \Rightarrow \exists \tau' \in \underline{\mathcal{L}}_{\mathcal{N}}(p_1, p_2)$ s.t. $\Sigma_\mu(\tau') = \Sigma_\mu(\tau)$

This result can be extended inductively for the case when t^o has an arbitrary number of input places and also for any order in which the input places of an unobservable transition are chosen.

Proposition 9. *Denote by $\mathfrak{X}$ set of all possible orders of explaining the tokens in the output places of the transitions that are considered. Then we have that for any two arbitrary orders $\forall \chi, \chi' \in \mathfrak{X}$:*

$$\underline{\mathcal{L}}_{\mathcal{N}}(\chi, t^o) \equiv_{\Sigma_\mu} \underline{\mathcal{L}}_{\mathcal{N}}(\chi', t^o)$$

that obviously implies that:

$$\underline{\mathcal{L}}_{\mathcal{N}}(\chi, t^o) \equiv_{\Sigma_\mu} \underline{\mathcal{L}}_{\mathcal{N}}(t^o)$$

where $\underline{\mathcal{L}}_{\mathcal{N}}(t^o) = \bigcup_{\chi \in \mathfrak{X}} \underline{\mathcal{L}}_{\mathcal{N}}(t^o)$

Proof. Straightforward applying Proposition 8. □

Proposition 10. *Consider a PN $\langle \mathcal{N}, M_0 \rangle$ satisfying Assumption 3 and the first observed event in the plant t_1^o. Then, given two unobservable strings $\sigma_{uo_1}, \sigma_{uo_2} \in \mathcal{T}_{uo}^*$ that are both legal from the initial marking M_0 ($\sigma_{uo}, \sigma'_{uo} \in \mathcal{L}_{\mathcal{N}}(M_0)$), s.t.:*

$$M_0 \xrightarrow{\sigma} M \geq Pre(\cdot, t_1^o)$$

$$M_0 \xrightarrow{\sigma'} M' \geq Pre(\cdot, t_1^o)$$

and $\overrightarrow{\sigma'_{uo}} < \overrightarrow{\sigma_{uo}}$

all together imply that exists an unobservable string $\exists \sigma''_{uo} \in \mathcal{T}^*_{uo}$ *s.t.* $i)$ $\sigma'_{uo}\sigma''_{uo} \in \mathcal{L}_{\mathcal{N}}(M_0)$ *and* $ii)$ $\overrightarrow{\sigma'_{uo}} + \overrightarrow{\sigma''_{uo}} = \overrightarrow{\sigma_{uo}}$

Proof. Straightforward applying Theorem 1. □

Consider $\tau \in \underline{\mathcal{L}}_{\mathcal{N}}(t^o)$ a minimal explanation of t^o where $\tau = \tau_1\tau_2$ s.t. $M' \overset{\tau_2}{\rightsquigarrow} M''$ with $M' = Pre(\cdot, t^o)$ and $M'' \leq M_0$. Based on Proposition 8 we have that $\exists \tau' = \tau_2\tau'_1$ s.t. $\tau' \in \underline{\mathcal{L}}_{\mathcal{N}}(t^o)$ and $\Sigma_\mu(\tau_1) = \Sigma_\mu(\tau'_1)$.

It means that when we find that a trace e.g. τ_2 is a minimal explanation for t^o then we do not calculate the minimal explanations $\tau = \tau_1\tau_2$ that are backwards extensions of τ_2 since for any τ_1 that extends τ_2 backwards providing a minimal explanation τ for t^o we have that there exists a trace τ'_1 that can be executed after τ_2 ($\tau' = \tau_2\tau'_1$) s.t. τ_1 and τ'_1 have the same Parikh vector.

Thus considering an arbitrary order $\chi \in \mathfrak{X}$ we can calculate a subset of minimal explanations $\underline{\mathcal{L}}_{\mathcal{N}}(\chi, t^o) \subset \underline{\mathcal{L}}_{\mathcal{N}}(t^o)$ s.t. $UR_{\mathcal{N}}(\underline{\mathcal{M}}(\chi, t^o)) = \underline{\mathcal{M}}(t^o))$ where:

$$\underline{\mathcal{M}}(\chi, t^o) = \left\{ M_\chi \mid \exists \tau_\chi \in \underline{\mathcal{L}}_{\mathcal{N}}(\chi, t^o) \text{ s.t. } M_0 \xrightarrow{\tau_\chi} M_\chi \right\}$$

$$\underline{\mathcal{M}}(t^o) = \left\{ M' \mid \exists \tau \in \underline{\mathcal{L}}_{\mathcal{N}}(t^o) \text{ s.t. } M_0 \xrightarrow{\tau} M \right\}$$

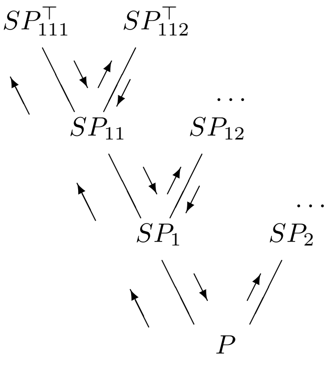


Figure 4.4:

The search algorithm is depth first, meaning that (see Fig. 4.4) for a given case first P is converted into $SP_1 \wedge SP_2$ then the subproblem SP_1 is selected and it is converted in a similar way ($SP_1 \rightarrow SP_{11} \wedge SP_{12}$) and then ($SP_{11} \rightarrow SP_{111} \wedge SP_{112}$). Assume that SP_{111} is proven (denoted $SP^{\top}_{111}$). Then we backtrack to the last node (SP_{11}) that is not proven yet and prove SP_{112}. If

SP_{112} is proven ($SP_{112}^{\top}$) then SP_{11} becomes proved and we backtrack to the last node left not proven.

A subproblem SP_ν is a terminal node if $SP_\nu = \{\text{token in p}\}$ and p is marked in M_0. If the marking (the multi-set of tokens) that corresponds with the terminal nodes $M^{\top}$ is smaller than M_0 then P has a solution for this case and the computation stops for this case (and continues for the other cases that correspond with the backward choice that are considered). To motivate this consider that a minimal explanation σ'_{uo} was found. By Proposition 10 we have that for all the minimal explanations σ_{uo} of the first observed event such that $\vec{\sigma}_{uo} > \vec{\sigma'}_{uo}$ there exists σ''_{uo} that can be executed after σ'_{uo} and σ''_{uo} contains the unobservable transitions that are considered in σ_{uo} but are not considered in σ'_{uo}.

This is important for the following two reasons:

1. first the consideration of all the extensions of σ'_{uo} (e.g. σ''_{uo}) will not change the set of faults that were diagnosed that for sure happened

2. secondly the marking that is reachable firing σ is reachable from the marking that result after firing σ'_{uo} from the initial marking.

A node $SP_\nu = \{\text{token in } p\}$ is aborted in the following two cases:

1. ${}^{\bullet}p \cap \mathcal{T}_{uo} = \emptyset$ and either p is not marked in M_0 or $M^{\top} > M_0$

2. $SP_\nu = \{t \quad \text{fired }\}$ has a predecessor subproblem $SP_\iota = \{t' \quad \text{fired }\}$ and $t = t'$

The second condition says that SP_ν cannot be part of the proof of SP_ι if SP_ν and SP_ι consist in the firing of the same transition. This is because the problem to be proven becomes harder since to prove the firing of t we must prove the firing of $t' = t$ and possibly something else. In other words by firing backwards the unobservable string σ_{uo_2} from M e.g. $M_{fin} \overset{\sigma_{uo_1}}{\rightsquigarrow} M \overset{\sigma_{uo_2}}{\rightsquigarrow} M'$ we have that $M' \geq M$. If $M' = M$ then σ_{uo_2} is a cycle that repeats the marking and consequently it is not continued while if $M' > M$ the marking decreases and obviously the further calculation of $\sigma_1 \sigma_{uo_2}$ is discarded. This is because σ_{uo_2} can be executed after σ_{uo_1}.

If a node SP_ν is aborted the entire case is aborted.

We formalize this intuitive algorithm using the (backwards) unfolding technique.

Algorithm 4 Minimal explanations for one observed event

```
Require: $\langle \mathcal{N}, M_0 \rangle$, $\mathcal{T}_o$, $\mathcal{T}_{uo}$, $t^o$
Ensure: $\underline{\mathcal{C}}'(\chi, t^o)$
$\nu = 1; \nu_{max} = 1; \underline{C}_\nu = \emptyset; SET[\nu] = \emptyset; Pred[\nu] = \emptyset; t_{cur_\nu} = t^o$
for all $p \in \mathcal{P}$ s.t. $M_0(p) \neq 0$ do
  for each token in $M_0(p)$ do
    $\underline{B}_{C_\nu} = \underline{B}_{C_\nu} \cup b$ and $\phi(b) = p$
    add $b$ to $AVAILABLE[\nu]$
  end for
end for
while $\nu \leq \nu_{max}$ do
  $Min\_Conf(t_{cur_\nu})$ (see Section 7.3 in Apendix)
  if $abort[\nu] \neq true$ then
    $\underline{\mathcal{C}}'(t^o) = \underline{\mathcal{C}}'(t^o) \cup \underline{C}_\nu$
  end if
  $\nu = \nu + 1$
end while
```

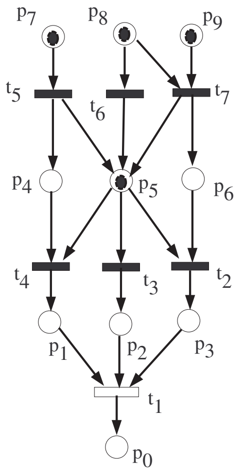


Figure 4.5:

Example 20. *Consider the PN $\mathcal{N}'$ displayed in 4.5 and consider that t_1 is the only observable transition in $\mathcal{N}'$ (where $\mathcal{N}'$ may be viewed as a proper subnet of a larger PN model $\mathcal{N}$). Let t_1 be observed. The algorithm for computing the minimal explanations described above works in the following way (see Fig. 4.6 from left to right):*

> *first the top condition – nodes b_7, b_5, b_8, b_9 that correspond to M_0 are added to $\underline{C}_1$ and then $^\bullet e_1 \cup e_1 \cup e_1^\bullet$ is added to $\underline{C}_1$ where $\phi(e_1 = t_1)$*

- $AVAILABLE[1] = \{b_7, b_5, b_8, b_9\}$

- $SET[1] = \{b_1, b_2, b_3\}$ *(in an arbitrary order)*

- b_1 *is chosen arbitrary as* $HEAD(SET[1])$

- e_4 *is appended and an arc is drawn from* e_4 *to* b_1

- b_4 *is appended and added to* $SET[1]$ *s.t.* $HEAD(SET[1]) = b_4$

- b_5 *is as input place of* e_4 *and removed from* $AVAILABLE[1]$

- *then for explaining* b_4*,* e_5 *is appended and an arc is drawn from* b_7 *to* e_5

- b_7 *is removed from* $AVAILABLE[1]$*;*

- b'_5 *is added to* $AVAILABLE[1]$*;*

- $b_2 = HEAD(SET[1])$ *and remove* b_2 *from* $SET[1]$

- e_3 *is appended*

- b'_5 *is removed from* $AVAILABLE[1]$*;*

- $b_3 = HEAD(SET[1])$

- *append* e_2

- *add* b''_5 *and* b_6 *to* $SET[1]$

- *chose* e_7 *for* $\underline{C}_1$ *and* e_4 *for* $\underline{C}_2$

- *append* e_7

- b''_5 *and* b_6 *from* $SET[1]$

- *remove* b_9 *from* $AVAILABLE[1]$

- $SET[1] = \emptyset$ *so that we obtain the solution displayed in Fig. 4.6-right*

- *for* $\underline{C}_2$ *there is not solution since* t_6 *and* t_7 *can not both be executed*

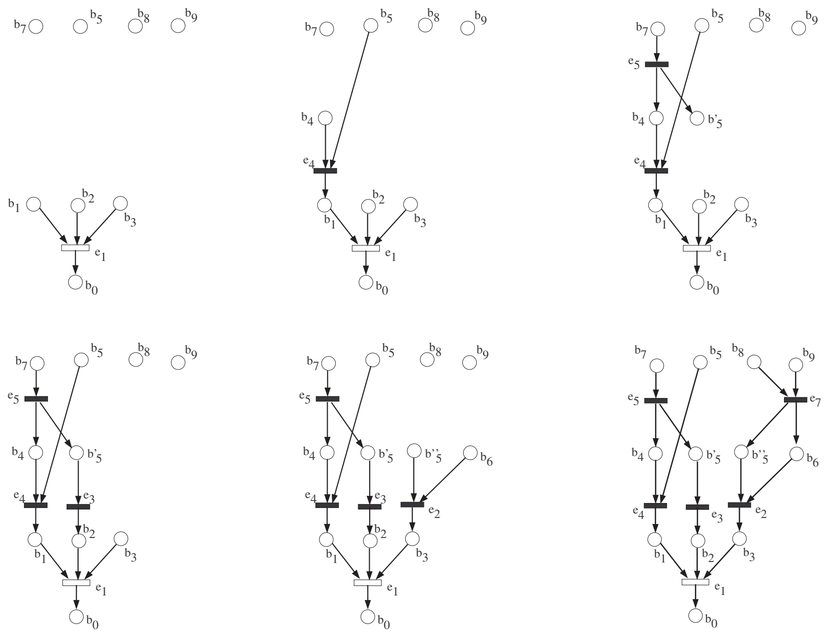


Figure 4.6:

For a sequence of observed events $\mathcal{O}_n = t_1^o \dots t_n^o$ Algorithm 4 is applied recursively where for explaining the k^{th} observed event the initial marking is given by the set of markings that are obtained after firing the minimal explanations of the first $k-1$ observed events.

Algorithm 5 Minimal explanations for a sequence of observed events

```
Require: $\langle \mathcal{N}, M_0 \rangle$, $\mathcal{T}_o$, $\mathcal{T}_{uo}$, $\mathcal{O}_n$
Ensure: $\underline{\mathcal{C}}'(\chi_n, \mathcal{O}_n)$
$k = 1$;
run Algorithm 4 for $\mathcal{O}_1 = t_1^o$
while $k \leq n$ do
    for all $\underline{\mathcal{C}}'(\chi_k, \mathcal{O}_k) \in \underline{\mathcal{C}}'(\chi_k, \mathcal{O}_k)$ do
        run Alg.4 for $t_{k+1}^o$ considering instead of $M_0$
        the marking $mark(\underline{\mathcal{C}}'(\chi_k, \mathcal{O}_k))$
    end for
    $k = k + 1$
end while
```

Denote by $\underline{\mathcal{C}}'(\chi_n, \mathcal{O}_n)$ the set of the minimal configurations derived as presented above. Notice that $\underline{\mathcal{C}}'(\chi_n, \mathcal{O}_n)$ does not comprise in general all the minimal configurations of $\mathcal{O}_n$ for the reasons explained above. Thus we have

$\underline{\mathcal{C}}'(\chi_n, \mathcal{O}_n) \subseteq \underline{\mathcal{C}}(\chi_n, \mathcal{O}_n)$ and $\underline{\mathcal{E}}'(\chi_n, \mathcal{O}_n) \subseteq \underline{\mathcal{E}}(\chi_n, \mathcal{O}_n)$ where:

$$\underline{\mathcal{E}}'(\chi_n, \mathcal{O}_n) = \left\{ \langle E_{\underline{C}'(\chi_n, \mathcal{O}_n)} \rangle \mid \underline{C}(\chi_n, \mathcal{O}_n) \in \underline{\mathcal{C}}'(\chi_n, \mathcal{O}_n) \right\}$$

and $\underline{\mathcal{L}}'_{\mathcal{N}}(\chi_n, \mathcal{O}_n) = \{\tau \mid \exists \sigma \in \underline{\mathcal{E}}(\chi_n, \mathcal{O}_n) \wedge \phi(\sigma) = \tau\}$.

Proposition 11. *Given a PN $\langle \mathcal{N}, M_0 \rangle$ and the observation $\mathcal{O}_n$ let $\underline{\mathcal{L}}'_{\mathcal{N}}(\chi_n, \mathcal{O}_n)$ be the set of minimal explanations derived running the Algorithm 5 for an arbitrary order $\chi_n \in \mathfrak{X}_n$. We have that:*

1. $\underline{\mathcal{L}}'_{\mathcal{N}}(\chi_n, \mathcal{O}_n) \subseteq \underline{\mathcal{L}}_{\mathcal{N}}(\chi_n, \mathcal{O}_n)$
2. $\forall \tau \in \underline{\mathcal{L}}_{\mathcal{N}}(\chi_n, \mathcal{O}_n) \Rightarrow \exists \tau' \in \underline{\mathcal{L}}'_{\mathcal{N}}(\chi_n, \mathcal{O}_n)$ *s.t.* $\vec{\tau}' \leq \vec{\tau}$
3. *and* $UR_{\mathcal{N}}(\underline{\mathcal{M}}'(\mathcal{O}_n)) = \underline{\mathcal{M}}(\mathcal{O}_n)$

Proof. The proof is by induction as follows.

$\mathcal{O}_1 = t_1^o$ (base) Let $\underline{\mathcal{L}}_{\mathcal{N}}(\chi_n, \mathcal{O}_1)$ be obtained running the Algorithm 5 for $M_{fin} = Pre(\cdot, t_1^o)$. The proof of $i)$ is straightforward. Then we have by Proposition 9 that $\underline{\mathcal{L}}_{\mathcal{N}}(\chi_1, \mathcal{O}_1) \equiv_{\Sigma_\mu} \underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_1)$. Then $\underline{\mathcal{L}}'_{\mathcal{N}}(\chi_1, \mathcal{O}_1) \subseteq \underline{\mathcal{L}}_{\mathcal{N}}(\chi_1, \mathcal{O}_1)$ and as implemented in Algorithm 4 we have that $\sigma_{uo} t_1^o \in \underline{\mathcal{L}}'_{\mathcal{N}}(\chi_1, \mathcal{O}_1)$ iff

$c.i)$ all the markings that are considered in τ are different ($\forall 0 \leq q < v \leq z \Rightarrow M_q \neq M_v$)

$c.ii)$ and $\forall q, 0 \leq q \leq z$, $M'_q \leq M_0 \Rightarrow q = 0$ where $M_{fin} \overset{\sigma_{uo}^{qz}}{\rightsquigarrow} M'_q$

with $M_{fin} = Pre(\cdot, t_1^o)$, $M_z \geq M_{fin}$ and:

$$\sigma_{uo} := M_0 \xrightarrow{t_1} \ldots \xrightarrow{t_q} M_q \ldots \xrightarrow{t_v} M_v \ldots \xrightarrow{t_z} M_z$$

Then consider a trace $\tau \in \underline{\mathcal{L}}_{\mathcal{N}}(\chi_1, \mathcal{O}_1) \setminus \underline{\mathcal{L}}'_{\mathcal{N}}(\chi_1, \mathcal{O}_1)$ that is discarded by running Algorithm 4.

If τ is discarded because of $c.i)$ we have two cases:

1. If $\exists M_q$ and M_v s.t. $M_q = M_v$ then we have that σ_{uo} contains a cycle thus $\exists \sigma'_{uo} t_1^o \in \underline{\mathcal{L}}'_{\mathcal{N}}(\chi_1, \mathcal{O}_1)$ s.t. $\Sigma_\mu(\sigma'_{uo}) \leq \Sigma_\mu(\sigma_{uo})$.
2. else if $\exists M_q$,M_v s.t. $M_q > M_v$ and $q < v$ then we have that $\sigma_{uo}^{vz} = t_{v+1} \ldots t_z$ can be executed from M_q thus $\sigma'_{uo} = t_1 \ldots t_q t_{v+1} \ldots t_z$ can be executed from M_0 and $\Sigma_\mu(\sigma'_{uo}) \leq \Sigma_\mu(\sigma_{uo})$.

If τ is discarded because of $c.ii)$ we have Proposition 10.

Then the proof that $UR_{\mathcal{N}}(\underline{\mathcal{M}}'(t_1^o)) = \underline{\mathcal{M}}(t_1^o)$ is straightforward by Proposition 10.

$\mathcal{O}_k$ (induction base) $\underline{\mathcal{L}}'_{\mathcal{N}}(\chi_k, \mathcal{O}_k) \subseteq \underline{\mathcal{L}}_{\mathcal{N}}(\chi_k, \mathcal{O}_k)$; $\forall \tau \in \underline{\mathcal{L}}_{\mathcal{N}}(\chi_k, \mathcal{O}_k) \Rightarrow \exists \tau' \in \underline{\mathcal{L}}'_{\mathcal{N}}(\chi_k, \mathcal{O}_k)$ s.t. $\Sigma_\mu(\tau') \subseteq \Sigma_\mu(\tau)$ and $UR_{\mathcal{N}}(\underline{\mathcal{M}}'(\mathcal{O}_k)) = \underline{\mathcal{M}}(\mathcal{O}_k)$.

$\mathcal{O}_{k+1}$ (induction step) The proof is straightforward since the $k+1^{th}$ observed event is explained considering as initial marking a marking $M \in \underline{\mathcal{M}}'(\chi_k, \mathcal{O}_k)$. □

Then we have the following result:

Proposition 12. *Consider a PN model that has the property that all the unobservable circuits in $\langle \mathcal{N}, M_0 \rangle$ are traps and any observation $\mathcal{O}_n$ that can be generated by the plant. The diagnosis result $\underline{\mathcal{DR}}'_{\mathcal{N}}(\mathcal{O}_n)$ based on the subset of minimal explanations $\underline{\mathcal{L}}'_{\mathcal{N}}(\chi_k, \mathcal{O}_n)$ (derived by running Algorithm 5) and the diagnosis result $\mathcal{DR}_{\mathcal{N}}(\mathcal{O}_n)$ based on the set of all the explanations of $\mathcal{O}_n$ are equivalent w.r.t. the detection of the faults that for sure happened, i.e.:*

$$\forall \mathcal{O}_n \qquad \underline{\mathcal{DR}}'_{\mathcal{N}}(\mathcal{O}_n) = \{F\} \Leftrightarrow \mathcal{DR}_{\mathcal{N}}(\mathcal{O}_n) = \{F\}$$

Proof. The proof is straightforward using Prop. 11 and Assumption 2. □

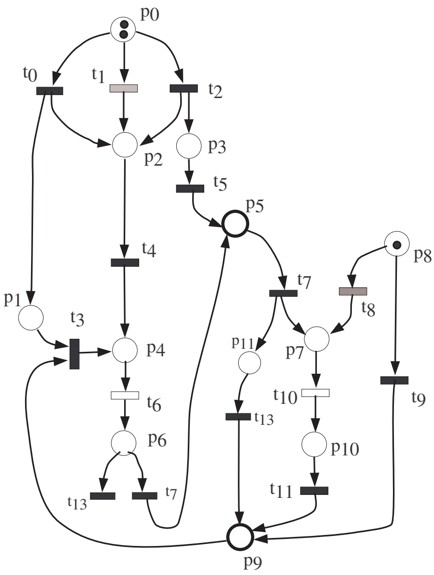


Figure 4.7:

Example 21. *Consider the PN $\langle \mathcal{N}, M_0 \rangle$ displayed in Fig. 4.7 where the observable transitions are t_6 and t_{10} and the fault transitions are t_1, t_8. The initial marking is $M_0 = \{m(p_0) = 2; m(p_8) = 1\}$. Let the first observed event be $\mathcal{O}_1 = t_6$. The set of all the explanations of $\mathcal{O}_1$ is:*

$$\mathcal{L}_{\mathcal{N}}(\mathcal{O}_1) = \{t_0t_4t_6; t_0t_0t_4t_6; t_0t_4t_6t_9; t_1t_4t_6t_0; \ldots\}$$

The set of minimal explanations of $\mathcal{O}_1$ is:

$$\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_1) = \{\tau_1 = t_0t_4t_6; \tau_2 = t_1t_4t_6; \tau_3 = t_2t_4t_6; \tau_4 = t_2t_5t_7t_{13}t_0t_3t_4t_6; \\ \tau_5 = t_0t_9t_3t_6\}$$

The diagnosis result $\underline{\mathcal{DR}}_{\mathcal{N}}(\mathcal{O}_n) = \{\texttt{UF}\}$ *since* $\underline{\mathcal{D}}_{\mathcal{N}}(\mathcal{O}_n) = \{\epsilon, t_1\}$ *that is also the case for* $\mathcal{DR}_{\mathcal{N}}(\mathcal{O}_n)$*. The set of markings calculated based on* $\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_1)$ $\underline{\mathcal{M}}(\mathcal{O}_n) = \{M_1, M_2, M_3, M_3\}$ *where* $M_0 \xrightarrow{\tau_\iota} M_\iota$ *for* $\iota = 1, \ldots, 5$ *and* $\tau_\iota \in \underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_1)$*:*

$$M_1 = \{p_0, p_1, p_6, p_8\}$$

$$M_2 = \{p_0, p_6, p_8\}$$

$$M_3 = \{p_0, p_3, p_6, p_8\}$$

$$M_4 = \{p_2, p_6, p_8\}$$

$$M_5 = \{p_0, p_6\}$$

Then consider the second observed event t_{10} *(*$\mathcal{O}_2 = t_6t_{10}$*). If we consider* $M_4 = \{p_2, p_6, p_8\}$ *as initial marking for minimally explaining* t_{10} *we obtain two minimal explanations for* t_{10} *e.g.* $\omega_1 = t_7t_{10}$ *and* $\omega_2 = t_8t_{10}$*.*

Remark 8. *Notice that a PN model free of unobservable circuits can be treated as an acyclic PN. Thus the marking equation (Eq.2.1) is a necessary and sufficient condition for the reachability analysis. For any explanation* $\tau \in \mathcal{L}_{\mathcal{N}}(t^o)$ *of the first observed event* t^o*,* $\tau = \sigma_{uo}t^o$ *we have that:*

$$M_0 + F \cdot \overrightarrow{\sigma_{uo}} \geq Pre(\cdot, t^o)$$

Thus the problem of deriving the set of minimal explanations can be formulated as a Multi Objective Integer Linear Programming Problem as follows:

Given the first observed event t_1^o *then* $\sigma_{uo}t^o$ *is a minimal explanation if:*

i) $\overrightarrow{\sigma_{uo}}$ *is a solution for 4.8*

$$F \cdot \overrightarrow{\sigma_{uo}} \geq Pre(\cdot, t^o) - M_0 \tag{4.8}$$

ii) and for any $\overleftarrow{\sigma'_{uo}}$ *that is solution for 4.8 then if* $\overrightarrow{\sigma'_{uo}} \leq \overrightarrow{\sigma_{uo}}$ *we have that* $\overrightarrow{\sigma'_{uo}} = \overrightarrow{\sigma_{uo}}$*.*

4.3 Distributed diagnosis

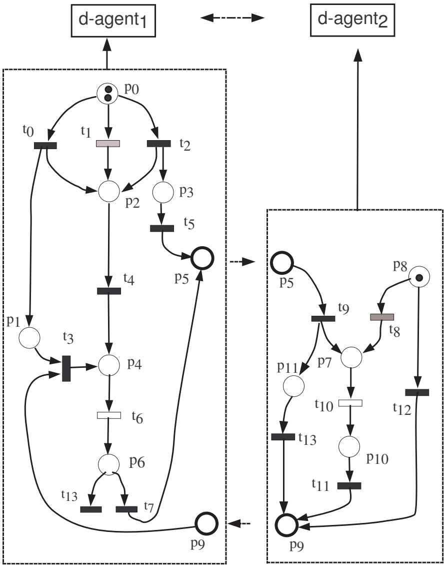


Figure 4.8:

The work in this section is inspired by the fundamental work of S. Lafortune and coworkers on distributed/decentralized diagnosis of DES [DLT00], [GL03], [DLT03] and also by the work of researchers in INRIA/IRISA Rennes [PCR01], [BFHJ03], [FBHJ05] and by R. Su and W.M. Wonham [SW04], [Su04].

We consider the distributed plant description as follows:

1. the overall plant is composed of a collection $J = \{1, 2, \ldots, \mid J \mid\}$ of components (local sites)

2. the PN model of the overall plant $\langle \mathcal{N}, M_0 \rangle$ is composed by a collection of bordered Petri Nets $\langle \mathcal{N}_i, M_{0_i} \rangle$, $i \in J$ (the interactions between components are modeled by the border places that allow tokens from one component to enter/exit the neighbouring components)

3. for each component $i \in J$ there is a local agent Ag_i that knows the PN model of the component i, $\langle \mathcal{N}_i, M_{0_i} \rangle$ and the disjoint partition of the local set of transitions $\mathcal{T}_i$ in to the set of locally observable transitions $\mathcal{T}_{o_i}$ and the set of unobservable transitions $\mathcal{T}_{uo_i}$

4. Ag_i knows also the set of border-places of its local PN model $\mathcal{N}_i$ knowing for each border place p if p is a place that allows tokens to enter

$\mathcal{N}_i$ or p is a place that allows tokens to leave $\mathcal{N}_i$ and moreover knows which are the components that can put tokens to p, respectively remove tokens from p

5. the occurrence of an observable transition $t_i^o \in \mathcal{T}_{o_i}$ is reported (only) to the local agent Ag_i correctly and without delays. The observation of t_i^o includes also the time t_i^o occurred in the plant, and this is measured with accuracy according with a global clock

Formally, the distributed plant description is:

i) $\mathcal{N} = \bigcup_{i \in J} \mathcal{N}_i$ where $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ and $\mathcal{N}_i = (\mathcal{P}_i, \mathcal{T}_i, F_i)$

ii) $\mathcal{P} = \bigcup_{i \in J} \mathcal{P}_i$, and $\forall i \in J, \exists j \in J, i \neq j$ s.t. $\mathcal{P}_i \cap \mathcal{P}_j \triangleq \mathcal{P}_{ij} \neq \emptyset$

iii) $\mathcal{T} = \bigcup_{i \in J} \mathcal{T}_i$ and $\forall i, j \in J, i \neq j \Rightarrow \mathcal{T}_i \cap \mathcal{T}_j = \emptyset$

iv) $F_i = F \mid_{\mathcal{N}_i}$

v) $\mathcal{P}_{ij} = \mathcal{P}_{IN_{ij}} \cup \mathcal{P}_{OUT_{ij}}, \mathcal{P}_{IN_{ij}} \cap \mathcal{P}_{OUT_{ij}} = \emptyset$

vi) $\mathcal{P}_{IN_{ij}} = \mathcal{P}_{OUT_{ji}} = \{p \in \mathcal{P}_{ij} \mid p^\bullet \subseteq \mathcal{T}_i \wedge {}^\bullet p \subseteq \mathcal{T}_j\}$

vii) $\mathcal{P}_{IN_{ji}} = \mathcal{P}_{OUT_{ij}} = \{p \in \mathcal{P}_{ji} \mid {}^\bullet p \subseteq \mathcal{T}_i \wedge p^\bullet \subseteq \mathcal{T}_j\}$

viii) $\mathcal{N}$ is structurally bounded w.r.t. the unobservable evolution i.e. $(\forall M \in \mathbb{N}^{|\mathcal{P}|})$, $(\forall \sigma_{uo} \in \mathcal{T}_{uo}^*)$ we have that:

$$((M \xrightarrow{\sigma_{uo}} M') \text{ and } (M' \geq M)) \Rightarrow (M' = M)$$

For simplicity we assume at item $v)$ above that $\mathcal{P}_{IN_{ij}}$ and $\mathcal{P}_{OUT_{ij}}$ are disjunct and we consider that in the initial marking of the overall plant the border places are not-marked that is $\forall i, j \in J$, $M_0(\mathcal{P}_{ij}) = 0$. Moreover to avoid unnecessary complications we consider that $\forall p \in \mathcal{P}$, ${}^\bullet p \subseteq \mathcal{T}_i$ for some $i \in J$ and similarly $\forall p \in \mathcal{P}, p^\bullet \subseteq \mathcal{T}_j$ for some $j \in J$ that is only a component can put and respectively remove tokens from any border place.

For a component i denote by $\mathcal{P}_{IN_i}$ and $\mathcal{P}_{OUT_i}$ the set of input border places respectively output border places:

$$\mathcal{P}_{IN_i} = \left\{\mathcal{P}_{IN_{ij}} \mid j \in J, j \neq i \wedge \mathcal{P}_{IN_{ij}} \neq \emptyset\right\} \text{ and}$$

$$\mathcal{P}_{OUT_i} = \left\{\mathcal{P}_{OUT_{ij}} \mid j \in J, j \neq i \wedge \mathcal{P}_{OUT_{ij}} \neq \emptyset\right\}$$

Given the set of agents $\mathcal{AG} = \{Ag_i \mid i \in J\}$, the knowledge an agent Ag_i has $KNW_i = \langle \mathcal{N}_i, \mathcal{T}_{o_i}, \mathcal{T}_{F_i}, M_{0_i}, \mathcal{P}_{IN_i}, \mathcal{P}_{OUT_i} \rangle$ considers that:

i) the plant observation is distributed $\mathcal{O}_{\theta_{com}} = \otimes^{gc}_{i\in J}\mathcal{O}^i_{\theta_{com}}$. $\mathcal{O}^i_{\theta_{com}} = t^o_{1_i} \dots t^o_{n_i}$ is the local observation recorded at site $i \in J$, where any observed event $\langle t^o_{k_i}, \theta_{k_i}\rangle$ has the time tag $\theta_{t_{k_i}}$ indicating the time event $t^o_{k_i}$ happened in the plant; $\theta_{t_{k_i}}$ is measured according with a global clock (denoted gc in short).

ii) the communication between agents is not event-driven that is the agents are allowed to communicate at times e.g. $\theta_{com_1}, \theta_{com_2}, \dots$ that do not necessarily depend on the plant observation. The communication session for information exchange at the time θ_{com_q} $(q = 1, \dots)$ comprises different pairwise communication rounds. A pairwise communication round between two neighbouring agents Ag_i and Ag_j consists of an instantaneously exchange of information between the two agents. The communication session is closed when the agents have no more information to exchange among them.

Problem formulation:

Given the setting described above, design a distributed algorithm such that:

R1 before communicating with its neighboring agents, each agent Ag_i $(i \in J)$ derives a local preliminary diagnosis of component i

R2 repeat for $q = 1, \dots q_{max}$: when the communication session is initiated (e.g. at the global time θ_{com_q} that does not necessarily depend on the time the observable events are reported) then for achieving the consistency of all the local calculations of agents $Ag_1, Ag_2, \dots Ag_{|J|}$ at the time θ_{com_q}:

R2.1 each local agent $Ag_i, i \in J$ derives the (limited) information that should be sent in a communication round to its neighboring agents

R2.2 after each communication round, the local calculation of site i is updated with the new information that is received

then each local agent iterates the step 2.1) and 2.2) until a stopping criterion is achieved (the communication session at time θ_{com_q} terminates)

R3 the completion of the communication session at the communication time θ_{com_q} guarantees that the agents recover the diagnosis result of a centralized agent by consistent pairs of local diagnostics

The assumption made above is that the communication exchange between two agents is simultaneous (synchronous) and takes place in different communication rounds and that the local calculations of each component do not include new observations (events observed happening after θ_{com_q}). The consideration of asynchronous communication exchange brings nothing new but some more notation.

In the following section we present a distributed algorithm that comprises:

i) a procedure for performing the local preliminary calculations in absence of of any external information

ii) a procedure for information exchange

iii) a procedure for updating a local calculation to incorporate the received information

Then we prove the main result of this section that is the distributed algorithm we propose terminates after finitely many communication rounds and by the completion of the information exchange (communication protocol) the centralized diagnosis result is recovered.

4.3.1 The distributed algorithm

We start this section by emphasizing first the difficulties in designing a distributed algorithm under the setting that we consider and then the three steps $R1, R2.1$ and $R2.2$ aforementioned are presented in detail.

Discussion

Consider the PN displayed in Fig. 4.7. and then consider $\langle \mathcal{N}, M_0 \rangle$ decomposed in two sub-nets $\langle \mathcal{N}_1, M_{0_1} \rangle$ and $\langle \mathcal{N}_2, M_{0_2} \rangle$ as shown in Fig.4.8. $\langle \mathcal{N}_1, M_{0_1} \rangle$ and $\langle \mathcal{N}_2, M_{0_2} \rangle$ are bordered-nets where p_5 and p_9 are their common border places.

Let the local observation at site 1 and site 2 be: $\mathcal{O}_1^1 = t_6$ and $\mathcal{O}_1^2 = t_{10}$ respectively (where the upper index indicates the agent and the lower index indicates how many events were locally observed). The input and output transitions of the border places p_5 and p_9 are unobservable thus Ag_1 and Ag_2 should analyze PN models whose initial markings are uncertain, that is even though the agents know the initial local marking M_{0_1} and resp. M_{0_2} tokens from the neighboring component could have entered the local PN models.

Since a preliminary local calculation before communicating with the other agents is required (see $R1$ above) we are in trouble because the local agents should handle PN models with uncertain markings (due to the unobservable interactions with the neighboring components).

Consider the case of Ag_2. Via the border place p_5 tokens can enter $\mathcal{N}_2$ and then leave unobservably via p_9. The question we must answer is: *"what*

Ag_2 should do before communicating with the neighboring agent Ag_1 ?". One solution would be to consider upper bounds for the marking of the input places of each component (e.g. p_5 for $\mathcal{N}_2$ in Fig. 4.8), computing in this way an over-estimate of the local site behavior that is checked for consistency by communication. As discussed in Section 2.7 to calculate upper bounds it is necessary to calculate the overall plant behavior. In fact, what we want to avoid by a distributed analysis of the plant.

In the following we propose a distributed algorithm based on the concept of minimal explanations. We illustrate the method via the following example. Consider again the model displayed in Fig. 4.8 and analyse what Ag_1 does after observing the first execution the transition t_6 at the time θ_{t_6}. Ag_1 infers backwards that at least one token was present in p_4 before the time θ_{t_6}. Then, there are two possibilities for the required token to have reached p_4 namely: *"t_4 must have happened"* or *"t_3 must have happened"*. For the case *"t_3 must have happened"*, Ag_1 deduces that a token must have entered in $\mathcal{N}_1$ via p_9, and this token must have arrived at p_9 at a time before θ_{t_6}. Moreover a token in p_1 must have been also present for allowing t_3 to fire. Then Ag_1 deduces that a token in p_1 would have been present if t_0 would have been executed before θ_{t_6}. Thus we have that *if a token arrived in p_9 before the time θ_{t_6}*, then $\tau_{1_1} = t_0t_3t_6$ provides a minimal explanation of the observed event t_6.

This minimal explanation is represented in the backward configuration $\overleftarrow{C}_{1_1}$ in the backward unfolding $\overleftarrow{\mathcal{U}}_{\mathcal{N}_1}(t_6)$ (displayed in the left of Fig. 4.9) where:

$$\overleftarrow{C}_{1_1} = (\overleftarrow{B}_{\overleftarrow{C}_{1_1}}, \overleftarrow{E}_{\overleftarrow{C}_{1_1}}, \preceq),\ \overleftarrow{B}_{\overleftarrow{C}_{1_1}} = \{b_0, b_{9_1}, b'_1, b'_2, b_4, b_6\},\ \overleftarrow{E}_{\overleftarrow{C}_{1_1}} = \{e'_0, e_3, e_6\}$$

On the other hand for the case *"t_4 must have happened"*, Ag_1 derives backward three possibilities for explaining how a token could have be present in p_2 namely either t_0 or t_1 or t_2 fired before θ_{t_6}. Thus we have three more minimal explanations for the occurrence of t_6: $\tau_{2_1} = t_0t_4t_6$; $\tau_{3_1} = t_1t_4t_6$; $\tau_{4_1} = t_2t_4t_6$ that correspond with the following backward configurations in $\overleftarrow{\mathcal{U}}_{\mathcal{N}_1}(t_6)$:

- $\overleftarrow{C}_{2_1} = (\overleftarrow{B}_{\overleftarrow{C}_{2_1}}, \overleftarrow{E}_{\overleftarrow{C}_{2_1}}, \preceq)$, $\overleftarrow{B}_{\overleftarrow{C}_{2_1}} = \{b'_0, b''_1, b''_2, b_4, b_6\}$, $\overleftarrow{E}_{\overleftarrow{C}_{2_1}} = \{e''_0, e_4, e_6\}$

- $\overleftarrow{C}_{3_1} = (\overleftarrow{B}_{\overleftarrow{C}_{3_1}}, \overleftarrow{E}_{\overleftarrow{C}_{3_1}}, \preceq)$, $\overleftarrow{B}_{\overleftarrow{C}_{3_1}} = \{b''_0, b''_2, b_4, b_6\}$, $\overleftarrow{E}_{\overleftarrow{C}_{3_1}} = \{e_1, e_4, e_6\}$

- $\overleftarrow{C}_{4_1} = (\overleftarrow{B}_{\overleftarrow{C}_{4_1}}, \overleftarrow{E}_{\overleftarrow{C}_{4_1}}, \preceq)$, $\overleftarrow{B}_{\overleftarrow{C}_{4_1}} = \{b'''_0, b''_2, b_3, b_4, b_6\}$, $\overleftarrow{E}_{\overleftarrow{C}_{4_1}} = \{e_2, e_4, e_6\}$

Then consider the analysis of what Ag_2 does after observing the first execution of t_{10} at the time $\theta_{t_{10}}$. Ag_2 infers backward that at least one token was present in p_7 before the time θ_{10}. Then there are two possibilities for the required token to have reached p_7: *"t_9 must have happened"* or *"t_8 must have happened"*. For the case *"t_9 must have happened"* Ag_2 makes the assumption that a token must have arrived at p_5 at a time before $\theta_{t_{10}}$ while the case *"t_8 must have happened"* is explained by the token in p_8 that is present in the initial marking.

Thus Ag_2 derives two minimal explanations for the first occurrence of t_{10} namely:

1. if a token arrived in p_5 before $\theta_{t_{10}}$ then $\tau_{1_2} = t_9 t_{10}$ is a minimal explanation for t_{10}
2. $\tau_{2_2} = t_8 t_{10}$ is a minimal explanation of the first occurrence of t_{10}

that correspond with the two backward configurations in $\overleftarrow{\mathcal{U}}_{\mathcal{N}_2}(t_{10})$ (see Fig. 4.9-right).

- $\overleftarrow{C}_{1_2} = (\overleftarrow{B}_{\overleftarrow{C}_{1_2}}, \overleftarrow{E}_{\overleftarrow{C}_{1_2}}, \preceq)$, $\overleftarrow{B}_{\overleftarrow{C}_{1_2}} = \{b_{5_2}, b_{11}, b_7, b_{10}\}$, $\overleftarrow{E}_{\overleftarrow{C}_{1_2}} = \{e_9, e_{10}\}$
- $\overleftarrow{C}_{2_2} = (\overleftarrow{B}_{\overleftarrow{C}_{2_2}}, \overleftarrow{E}_{\overleftarrow{C}_{2_2}}, \preceq)$, $\overleftarrow{B}_{\overleftarrow{C}_{2_2}} = \{b_8, b_7, b_{10}\}$, $\overleftarrow{E}_{\overleftarrow{C}_{2_2}} = \{e_8, e_{10}\}$

Notice that the $\overleftarrow{\mathcal{U}}_{\mathcal{N}_1}(t_6)$ and $\overleftarrow{\mathcal{U}}_{\mathcal{N}_2}(t_{10})$ are obtained by running Algorithm 3 for component 1 and component 2 considering the marking of the input places $\mathcal{P}_{IN_i}$ respectively $\mathcal{P}_{IN_j}$ arbitrary large.

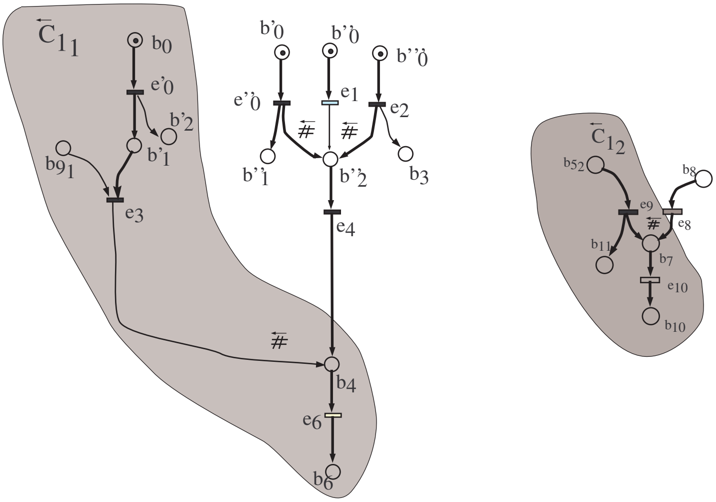


Figure 4.9:

Consider the case of $\overleftarrow{C}_{1_1}$ for component 1. By appending the token from the local minimal marking M_{0_1} that was not used for explaining the local observation (e.g. bb_0 that corresponds to the second token in p_0) we obtain $\underline{C}_{1_1}$ that is a minimal configuration in the forward unfolding $\mathcal{U}_{\mathcal{N}_1}(t_6)$ that would have been derived for $\mathcal{N}_1$ considering as initial marking $M_{0_1} \cup \{(p_9, 1)\}$. Then $\underline{C}_{1_1}$ is extended by appending only unobservable strings that can be executed from $CUT(\underline{C}_{1_1})$ (see Fig. 4.10-left). In this way we calculate the set of local configurations $\mathcal{C}_{1_1}(t_6)$ that are unobservable extensions of $\underline{C}_{1_1}$.

Consider Ag_2 making the same calculations e.g. for $\overleftarrow{C}_{1_2}$ it derives the local minimal configuration $\underline{C}_{1_2}$ by adding to $\overleftarrow{C}_{1_2}$ the condition b_8 and then calculates forward the set of configurations $\mathcal{C}_{1_2}(t_{10})$ that are unobservable extensions of $\underline{C}_{1_2}$ (see Fig. 4.10-right).

Notice that the extensions of a minimal local configuration are calculated whenever the communication exchange is about to start. This is because Ag_1, resp. Ag_2 should exchange information about the tokens that are required and the tokens that could have been provided. Consequently we refer to $\mathcal{C}_1(t_6)$ and $\mathcal{C}_2(t_{10})$ as the set of preliminary local configurations of component i and component j respectively.

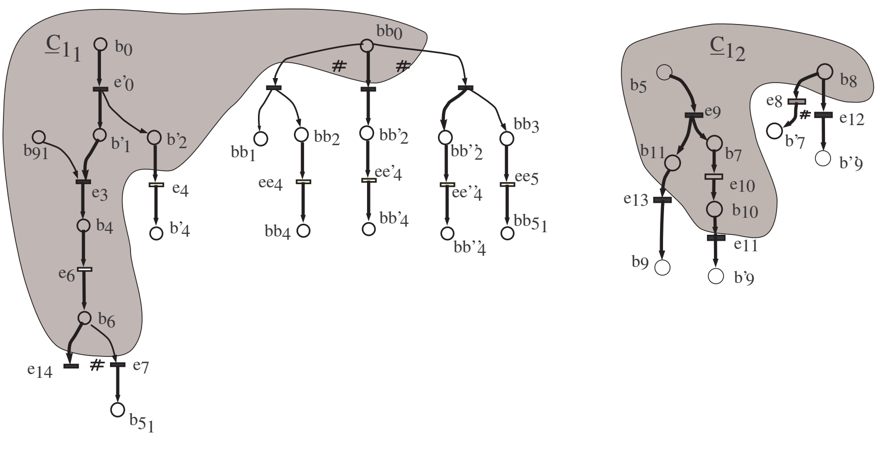


Figure 4.10:

Consider for component 1 the preliminary local configuration $C_{1_1} \in \mathcal{C}_1(t_6)$ (Fig. 4.11-left). For C_{1_1}, Ag_1 derives that a token in p_9 should have been present at a time before θ_{t_6} (see b_{9_1} in Fig. 4.9-left) and if this is true, then two tokens can be provided to $\mathcal{N}_2$ via the place p_5. One token could have been provided after the time θ_{t_6} (see b_{5_1}) while the second token could have been provided at any time after the process starts (see bb_{5_1}).

Similarly consider for site 2 the preliminary local configuration $C_{1_2} \in \mathcal{C}_2(t_{10})$ (Fig. 4.11-right). For C_{1_2}, Ag_2 derives that a token in p_5 should have been present at a time before $\theta_{t_{10}}$ (see b_{5_2} in Fig. 4.9-right) and if this is true, then three tokens can be provided to $\mathcal{N}_1$ via the place p_9. One token could have been provided after the time $\theta_{t_{10}}$ (see b'_{9_2}), the second token (b_{9_1}) could have been provided after the time the required token in p_5 has entered $\mathcal{N}_2$ (b_{9_2}) while the third token could have been provided to $\mathcal{N}_1$ at any time after the process starts (see b''_{9_2}).

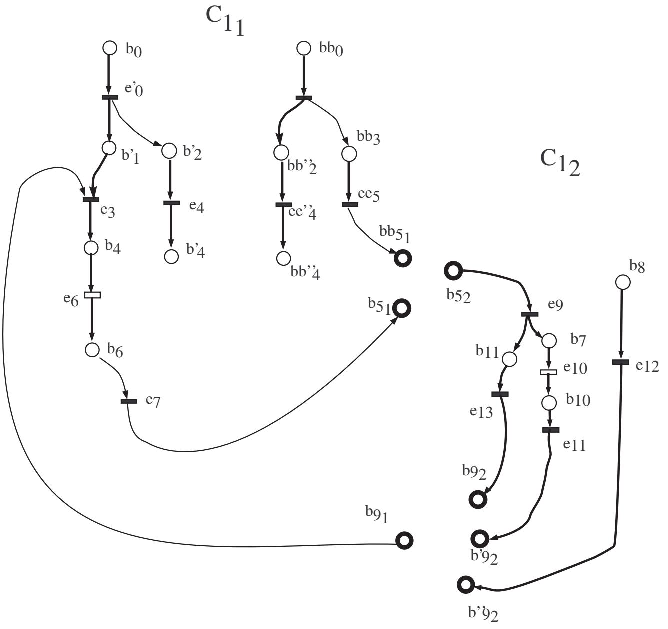


Figure 4.11:

The main goal of any distributed algorithm is to recover the overall plant behaviour. It means that the local agent should check the consistency of its preliminary results (local configurations) for deriving consistent global configurations.

Since a local agent knows only the model of its component and its interactions with the neighbouring components, it means that the consistency of the local configurations is achieved by the local agent exchanging information about the common border-places.

As presented in Section 2.7 to check the consistency of two local configurations C_{ν_i} and C_{ν_j} requires that Ag_i and Ag_j exchange not only information regarding the number of input and output border conditions considered in C_{ν_i} and C_{ν_j} respectively but also information about how the input and output border conditions are causally related in C_{ν_i} and C_{ν_j} respectively.

For instance to check the consistency of the local configurations C_{1_1} and C_{1_2} that are displayed in Fig. 4.11 requires that Ag_1 sends to Ag_2 the message that:

- at least a token is required to arrive at p_9 before the time θ_{t_6} when t_6 was executed (the input border-condition b_{9_1})
- and at least two tokens can arrive at p_5 namely:
    - one token can arrive in p_5 after the time θ_6 (the output border conditions b_{5_1})
    - and one token can arrive in p_5 after the process starts (the output border condition bb_{5_1})

This message can be summarized as:

$$\{b_{9_1} \preceq e_6\} \wedge \{e_6 \preceq b_{5_1}\} \wedge \{bb_{5_1}\}$$

On the other hand Ag_2 sends to Ag_1 the message that:

- at least a token is required to arrive at p_5 before the time $\theta_{t_{10}}$ when t_{10} was executed (the input border condition b_{5_2})
- and three tokens can arrive at p_9 namely:
    - one token can arrive in p_9 after the time the required token in p_5 has entered in $\mathcal{N}_2$ (the input border condition b_{9_2})
    - one token can arrive in p_9 after time $\theta_{t_{10}}$ (the input border condition b_{9_2})
    - and one token can arrive in p_9 after the process starts (the input border condition b''_{9_2})

This message can be summarized as:

$$\{b_{5_2} \preceq e_{10}\} \wedge \{b_{9_2} \preceq b_{5_2}\} \wedge \left\{b'_{9_2} \preceq e_{10}\right\} \wedge \left\{b''_{9_2}\right\}$$

Based on the received information the local agents check whether there is an interpretation function $\psi_{\ell_{1_1 1_2}} \in \Psi_{1_1 1_2}$ ($\ell_{1_1 1_2} \in \mathcal{V}_{1_1 1_2}$) of the border conditions (see Section 2.7) so that $(C_{1_1}, C_{1_2}, \psi_{\ell_{1_1 1_2}})$ is a global configuration. $C_{\ell_{1_1 1_2}} = (C_{1_1}, C_{1_2}, \psi_{\ell_{1_1 1_2}})$ is a global configuration if the PN obtained by from C_{1_1} and C_{1_2} by merging their border-places is acyclic and there are no input border-conditions in C_{1_1} and C_{1_2} that have no input events in $C_{\ell_{1_1 1_2}}$.

If not all the input border conditions of C_{1_1} and C_{1_2} have input events in $C_{\ell_{1_1 1_2}}$ the local agents should check if $C_{\ell_{1_1 1_2}}$ can be further extended by using the *"new"* tokens that become *"available"* after the information exchange. For a local agent i the *"new"* tokens in $C_{\ell_{1_1 1_2}}$ are the output border conditions of C_{1_2} that were not assigned to input border-conditions of C_{1_1}.

By further (unobservable) extensions new pairs of local configurations in two different components may become consistent. Since the overall plant model $\mathcal{N}$ is bounded with respect to the unobservable evolution (see item $viii$) in setting) the consistency check can be proven to terminate after a finite number of communication rounds between the local agents.

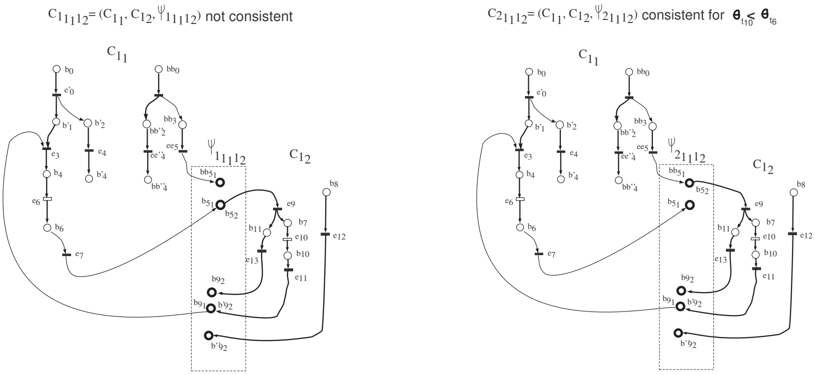


Figure 4.12:

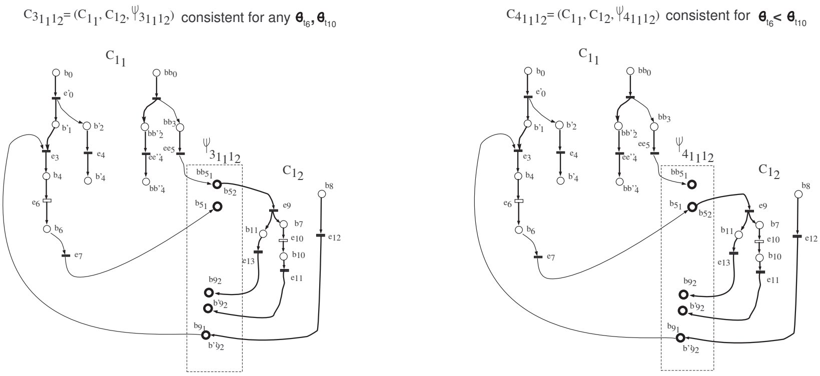


Figure 4.13:

Example 22. *Consider again the two local configurations C_{1_1} and C_{1_2} displayed in Fig. 4.11. Then consider the following interpretation functions:*

$\psi_{1_{1_1 1_2}}(b_{9_1}) = b'_{9_2}$, $\psi_{1_{1_1 1_2}}(b_{5_2}) = b_{5_1}$: $C_{1_{1_1 1_2}}$ *is not consistent and is dis-*

carded since $C_{1_1 1_2}$ is not acyclic; see the circuit (Fig. 4.12-left):

$$\zeta = b_4 e_6 b_6 e_7 b_{5_1} e_9 b_7 e_{10} b_{10} e_{11} b_{9_1} e_3$$

$\psi_{2_{1_1 1_2}}(b_{9_1}) = b'_{9_2}$, $\psi_{2_{1_1 1_2}}(b_{5_2}) = bb_{5_1}$: *$C_{2_{1_1 1_2}}$ is consistent only if t_{10} fires at a time before t_6 ($\theta_{t_{10}} < \theta_{t_6}$) since e_{10} is a predecessor of e_6 (Fig. 4.12-right).*

$\psi_{3_{1_1 1_2}}(b_{9_1}) = b''_{9_2}$, $\psi_{3_{1_1 1_2}}(b_{5_2}) = bb_{5_1}$: *$C_{3_{1_1 1_2}}$ is consistent for any order in which t_{10} and t_6 are executed since e_{10} and e_6 are concurrent in $C_{3_{1_1 1_2}}$ (notice that the order in which t_{10} and t_6 fired is known since a the time an observable transition fires is measured with accuracy according with a global clock) (Fig. 4.13-left).*

$\psi_{4_{1_1 1_2}}(b_{9_1}) = b''_{9_2}$, $\psi_{4_{1_1 1_2}}(b_{5_2}) = b_{5_1}$: *$C_{4_{1_1 1_2}}$ is consistent only if t_6 fires at a time before t_{10} ($\theta_{t_6} < \theta_{t_{10}}$) since e_6 is a predecessor of e_{10} (Fig. 4.13-right).*

$\psi_{5_{1_1 1_2}}(b_{9_1}) = \varepsilon$, $\psi_{5_{1_1 1_2}}(b_{5_2}) = \varepsilon$: *$C_{5_{1_1 1_2}}$ is not consistent since the input border-conditions are not assigned to output-border conditions. $C_{5_{1_1 1_2}}$ is extendable since b_{5_1}, bb_{5_1} are new tokens that can be used to unobservably extend $C_{5_{1_1 1_2}}$ in component 2 and b_{9_2}, b'_{9_2} and b''_{9_2} are new tokens that can be used to unobservably extend $C_{5_{1_1 1_2}}$ in component 1. However further unobservable extensions of $C_{5_{1_1 1_2}}$ in component 1 and 2 do not lead to global configurations but this is not in general true. The reason why further extensions of $C_{5_{1_1 1_2}}$ are not global configurations is that it is not possible to fire unobservable transitions (append unobservable events) s.t. an output border condition is produced at the border place p_5 for explaining the input border condition b_{5_2}.*

The approach we follow for designing the distributed algorithm can be outlined as follows:

i) the local preliminary computation comprises two phases: $i.1$) first a backward calculation is performed for deriving the set of minimal configurations that provide the minimal explanations of the local observation based on the assumption that the minimal number of tokens have entered the local site; $i.2$) when the communication is allowed the minimal configurations are extended (forward) for finding the tokens that could have exited from the local site.

ii) then by communicating information with its neighbors Ag_i checks the consistency of its local results and also generates new local traces that are checked for consistency in a new communication round.

iii) when a fix-point is achieved the consistent set of local results recover the centralized diagnosis result.

4.3.2 Formal specification and proofs of correctness

Procedure for performing local preliminary calculations ($R.1$)

In this subsection we present formally the preliminary calculations performed by a local agent Ag_i ($i \in J$) prior to the time the agents are allowed to communicate the first time θ_{com_1} and between two communication sessions (e.g. between θ_{com_q} and $\theta_{com_{q+1}}$).

Consider in the following the case of the agent Ag_i having received by the time θ_{com}, when the first communication session is allowed, the local observation $\mathcal{O}^i_{\theta_{com}} = t^o_{1_i} \dots t^o_{n_i}$. As assumed before, there are no delays in receiving the plant observation and the execution of an observable transition is correctly reported. This implies that all the observable events that are executed in component $i \in J$ before the time θ_{com} are included in the local observation while no event observed executed after the time θ_{com} when the communication session is initiated is included in a local preliminary calculation.

Since there is no *a priori* knowledge of the marking of the input places $\mathcal{P}_{IN_i}$, Ag_i cannot make any assumption on the number of tokens that could have entered $\mathcal{N}_i$. To deal with the unknown marking of the input places $\mathcal{P}_{IN_i}$, Ag_i performs a backward search for finding the minimal configurations (minimal explanations) of $\mathcal{O}^i_{\theta_{com}}$. Whenever the backward search considers a token in an input place $p \in \mathcal{P}_{IN_i}$ the backward search stops with the assumption that a token must have entered $\mathcal{N}_i$ via p.

As presented in Section 3.2 a local agent Ag_i can construct the backward unfolding $\overleftarrow{\mathcal{U}}_{\mathcal{N}_i}(\mathcal{O}^i_{\theta_{com}})$ by running the algorithm **B_Unfold(**$\mathcal{O}_{\theta_{com}}$). However some technical conditions are required to be fulfilled in order to prevent the local agents to assume an infinite number of tokens that are required to minimally explain the local observation.

Given an unobservable elementary circuit ζ_{uo}, denote by Υ_ζ the set of limiting places of ζ_{uo}: $\Upsilon_{\zeta_{uo}} \triangleq \{p : (p \notin \zeta_{uo}) \wedge (\exists t \in \zeta_{uo} \text{ s.t. } p \in {}^\bullet t)\}$. A place $p \in \Upsilon_{\zeta_{uo}}$ is a limiting places of ζ_{uo} since every complete execution of ζ_{uo} consumes tokens from p. Denote $M_{\Upsilon_{\zeta_{uo}}}$ the minimal marking of $\Upsilon_{\zeta_{uo}}$ that allows for a complete execution of ζ_{uo}.

Assumption 4. *For any local model $\mathcal{N}_i$ and for any* unobservable elementary circuit ζ_{uo_i}, *there does not exist an executable sequence of unobservable transitions σ_{uo_i} with initial marking M that has tokens only in the input places $\mathcal{P}_{IN_i}$ ($M(p) = 0$ for $p \notin \mathcal{P}_{IN_i}$) s.t. by firing from M, σ_{uo_i} produces a marking M' greater than the limiting marking of ζ_{uo_i}, $M_{\Upsilon_{\zeta_{uo_i}}}$. $\not\exists \sigma_{uo} \in \mathcal{T}^*_{uo_i}$ s.t. ($M_{\Upsilon_{\zeta_{uo_i}}} \overset{\sigma_{uo_i}}{\rightsquigarrow} M$) $\wedge$ ($M(p) \neq 0 \Rightarrow p \in \mathcal{P}_{IN_i}$).*

Proposition 13. *Given a PN model $\mathcal{N}$ s.t. $\forall i \in J$ Assumption 4 holds true for $\mathcal{N}_i$, then, for any observation $\mathcal{O}^i_{\theta_{com}}$ generated by component i, by running*

***B_Unfold(*$\mathcal{O}_{\theta_{com}}$*)** (Algorithm 3) we obtain for every backward configuration* $\forall \overleftarrow{C}_{\overleftarrow{\nu}_i}(\mathcal{O}^i_{\theta_{com}}) \in \overleftarrow{\mathcal{C}}_i(\mathcal{O}^i_{\theta_{com}})$ *a finite set of input border-conditions, that is* $\mid \overleftarrow{B}_{\overleftarrow{C}_{\overleftarrow{\nu}_i}}(IN_i) \mid < +\infty$.

Proof. The proof is as follows. Consider that $\mathcal{N}_i$ is acyclic. For this case the proof that running **B_Unfold(**$\mathcal{O}_{\theta_{com}}$) we obtain a finite set of input border-conditions $\overleftarrow{B}_{\overleftarrow{C}_{\overleftarrow{\nu}_i}}(IN_i)$ is trivial.

Then consider that $\mathcal{N}_i$ contains only one circuit ζ_{uo_i}. If all the transitions in the circuit are backfired and the marking before the execution of ζ_{uo_i} e.g. $M1_i$ is smaller then the marking after the execution of ζ_{uo_i} (e.g. $M2_i$) then $M2_i$ is *stopped* until $M1_i$ is found that is part of a minimal local explanation. If $M1_i$ is found as part of minimal local explanation then the transitions of the unobservable circuit ζ_{uo_i} can be backfired from $M2$.

We have by assumption $viii)$ in the setting that $\mathcal{N}$ is structurally bounded w.r.t. the unobservable evolution i.e. $\forall M \in \mathbb{N}^{|\mathcal{P}|} \wedge \forall \sigma_{uo} \in \mathcal{T}^*_{uo} : M \xrightarrow{\sigma_{uo}} M'$ and $M' \geq M \Rightarrow M' = M$ and by Assumption 4 we have that any unobservable circuit in component i cannot be executed only with tokens coming from the input places $\mathcal{P}_{IN_i}$.

Hence for any (finite) local marking M_i, $\langle \mathcal{N}_i, M_i \rangle$ is bounded w.r.t. the unobservable evolution, i.e.:

$$\forall \sigma_{uo_i} \in \mathcal{T}^*_{uo_i} : M_i \xrightarrow{\sigma_{uo}} M'_i \text{ and } M'_i \geq M_i \Rightarrow M'_i = M_i$$

Thus ζ_{uo_i} can be executed finitely many times. This proves the statement since after firing the last time a transition in the circuit then only transitions in an acyclic sub-net of $\mathcal{N}_i$ will be considered for backfiring.

If $\mathcal{N}_i$ contains an arbitrary number of circuits then the proof is similar since the condition checked in the algorithm is if the marking repeats or increases along a trace that is derived backwards. □

From the set of backward configurations $\overleftarrow{\mathcal{C}}_i(\mathcal{O}^i_{\theta_{com}})$ that explain the local observation, Ag_i derives the set of minimal (preliminary) configurations $\underline{\mathcal{C}}_i(\mathcal{O}^i_{\theta_{com}})$.

Given a configuration $\underline{C}_{\underline{\nu}_i} \in \underline{\mathcal{C}}_i(\mathcal{O}^i_{\theta_{com}})$ denote $\underline{B}_{\underline{C}_{\underline{\nu}_i}}(IN_i)$ the set of conditions that correspond to the input places $p_i \in \mathcal{P}_{IN_i}$:

$$\underline{B}_{\underline{C}_{\underline{\nu}_i}}(IN_i) = \Big\{ b_i \mid b_i \in \underline{B}_{\underline{C}_{\underline{\nu}_i}} \wedge \phi(b_i) = p_i \wedge p_i \in \mathcal{P}_{IN_i} \Big\}$$

Remark 9. *Notice that if there is a priori knowledge about the maximum number of tokens that can enter a component in between two communication sessions this information can be used to discard a local configuration that considers a marking of the input border places that is not smaller than the known upper bound or equal.*

Consider that at the time θ_{com} when the communication with the neighboring agents is the first time allowed Ag_i has computed the set of minimal configurations $\underline{\mathcal{C}}_i(\mathcal{O}^i_{\theta_{com}})$ for the observation received up to the time $\mathcal{O}^i_{\theta_{com}}$.

Then Ag_i must also calculate the estimate of the number of tokens that could have exited $\mathcal{N}_i$ via the output places $\mathcal{P}_{OUT_i}$.

For doing this Ag_i calculates for every minimal configuration $\underline{C}_{\nu_i} \in \underline{\mathcal{C}}_i(\mathcal{O}^i_{\theta_{com}})$ the forward extensions by considering all the unobservable extensions (sequences of unobservable transitions) that can be appended.

Denote by $\mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ the set of all the unobservable extensions of the minimal configurations:

$$\mathcal{C}_i(\mathcal{O}^i_{\theta_{com}}) = \Big\{C_{\nu_i} \mid C_{\nu_i} = \underline{C}_{\nu_i} \odot e_{1_i} \ldots \odot e_{m_i} \wedge \\ \wedge\ \phi(e_{q_i}) \in \mathcal{T}_{uo_i}, q_i = 1_i, \ldots, m_i \wedge \underline{C}_{\nu_i} \in \underline{\mathcal{C}}_i(\mathcal{O}^i_{\theta_{com}})\Big\}$$

Denote by $\mathcal{E}_i(\mathcal{O}^i_{\theta_{com}})$ the set of preliminary explanations of the local observation $\mathcal{O}^i_{\theta_{com}}$:

$$\mathcal{E}_i(\mathcal{O}^i_{\theta_{com}}) = \{\sigma \mid \sigma \in \langle E_{C_{\nu_i}}\rangle \wedge C_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})\}$$

Let $\mathcal{L}_{\mathcal{N}_i}(\mathcal{O}^i_{\theta_{com}})$ be the set of preliminary local traces in $\mathcal{N}_i$ that correspond to $\mathcal{E}_i(\mathcal{O}^i_{\theta_{com}})$:

$$\mathcal{L}_{\mathcal{N}_i}(\mathcal{O}^i_{\theta_{com}}) = \{\tau_i \mid \tau_i = \phi(\sigma) \wedge \sigma \in \mathcal{E}_i(\mathcal{O}^i_{\theta_{com}})\} \tag{4.9}$$

The local preliminary diagnosis $\mathcal{LPD}_i(\mathcal{O}^i_{\theta_{com}})$ is:

$$\mathcal{LPD}_i(\mathcal{O}^i_{\theta_{com}}) = \{\tau_{f_i} \mid \tau_{f_i} = \Pi_{\mathcal{T}_{F_i}}(\tau_i) \wedge \tau_i \in \mathcal{L}_{\mathcal{N}_i}(\mathcal{O}^i_{\theta_{com}})\} \tag{4.10}$$

In the following we compare the local preliminary diagnosis $\mathcal{LPD}_i(\mathcal{O}^i_{\theta_{com}})$ with the diagnosis result for component i that would have been derived by a centralized agent (knowing the overall plant model and having the overall plant observation).

Denote by $\mathcal{D}_i(\mathcal{O}_{com})$ the centralized diagnosis for site i:

$$\mathcal{D}_i(\mathcal{O}_{com}) = \Big\{\sigma_{f_i} \mid \sigma_{f_i} = \Pi_{\mathcal{T}_{f_i}}(\sigma_f) \wedge \sigma_f \in \mathcal{D}_{\mathcal{N}}(\mathcal{O}_{com})\Big\} \tag{4.11}$$

where $\mathcal{D}_{\mathcal{N}}(\mathcal{O}_{com})$ is the centralized diagnosis (see Equation 4.3).

Proposition 14. *If the PN model is such that $\forall i \in J$ any oriented path $\wp_i$ in $\mathcal{N}_i$ that starts in an input place p ($p \in \mathcal{P}_{IN_i}$) and ends in an output place p' ($p' \in \mathcal{P}_{OUT_i}$) contains at least one observable transition then the local preliminary diagnosis $\mathcal{LPD}_i(\mathcal{O}^i_{\theta_{com}})$ is an over-diagnosis of the local site diagnosis $\mathcal{D}_i(\mathcal{O}_{\theta_{com}})$ w.r.t. the detection of the faults that for sure have happened, i.e.:*

$$(\forall \sigma_f \in \mathcal{D}_i(\mathcal{O}_{com}),\ \ t_f \in \Sigma(\sigma_f)) \Rightarrow (\exists \sigma'_f \in \mathcal{LPD}_i(\mathcal{O}^i_{\theta_{com}})\ s.t.\ t_f \in \Sigma(\sigma'_f))$$

Proof. Consider a local preliminary configuration $C_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ that was derived by Ag_i before communication with Ag_j. For the local configuration C_{ν_i} Ag_i considers the set of input-border conditions $B_{C_{\nu_i}}(IN_i)$ that corresponds with the minimum number of tokens that is required to enter in component i for explaining the local observation $\mathcal{O}^i_{\theta_{com}}$.

By the assumption that any oriented path $\wp_i$ in $\mathcal{N}_i$ that starts in an input place p ($p \in \mathcal{P}_{IN_i}$) and ends in an output place p' ($p' \in \mathcal{P}_{OUT_i}$) contains at least one observable transition we have that no more output border conditions can be generated with the extra input border conditions.

This means that if there does not exist a preliminary local configuration $C_{\nu_j} \in \mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$ derived by Ag_j for component j s.t. the input border conditions $B_{C_{\nu_i}}(IN_i)$ are satisfied by output border conditions $B_{C_{\nu_j}}(OUT_j)$, then C_{ν_j} is unfeasible.

If the number of tokens required can be provided then the local traces in i have been already derived.

If there can be provided more tokens than required then the local traces in i that have been already derived are still possible by the monotonicity (see Lemma 2).

Moreover with the extra-tokens that could have entered there is possible to extend C_{ν_j} by appending some unobservable events. We have that all the unobservable events that are newly appended to extend C_{ν_j} are concurrent with the event-nodes that correspond to the observation that means that these events could but non-necessarily must have happened. By Assumption 2 we have that the diagnosis result w.r.t. the detection of the faults that for sure happened is not modified.

Then consider that the tokens that correspond to $B_{C_{\nu_i}}(IN_i)$ cannot be provided by any preliminary local configuration $C_{\nu_j} \in \mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$ derived by Ag_j for component j. It means that the consideration of an unfeasible local configuration C_{ν_i} leads to an over-diagnosis.

The statement is proved straightforward considering all the preliminary local configurations. □

Remark 10. *If the assumption that along any oriented path $\wp_i$ in $\mathcal{N}_i$ that starts in an input place p ($p \in \mathcal{P}_{IN_i}$) and ends in an output place p' ($p' \in \mathcal{P}_{OUT_i}$) is at least an observable event is not satisfied for a local component $\mathcal{N}_i$ the local preliminary diagnosis of the local component $\mathcal{LPD}_i(\mathcal{O}^i_{\theta_{com}})$ is not in general an over-diagnosis of the local site diagnosis $\mathcal{D}_i(\mathcal{O}_{\theta_{com}})$ w.r.t. the detection of the faults that for sure have happened. This is because fault-transitions that are situated along unobservable paths that start in an input place ($p \in \mathcal{P}_{IN_i}$) and ends in an output place p' ($p' \in \mathcal{P}_{OUT_i}$) may not be detected by the local agent in its preliminary diagnosis.*

This result justifies why one carries out a calculation after each observation. If one did not have this property of over-diagnosis one could - given an infinite fast processor - carry out all calculations at θ_{com}. Moreover this result is also important for distributed supervisory architectures with unreliable communication channels since it proves that a local agent in absence of any external information can provide an over-diagnosis of local faults. Communication between agents serves to refine the diagnosis result.

Algorithm 6 Preliminary_Local_Calculation (Ag_i considered)

```
Require: $\mathcal{O}^i_{\theta_{com}}$; $\theta_{com}$;
Ensure: $\mathcal{LPD}(\mathcal{O}^i_{\theta_{com}})$
calculate $\overleftarrow{\mathcal{C}}_i(\mathcal{O}^i_{\theta_{com}})$
calculate $\underline{\mathcal{C}}_i(\mathcal{O}^i_{\theta_{com}})$
calculate $\mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$
calculate $\mathcal{E}_i(\mathcal{O}^i_{\theta_{com}})$
calculate $\mathcal{L}_{\mathcal{N}_i}(\mathcal{O}^i_{\theta_{com}})$
$\mathcal{LPD}(\mathcal{O}^i_{\theta_{com}}) = \Pi_{\mathcal{T}_f}(\mathcal{L}_{\mathcal{N}_i}(\mathcal{O}^i_{\theta_{com}}))$
```

Procedure for information exchange at θ_{com}

In what follows we restrict the presentation considering a preliminary configuration $C_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ where $\nu_i \in \mathcal{V}_i$ is the set of indexes of the preliminary local configurations derived by Ag_i for the local component i.

Thus $C_{\nu_i} = (B_{C_{\nu_i}}, E_{C_{\nu_i}}, \preceq_i)$ requires that the border conditions represented by $B_{C_{\nu_i}}(IN_i)$ to be satisfied. Notice that we have $\forall b_{IN_i} \in B_{C_{\nu_i}}(IN_i)$ implies that $\exists e^o_{q_i} \in E_{C_{\nu_i}}$ s.t. $\phi(e^o_{q_i}) = t^o_{q_i} \wedge b_{IN_i} \preceq e^o_{q_i}$.

For $b_{IN_i} \in B_{C_{\nu_i}}(IN_i)$ denote by $E^o_{C_{\nu_i}}(b_{IN_i})$ the set of event-nodes in C_{ν_i} that correspond to observed events that have b_{IN_i} as a predecessor:

$$E^o_{C_{\nu_i}}(b_{IN_i}) = \left\{ e^o_{q_i} \in E_{C_{\nu_i}} \mid b_{IN_i} \prec e^o_{q_i} \wedge \phi(e^o_{q_i}) \in \mathcal{T}_{o_i} \right\}$$

The latest time b_{IN_i} *"must have been satisfied"* (a token must have entered $\mathcal{N}_i$ via $p_i = \phi(b_{IN_i})$) is:

$$\theta_{b_{IN_i}} \leq \texttt{min}_{e^o_{q_i}}(\theta_{t^o_{q_i}}) \quad e_{q_i} \in E^o_{C_{\nu_i}}(b_{IN_i}) \;\wedge\; \phi(e^o_{q_i}) = t^o_{q_i} \tag{4.12}$$

To each border condition b_{IN_i} we can associate a temporal constraint of the form given by (4.12). For simplifying the notation use $b^\theta_{IN_i}$ as shorthand for $b_{IN_i} \in B_{C_{\nu_i}}(IN_i) : \theta_{b_{IN_i}} \leq \texttt{min}_{e^o_{q_i}} \theta_{t^o_{q_i}}$:

$$b^\theta_{IN_i} = \left\{ b_{IN_i} \in B_{IN_i} \wedge \theta_{b_{IN_i}} \leq \texttt{min}_{e^o_{q_i}} \theta_{t^o_{q_i}} \right\}$$

When omitted, the left resp. the right timing constraints are $0 \leq \theta$ and $\theta < \infty$ respectively.

Hence the minimal requirement of $C_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ can be expressed as a conjunction of temporal conditions on the border places:

$$B^{\theta}_{C_{\nu_i}}(IN_i) = \bigwedge_{b_{IN_i} \in B_{C_{\nu_i}}(IN_i)} b^{\theta}_{IN_i} \tag{4.13}$$

The timing constraints for the output conditions B_{OUT_i} are derived as follows. Denote by $E^o_{C_{\nu_i}}(b_{OUT_i})$ the set of event-nodes in C_{ν_i} that are predecessors of the output condition node b_{OUT_i}:

$$E^o_{C_{\nu_i}}(b_{OUT_i}) = \left\{e^o_{q_i} \in E_{C_{\nu_i}} \mid e^o_{q_i} \preceq b_{OUT_i} \wedge \phi(e^o_{q_i}) \in \mathcal{T}_{o_i}\right\}$$

Denote by $B_{IN_i}(b_{OUT_i})$ the set of input border conditions that are predecessors of the output border condition b_{OUT_i}:

$$B_{IN_i}(b_{OUT_i}) = \left\{b_{IN_i} \in B_{C_{\nu_i}}(IN_i) \mid b_{IN_i} \preceq b_{OUT_i}\right\}$$

We have that:

$$\theta_{b_{OUT_i}} \geq \mathtt{max}(\mathtt{max}_{e^o_{q_i}}(\theta_{e^o_{q_i}}), \mathtt{max}_{b_{IN_i}}(\theta_{b_{IN_i}})) \tag{4.14}$$

where $e^o_{q_i} \in E^o_{C_{\nu_i}}(b_{OUT_i})$ and $b_{IN_i} \in B_{IN_i}(b_{OUT_i})$.

Notice that if $E^o_{C_{\nu_i}}(b_{OUT_i}) = \emptyset$ and $B_{IN_i}(b_{OUT_i}) = \emptyset$ then $\theta_{b_{OUT_i}} \geq 0$.

Similarly denote by $b^{\theta}_{OUT_i}$ the timing constraint given by (4.14), where $\theta^{\theta}_{b_{OUT_i}}$ is the global time a token *could have exited* $\mathcal{N}_i$ given that $B^{\theta}_{C_{\nu_i}}(IN_i)$ is satisfied.

For the output border-conditions that could have been satisfied we derive a conjunction of temporal constraints having the form:

$$B^{\theta}_{C_{\nu_i}}(OUT_i) = \bigwedge_{b_{OUT_i} \in B_{C_{\nu_i}}(OUT_i)} b^{\theta}_{OUT_i} \tag{4.15}$$

Since a local agent does not know the models of the neighboring agents but only the set of input and output places $(\mathcal{P}_{IN_i}, \mathcal{P}_{OUT_i})$ the message that is sent by Ag_i to its neighbor Ag_j regarding its local configuration $C_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ comprises only information about their common border-places

$$\mathcal{MSG}_{i \to j} = \left\{(B^{\theta}_{C_{\nu_i}}(IN_i), B^{\theta}_{C_{\nu_i}}(OUT_i)) \mid C_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})\right\} \tag{4.16}$$

Algorithm 7 Communication_exchange (Ag_i considered)

```
Require: $\mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$
Ensure: $\mathcal{MSG}_{i \to j}$
for all $C_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ do
    calculate $B^{\theta}_{C_{\nu_i}}(IN_i)$ {see Eq. 4.13}
    calculate $B^{\theta}_{C_{\nu_i}}(OUT_i)$ {see Eq. 4.14}
end for
$\mathcal{MSG}_{i \to j}$ {see Eq. 4.16}
```

Procedure for updating the local calculation in agent Ag_i after receiving message from agent Ag_j

This subsection describes the calculation performed by Ag_i after it has calculated the set of preliminary local configurations $\mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ and after it has received the message $\mathcal{MSG}_{j \to i}$ sent by Ag_j based on a similar preliminary computation of component j.

To simplify the presentation consider the case of only two agents Ag_i and Ag_j and then consider:

1. an arbitrary local configuration $C_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$
2. an arbitrary local configuration $C_{\nu_j} \in \mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$
3. and the information $\mathcal{MSG}_{j \to i}$ sent by Ag_j and received by Ag_i regarding the local configuration C_{ν_j}: $(B_{C_{\nu_j}}(IN_j), B_{C_{\nu_j}}(OUT_j))$

We have that C_{ν_i} requires $B^{\theta}_{C_{\nu_i}}(IN_i)$ to be satisfied providing then $B^{\theta}_{C_{\nu_i}}(OUT_i)$ while $B^{\theta}_{C_{\nu_j}}(OUT_j)$ is the set of border-conditions in C_{ν_j} that may be satisfied whenever $B^{\theta}_{C_{\nu_j}}(IN_j)$ is satisfied.

Definition 48. *Given two local configurations $C_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ and $C_{\nu_j} \in \mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$ denote by $\Psi_{\nu_i\nu_j}$ the collection of interpretation functions of the border conditions:*

$$\Psi_{\nu_i\nu_j} = \left\{ \psi_{\ell_{\nu_i\nu_j}} \mid \ell_{\nu_i\nu_j} \in \mathcal{V}_{\nu_i\nu_j} \right\}$$

where an interpretation function $\psi_{\ell_{\nu_i\nu_j}} \in \Psi_{\nu_i\nu_j}$ associates to each input-border condition (e.g. $b_{IN_i} \in B_{C_{\nu_i}}(IN_i)$) either:

- *an output border condition ($b_{OUT_j} \in B_{C_{\nu_j}}(OUT_j)$) s.t.:*

  b_{IN_i} and b_{OUT_j} correspond to the same border place
  and $\theta_{b_{OUT_j}} \leq \theta_{b_{IN_i}}$

- *or the symbol ε that means that b_{IN_i} remains not interpreted (not matched)*

Moreover if $\psi_{\ell_{\nu_i\nu_j}}(b_{IN_i}) = b_{OUT_j} \in B_{C_{\nu_j}}(OUT_j)$ *then* $\forall b'_{IN_i} \in B_{C_{\nu_i}}(IN_i)$*,* $\psi_{\ell_{\nu_i\nu_j}}(b_{IN_i}) = \psi_{\ell_{\nu_i\nu_j}}(b'_{IN_i}) \Rightarrow b_{IN_i} = b'_{IN_i}$*, i.e.* $\psi_{\ell_{\nu_i\nu_j}}$ *is injective in* $B_{C_{\nu_j}}(OUT_j)$ *and similarly* $\psi_{\ell_{\nu_i\nu_j}}$ *is injective in* $B_{C_{\nu_i}}(OUT_i)$*.*

Given two local configurations $C_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ and $C_{\nu_j} \in \mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$ and an interpretation function of their border conditions $\psi_{\ell_{\nu_i\nu_j}} \in \Psi_{\nu_i\nu_j}$ denote by $C_{\ell_{\nu_i\nu_j}}$ the configuration obtained by merging the border-conditions of C_{ν_i} and C_{ν_j} according with $\psi_{\ell_{\nu_i\nu_j}}$:

$$C_{\ell_{\nu_i\nu_j}} = (C_{\nu_i}, C_{\nu_j}, \psi_{\ell_{\nu_i\nu_j}})$$

Denote by $B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(IN_i)$ respectively $B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(IN_j)$ the set of input-border conditions in the configuration $C_{\ell_{\nu_i\nu_j}}$ that are not assigned to output border-conditions:

$$B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(IN_i) = \left\{ b_{IN_i} \mid \psi_{\ell_{\nu_i\nu_j}}(b_{IN_i}) = \varepsilon \right\}$$
$$B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(IN_j) = \left\{ b_{IN_j} \mid \psi_{\ell_{\nu_i\nu_j}}(b_{IN_j}) = \varepsilon \right\}$$

Definition 49. *Given two local configurations* $C_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ *and* $C_{\nu_j} \in \mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$ *and an interpretation function* $\psi_{\ell_{\nu_i\nu_j}} \in \Psi_{\nu_i\nu_j}$ *we have that* $C_{\ell_{\nu_i\nu_j}} = (C_{\nu_i}, C_{\nu_j}, \psi_{\ell_{\nu_i\nu_j}})$ *is a consistent global configuration if:*

i) $C_{\ell_{\nu_i\nu_j}}$ *is acyclic*

ii) all the input border conditions of C_{ν_i} *and respectively* C_{ν_j} *are are assigned by the interpretation function* $\psi_{\ell_{\nu_i\nu_j}}$ *to output border conditions that is* $B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(IN_i) = \emptyset$ *and* $B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(IN_j) = \emptyset$*.*

Denote by $B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(OUT_j)$ respectively $B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(OUT_i)$ the set of output-border conditions in the configuration $C_{\ell_{\nu_i\nu_j}}$ that are not assigned to input-border conditions:

$$B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(OUT_j) = \left\{ b_{OUT_j} \mid \forall b_{IN_i} \in B_{C_{\nu_i}}(IN_i)\ \psi_{\ell_{\nu_i\nu_j}}(b_{IN_i}) \neq b_{OUT_j} \right\}$$
$$B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(OUT_i) = \left\{ b_{OUT_i} \mid \forall b_{IN_j} \in B_{C_{\nu_j}}(IN_j)\ \psi_{\ell_{\nu_i\nu_j}}(b_{IN_j}) \neq b_{OUT_i} \right\}$$

Consider an output border condition $b_{OUT_j} \in B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(OUT_j)$ of the local configuration C_{ν_j} that is not assigned in $C_{\ell_{\nu_i\nu_j}}$ to an input border condition of the local configuration C_{ν_i}. If b_{OUT_j} has no predecessor in $C_{\ell_{\nu_i\nu_j}}$ that is an unsigned input border condition (i.e. $\forall b'_{IN_i} \in B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(IN_i) \Rightarrow b'_{IN_i} \not\preceq b_{OUT_j}$ and $\forall b'_{IN_j} \in B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(IN_j) \Rightarrow b'_{IN_j} \not\preceq b_{OUT_j}$) then b_{OUT_j} is new input border condition to component i.

Denote by $B^{new}_{C_{\ell_{\nu_i\nu_j}}}(IN_i)$ the set of new input border conditions in component i:

$$\begin{aligned}B^{new}_{C_{\ell_{\nu_i\nu_j}}}(IN_i) = \Big\{ b_{OUT_j} \mid (\forall b'_{IN_i} \in B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(IN_i) \Rightarrow b'_{IN_i} \not\preceq b_{OUT_j}) \ \wedge \\ \wedge(\forall b'_{IN_j} \in B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(IN_j) \Rightarrow b'_{IN_j} \not\preceq b_{OUT_j})\Big\}\end{aligned}$$

In words $B^{new}_{C_{\ell_{\nu_i\nu_j}}}(IN_j)$ comprises the set of input-border conditions that result after merging C_{ν_i} and C_{ν_j} via the interpretation function $\psi_{\ell_{\nu_i\nu_j}}$ in the global configuration $C_{\ell_{\nu_i\nu_j}}$ and can be used by Ag_i to extend the configuration in $\mathcal{N}_i$.

Remark 11. *Notice that an output border condition $b'_{OUT_j} \in B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(OUT_j)$ of the local configuration C_{ν_j} that is not assigned in $C_{\ell_{\nu_i\nu_j}}$ to an input border condition of C_{ν_i} but has as predecessor in $C_{\ell_{\nu_i\nu_j}}$ an input-border condition that is not yet assigned is not used to extend $C_{\ell_{\nu_i\nu_j}}$ in component i until all its predecessors that are input-border conditions are assigned.*

Similarly denote by $B^{new}_{C_{\ell_{\nu_i\nu_j}}}(IN_j)$ the set of new input border conditions in component j:

$$\begin{aligned}B^{new}_{C_{\ell_{\nu_i\nu_j}}}(IN_j) = \Big\{ b_{OUT_i} \mid (\forall b'_{IN_i} \in B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(IN_i) \Rightarrow b'_{IN_i} \not\preceq b_{OUT_i}) \ \wedge \\ \wedge \ (\forall b'_{IN_j} \in B^{ua}_{C_{\ell_{\nu_i\nu_j}}}(IN_j) \Rightarrow b'_{IN_j} \not\preceq b_{OUT_i})\Big\}\end{aligned}$$

Definition 50. *Given two local configurations $C_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ and $C_{\nu_j} \in \mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$ and an interpretation function $\psi_{\ell_{\nu_i\nu_j}} \in \Psi_{\nu_i\nu_j}$ we have that $C_{\ell_{\nu_i\nu_j}} = (C_{\nu_i}, C_{\nu_j}, \psi_{\ell_{\nu_i\nu_j}})$ is an extendable configuration if there are new input-border conditions that is, either $B^{new}_{C_{\ell_{\nu_i\nu_j}}}(IN_i) \neq \emptyset$ or $B^{new}_{C_{\ell_{\nu_i\nu_j}}}(IN_j) \neq \emptyset$.*

Consider two local configurations $C_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ and $C_{\nu_j} \in \mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$ and an interpretation function $\psi_{\ell_{\nu_i\nu_j}} \in \Psi_{\nu_i\nu_j}$ s.t. $C_{\ell_{\nu_i\nu_j}} = (C_{\nu_i}, C_{\nu_j}, \psi_{\ell_{\nu_i\nu_j}})$ is an extendable configuration and consider that $B^{new}_{C_{\ell_{\nu_i\nu_j}}}(IN_i) \neq \emptyset$ and $B^{new}_{C_{\ell_{\nu_i\nu_j}}}(IN_j) \neq \emptyset$.

Then Ag_i extends the global configuration $C_{\ell_{\nu_i\nu_j}}$ in its local component i by appending unobservable events that become possible with the new input border conditions $B^{new}_{C_{\ell_{\nu_i\nu_j}}}(IN_i) \neq \emptyset$. $C_{\nu_i(\ell_{\nu_i\nu_j})}$ is an unobservable extensions of the global configuration $C_{\ell_{\nu_i\nu_j}}$ in $\mathcal{N}_i$ if:

1. $C_{\ell_{\nu_i\nu_j}} \sqsubseteq C_{\nu_i(\ell_{\nu_i\nu_j})}$
2. $\forall e \in E_{C_{\nu_i(\ell_{\nu_i\nu_j})}} \setminus E_{C_{\ell_{\nu_i\nu_j}}} \Rightarrow \phi(e) \in \mathcal{T}_{uo_i}$ and $\exists b^{new}_{IN_i} \in B^{new}_{C_{\ell_{\nu_i\nu_j}}}(IN_i)$ s.t. $b^{new}_{IN_i} \preceq e$

Iteratively Ag_i and Ag_j merge the local configurations in to global configurations and then extend the global configurations in their local components until a fix-point is achieved. Even though not all the global configurations can be extended in the local components consider for convenience the trivial extension of a local configuration by appending the empty event. The fix-point is achieved when all the global configurations cannot be extended by unobservable events but only by appending empty events.

Remark 12. *Notice that for a global configuration $C_{\ell_{\nu_i\nu_j}} = (C_{\nu_i}, C_{\nu_j}, \psi_{\ell_{\nu_i\nu_j}})$ a local agent Ag_i knows the local configuration C_{ν_i}, the interpretation function $\psi_{\ell_{\nu_i\nu_j}}$ and the name (index) of the local configuration in set of local configurations of component j to whom C_{ν_i} is matched namely C_{ν_j} but not what C_{ν_j} contains since Ag_i does not know the model of the neighbouring component j. Thus for a global configuration $C_{\ell_{\nu_i\nu_j}} = (C_{\nu_i}, C_{\nu_j}, \psi_{\ell_{\nu_i\nu_j}})$, Ag_i and Ag_j have the information $C_{\ell_{\nu_i\nu_j}} = (C_{\nu_i}, \nu_j, \psi_{\ell_{\nu_i\nu_j}})$ and $C_{\ell_{\nu_i\nu_j}} = (\nu_i, C_{\nu_j}, \psi_{\ell_{\nu_i\nu_j}})$ respectively.*

Remark 13. *We consider that Ag_i and Ag_j exchange simultaneously information at the very same time. This means that the first message sent by Ag_i to Ag_j is received by Ag_j at the same time when the first message sent by Ag_j to Ag_i is received by Ag_i. As already mentioned an asynchronous communication protocol where Ag_i receives first the message from Ag_j and then sends a message to Ag_j derived based on the local update of the local calculations of component i that includes what Ag_j just sent as a message brings nothing new but more notation.*

Denote by $\mathcal{C}_i^1(\mathcal{O}^i_{\theta_{com}})$ and $\mathcal{C}_j^1(\mathcal{O}^i_{\theta_{com}})$ the set of preliminary local configurations $\mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ respectively $\mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$ where the upper index 1 denotes that Ag_i and Ag_j derived the set of local configurations that are checked for consistency in the first communication round of the first communication session at the time θ_{com}:

$$\mathcal{C}_i^1(\mathcal{O}^i_{\theta_{com}}) = \left\{ C_{\nu_i^1} \mid \nu_i^1 \in \mathcal{V}^{1_i} \right\}$$

$$\mathcal{C}_j^1(\mathcal{O}^j_{\theta_{com}}) = \left\{ C_{\nu_j^1} \mid \nu_j^1 \in \mathcal{V}^{1_j} \right\}$$

For $C_{\nu_i^1}$ and $C_{\nu_j^1}$ we have $\Psi_{\nu_i^1\nu_j^1} = \left\{ \psi_{\ell_{\nu_i^1\nu_j^1}} \mid \ell_{\nu_i^1\nu_j^1} \in \mathcal{V}_{\nu_i^1\nu_j^1} \right\}$ the collection of interpretation functions of the local configurations $C_{\nu_i^1}$ and $C_{\nu_j^1}$.

Let $\mathcal{C}_{ij}^1(\mathcal{O}^i_{\theta_{com}})$ be the set of configurations obtained after the first communication round where:

$$\mathcal{C}_{ij}^1(\mathcal{O}^i_{\theta_{com}}) = \left\{ C_{\ell_{\nu_i^1\nu_j^1}} = (C_{\nu_i^1}, C_{\nu_j^1}, \psi_{\ell_{\nu_i^1\nu_j^1}}) \mid \nu_i^1 \in \mathcal{V}_i^1; \nu_j^1 \in \mathcal{V}_j^1; \ell_{\nu_i^1\nu_j^1} \in \mathcal{V}_{\nu_i^1\nu_j^1} \right\}$$

Denote by $\mathcal{C}^{2_i}_{\ell_{\nu^1_i\nu^1_j}}$ the set of configurations that are unobservable extensions of the configuration $C_{\ell_{\nu^1_i\nu^1_j}}$ in the local component i:

$$\mathcal{C}^{2_i}_{\ell_{\nu^1_i\nu^1_j}} = \left\{ C_{\nu^2_i(\ell_{\nu^1_i\nu^1_j})} \mid \nu^2_i(\ell_{\nu^1_i\nu^1_j}) \in \mathcal{V}^{2_i}_{\ell_{\nu^1_i\nu^1_j}} \right\}$$

and similarly denote by $\mathcal{C}^{2_j}_{\ell_{\nu^1_i\nu^1_j}}$ the set of configurations that are unobservable extensions of the configuration $C_{\ell_{\nu^1_i\nu^1_j}}$ in j:

$$\mathcal{C}^{2_j}_{\ell_{\nu^1_i\nu^1_j}} = \left\{ C_{\nu^2_j(\ell_{\nu^1_i\nu^1_j})} \mid \nu^2_j(\ell_{\nu^1_i\nu^1_j}) \in \mathcal{V}^{2_j}_{\ell_{\nu^1_i\nu^1_j}} \right\}$$

and then denote by $\mathcal{C}^2_i(\mathcal{O}^i_{\theta_{com}})$ and $\mathcal{C}^2_j(\mathcal{O}^j_{\theta_{com}})$ the set of unobservable extensions in $\mathcal{N}_i$ respectively $\mathcal{N}_j$ of the global configurations $\mathcal{C}^1_{ij}(\mathcal{O}^i_{\theta_{com}})$ derived after the first communication round:

$$\mathcal{C}^2_i(\mathcal{O}^i_{\theta_{com}}) = \left\{ C_{\nu^2_i(\ell_{\nu^1_i\nu^1_j})} \mid \nu^2_i(\ell_{\nu^1_i\nu^1_j}) \in \mathcal{V}^{2_i}_{\ell_{\nu^1_i\nu^1_j}}; \ell_{\nu^1_i\nu^1_j} \in \mathcal{V}^1_{\nu^1_i\nu^1_j} \right\}$$

$$\mathcal{C}^2_j(\mathcal{O}^j_{\theta_{com}}) = \left\{ C_{\nu^2_j(\ell_{\nu^1_i\nu^1_j})} \mid \nu^2_j(\ell_{\nu^1_i\nu^1_j}) \in \mathcal{V}^{2_j}_{\ell_{\nu^1_i\nu^1_j}}; \ell_{\nu^1_i\nu^1_j} \in \mathcal{V}^1_{\nu^1_i\nu^1_j} \right\}$$

For two local configurations $C_{\nu^2_i(\ell_{\nu^1_i\nu^1_j})} \in \mathcal{C}^2_i(\mathcal{O}^i_{\theta_{com}})$ and $C_{\nu^2_j(\ell_{\nu^1_i\nu^1_j})} \in \mathcal{C}^2_j(\mathcal{O}^j_{\theta_{com}})$ obtained after extending the global configurations matched in first communication round denote by $\Psi_{\nu^2_i\nu^2_i(\ell_{\nu^1_i\nu^1_j})}$ the collection of interpretation functions:

$$\Psi_{\nu^2_i\nu^2_i(\ell_{\nu^1_i\nu^1_j})} = \left\{ \psi_{\ell_{\nu^2_i\nu^2_i(\ell_{\nu^1_i\nu^1_j})}} \mid \ell_{\nu^2_i\nu^2_i}(\ell_{\nu^1_i\nu^1_j}) \in \mathcal{V}_{\nu^2_i\nu^2_i(\ell_{\nu^1_i\nu^1_j})} \right\}$$

Denote by $\mathcal{C}^2_{ij}(\mathcal{O}^i_{\theta_{com}})$ the set of configurations obtained after the second communication round:

$$\mathcal{C}^2_{ij}(\mathcal{O}^i_{\theta_{com}}) = \left\{ C_{\ell_{\nu^2_i\nu^2_j(\nu^1_i\nu^1_j)}} \mid \nu^2_i(\ell_{\nu^1_i\nu^1_j}) \in \mathcal{V}^{2_i}_{\ell_{\nu^1_i\nu^1_j}} \quad \nu^2_j(\ell_{\nu^1_i\nu^1_j}) \in \mathcal{V}^{2_j}_{\ell_{\nu^1_i\nu^1_j}} \right.$$
$$\left. \ell_{\nu^2_i\nu^2_j}(\ell_{\nu^1_i\nu^1_j}) \in \mathcal{V}_{\nu^2_i\nu^2_j(\ell_{\nu^1_i\nu^1_j})} \right\}$$

where a global configuration is:

$$C_{\ell_{\nu^2_i\nu^2_j(\ell_{\nu^1_i\nu^1_j})}} = (C_{\nu^2_i(\ell_{\nu^1_i\nu^1_j})}, C_{\nu^2_j(\ell_{\nu^1_i\nu^1_j})}, \psi_{\ell_{\nu^2_i\nu^2_i(\ell_{\nu^1_i\nu^1_j})}})$$

Denote by $\mathcal{C}_i^{k+1}(\mathcal{O}^i_{\theta_{com}})$ and $\mathcal{C}_j^{k+1}(\mathcal{O}^j_{\theta_{com}})$ the set of unobservable extensions of the global configurations derived after the k^{th} communication round in component $\mathcal{N}_i$ and component $\mathcal{N}_j$ respectively:

$$\mathcal{C}_i^{k+1}(\mathcal{O}^i_{\theta_{com}}) = \left\{ C_{\nu_i^{k+1}(\ell_{\nu_i^k\nu_j^k})} \mid \nu_i^{k+1}(\ell_{\nu_i^k\nu_j^k}) \in \mathcal{V}^{k+1_i}_{\ell_{\nu_i^k\nu_j^k}}; \ell_{\nu_i^k\nu_j^k} \in \mathcal{V}^{k+1}_{\ell_{\nu_i^k\nu_j^k}} \right\}$$

$$\mathcal{C}_j^{k+1}(\mathcal{O}^j_{\theta_{com}}) = \left\{ C_{\nu_j^{k+1}(\ell_{\nu_i^k\nu_j^k})} \mid \nu_j^{k+1}(\ell_{\nu_i^k\nu_j^k}) \in \mathcal{V}^{k+1_j}_{\ell_{\nu_i^k\nu_j^k}}; \ell_{\nu_i^k\nu_j^k} \in \mathcal{V}^{k+1}_{\ell_{\nu_i^k\nu_j^k}} \right\}$$

Inductively after the k^{th} communication round at the first communication session at the time θ_{com} we have the set of global configurations:

$$\mathcal{C}_{ij}^{k+1} = \Big\{ C_{\ell_{\nu_i^{k+1}\nu_j^{k+1}(\ell_{\nu_i^k\nu_j^k})}} \mid \nu_i^{k+1}(\ell_{\nu_i^k\nu_j^k}) \in \mathcal{V}^{k+1_i}_{\ell_{\nu_i^k\nu_j^k}}; \nu_j^{k+1}(\ell_{\nu_i^k\nu_j^k}) \in \mathcal{V}^{k+1_j}_{\ell_{\nu_i^k\nu_j^k}};$$
$$\ell_{\nu_i^{k+1}\nu_j^{k+1}}(\ell_{\nu_i^k\nu_j^k}) \in \mathcal{V}_{\nu_i^{k+1}\nu_j^{k+1}(\ell_{\nu_i^k\nu_j^k})} \Big\}$$

where a global configuration is:

$$C_{\ell_{\nu_i^{k+1}\nu_j^{k+1}(\ell_{\nu_i^k\nu_j^k})}} = (C_{\nu_i^{k+1}(\ell_{\nu_i^k\nu_j^k})}, C_{\nu_j^{k+1}(\ell_{\nu_i^k\nu_j^k})}, \psi_{\ell_{\nu_i^{k+1}\nu_i^{k+1}(\ell_{\nu_i^k\nu_j^k})}})$$

Algorithm 8 Update_Local_Calculation (Ag_i considered)

```
Require: $\mathcal{C}_{ij}^{k+1}(\mathcal{O}^i_{\theta_{com}})$; $\mathcal{MSG}^{k+1}_{j\to i}$
Ensure: $\mathcal{C}_i^{k+1}(\mathcal{O}^i_{\theta_{com}})$
for all $C_{\ell_{\nu_i^{k+1}\nu_j^{k+1}}} \in \mathcal{C}_{ij}^{k+1}(\mathcal{O}^i_{\theta_{com}})$ do
    calculate $\mathcal{C}^{k+1}_{\nu_i(\ell_{\nu_i^{k+1}\nu_j^{k+1}})}$ {the set of unobs. extensions}
    $\mathcal{C}_i^{k+1}(\mathcal{O}^i_{\theta_{com}}) = \mathcal{C}_i^{k+1}(\mathcal{O}^i_{\theta_{com}}) \cup \left\{ \mathcal{C}^{k+1}_{\nu_i^{k+1}(\ell_{\nu_i^{k+1}\nu_j^{k+1}})} \mid \ell_{\nu_i^{k+1}\nu_j^{k+1}} \in \mathcal{V}_{\nu_i^{k+1}\nu_j^{k+1}} \right\}$
end for
```

Remark 14. *Notice that the conditions for checking the execution of a cycle are not included in the Algorithm 8. This can be implemented as follows. Consider two local explanations that were matched in to a global configuration $C_{\ell_{\nu_i^k\nu_j^k}}$ after the k^{th} communication round.*

Then consider $C_{\nu_i^{k+1}(\ell_{\nu_i^k\nu_j^k})}$ and $C_{\nu_j^{k+1}(\ell_{\nu_i^k\nu_j^k})}$ two unobservable extensions of $C_{\ell_{\nu_i^k\nu_j^k}}$ in component i respectively in component j. If $mark(C_{\nu_i^{k+1}(\ell_{\nu_i^k\nu_j^k})}) = mark(C_{\nu_i^k(\ell_{\nu_i^{k-1}\nu_j^{k-1}})})$ then it means that the marking repeats in component i. If this is also true in component j ($mark(C_{\nu_j^{k+1}(\ell_{\nu_i^k\nu_j^k})}) = mark(C_{\nu_j^k(\ell_{\nu_i^{k-1}\nu_j^{k-1}})})$

then the pair of the two local configuration $(C_{\nu_i^{k+1}(\ell_{\nu_i^k \nu_j^k})}, C_{\nu_j^{k+1}(\ell_{\nu_i^k \nu_j^k})})$ *is discarded.*

This is because the execution of the cycle repeats the marking and any unobservable cycle does not contain fault events. This is a natural assumption since otherwise the faults within the cycle are not diagnosable [SSL$^+$95].

4.3.3 The main result

Consider the following distributed diagnosis algorithm presented for the case of only two agents: Ag_i and Ag_j (see Algorithm 9).

Algorithm 9 DD_Algo_2 for two agents: Ag_i,Ag_j (Ag_i considered)

```
Require: $\langle \mathcal{N}_i, M_{0_i} \rangle$, $\mathcal{T}_{o_i}$, $\mathcal{T}_{uo_i}$, $\mathcal{O}^{\theta}_{\theta_{com}}$
Ensure: $\mathcal{C}_{ij}(\mathcal{O}^i_{\theta_{com}})$
k = 1;
Preliminary_Local_Calculation($\mathcal{O}_{n_i}$, $\theta_{com}$)
Communication_exchange($\mathcal{C}^k_i(\mathcal{O}^i_{\theta_{com}}$)
repeat
    send $\mathcal{MSG}^k_{i \to j}$
    receive $\mathcal{MSG}^k_{j \to i}$
    Update_Local_Calculation($\mathcal{C}^k_i(\mathcal{O}^i_{\theta_{com}})$,$\mathcal{MSG}^k_{j \to i}$)
    Communication_exchange($\mathcal{C}^{k+1}_{ij}(\mathcal{O}_{\theta_{com}})$){Update_Message}
    if $\mathcal{MSG}^{k+1}_{i \to j} = \emptyset$ then
        send $\mathcal{MSG}^{k+1}_{i \to j} = stop$
    end if
    $k = k + 1$
until $\mathcal{MSG}^{k+1}_{j \to i} = stop$
$\mathcal{C}_{ij}(\mathcal{O}_{\theta_{com}}) = \mathcal{C}^{k+1}_{ij}(\mathcal{O}^i_{\theta_{com}})$
```

Denote by $c^i_{\wp}$ and $c^j_{\wp}$ how many times an unobservable oriented path $\wp$ that starts in $p \in \mathcal{P}_i$, respectively $p \in \mathcal{P}_j$ alternates places of $\mathcal{P}_i \setminus \mathcal{P}_{ij}$ and places of $\mathcal{P}_j \setminus \mathcal{P}_{ij}$ respectively. Let $c_{\wp} = max(c^i_{\wp}, c^j_{\wp})$ and denote $K_c = \mathtt{max}_{\wp \in \mathcal{N}}(c_{\wp})$ ($\mathcal{N} = \mathcal{N}_i \cup \mathcal{N}_j$). If $\exists \wp \in \mathcal{N}$ s.t. $\wp$ is an unobservable elementary circuit then $K_c = \infty$ otherwise K_c is finite.

Proposition 15. *Consider a distributed description of the plant comprising two components and two local agents and a distributed observation* $\mathcal{O}_{\theta_{com}} = \mathcal{O}^i_{\theta_{com}} \otimes^{gc} \mathcal{O}^j_{\theta_{com}}$. *There exists a finite number k of communication rounds at the time θ_{com} such that after the k^{th} communication round the algorithm* DD_Algo_2 *stops (the distributed calculations achieved a fix-point). We have that:*

i) if K_c is infinite then $\exists$ $K_{max} \in \mathbb{N}^+$ finite s.t. at the K^{th}_{max} communication *round the distributed calculation achieves a fix-point (K_{max} depends on the PN topology, the initial marking M_0 and the observation $\mathcal{O}_{\theta_{com}}$).*

ii) if K_c is finite then $K_{max} \leq K_c$ (K_c depends only on the PN topology)

Proof. First we prove $i)$.

Consider the set of preliminary local configurations derived by the local agents before starting to communicate at the time θ_{com}:

$$\mathcal{C}_i^1(\mathcal{O}^i_{\theta_{com}}) = \left\{ C_{\nu_i^1} \mid \nu_i^1 \in \mathcal{V}^{1_i} \right\}$$

$$\mathcal{C}_j^1(\mathcal{O}^j_{\theta_{com}}) = \left\{ C_{\nu_j^1} \mid \nu_j^1 \in \mathcal{V}^{1_j} \right\}$$

Then consider an arbitrary pair of local configurations $(C_{\nu_i^1}, C_{\nu_j^1}) \in \mathcal{C}_i^1(\mathcal{O}^i_{\theta_{com}}) \times \mathcal{C}_j^1(\mathcal{O}^j_{\theta_{com}})$.

Denote by $M_{\nu_i^1}(IN_i)$ the sub-vector marking of the input places $\mathcal{P}_{IN_i}$ in $C_{\nu_i^1}$ that correspond with the set of input border conditions considered in $C_{\nu_i^1}$, i.e. $M_{\nu_i^1}(IN_i) = \phi(B_{C_{\nu_i^1}}(IN_i))$. Similarly denote $M_{\nu_j^1}(IN_j)$ the sub-vector marking that corresponds with the set of input border conditions considered in $C_{\nu_j^1}$, i.e. $M_{\nu_j^1}(IN_j) = \phi(B_{C_{\nu_j^1}}(IN_j))$.

Denote by $M_{\nu_i^1\nu_j^1}$ the marking obtained from the sub-vector markings $M_{0_i}, M_{0_j}, M_{\nu_i^1}(IN_i)$, and $M_{\nu_j^1}(IN_j)$:

$$M_{\nu_i^1\nu_j^1} = M_{0_i} \uplus M_{0_j} \uplus M_{\nu_i^1}(IN_i) \uplus M_{\nu_j^1}(IN_j)$$

where $\uplus$ denotes the union with addition of two multi-sets and a marking is considered a multi-set of tokens.

Let $\langle \mathcal{N}, M_{\nu_i^1\nu_j^1} \rangle$ be the overall plant model $\mathcal{N}$ with the initial marking $M_{\nu_i^1\nu_j^1}$. By Proposition 13 we have that $M_{\nu_i^1\nu_j^1}$ is finite and then by assumption $viii)$ in the setting we have that $\langle \mathcal{N}, M_{\nu_i^1\nu_j^1} \rangle$ is bounded w.r.t. the unobservable evolution.

Let $\mathcal{U}_{\mathcal{N}}(M_{\nu_i^1\nu_j^1})$ be the net-unfolding of $\langle \mathcal{N}, M_{\nu_i^1\nu_j^1} \rangle$. Then consider the (finite) sequence of observed events $\mathcal{O}_n$ and denote by $\mathcal{U}_{\mathcal{N}}(M_{\nu_i^1\nu_j^1}, \mathcal{O}_n)$ the prefix of $\mathcal{U}_{\mathcal{N}}(M_{\nu_i^1\nu_j^1})$ that correspond with the observation. Denote by $\mathcal{C}(M_{\nu_i^1\nu_j^1}, \mathcal{O}_n)$ the set of configurations that obey the observation $\mathcal{O}_n$.

Consider that all unobservable cycles in $\langle \mathcal{N}, M_{\nu_i^1\nu_j^1} \rangle$ that repeat the marking are either executed only once or are not at all executed.

Since we have a finite number of observed events we have that any configuration $C(M_{\nu_i^1\nu_j^1}, \mathcal{O}_n) \in \mathcal{C}(M_{\nu_i^1\nu_j^1}, \mathcal{O}_n)$ contains a finite number of condition nodes and a finite number of event nodes.

We have that for any interpretation function $\psi_{\ell_{\nu_i^i \nu_j^i}} \in \Psi_{\nu_i^1 \nu_j^1}$ the global configuration $C_{\ell_{\nu_i^1 \nu_j^1}} = (C_{\nu_i^1}, C_{\nu_j^1}, \psi_{\ell_{\nu_i^i \nu_j^i}})$ is a configuration in $\mathcal{U}_{\mathcal{N}}(M_{\nu_i^1 \nu_j^1}, \mathcal{O}_n)$.

Inductively we have that any global configuration $C_{\ell_{\nu_i^{k+1} \nu_j^{k+1}}}$ that results at the $k+1$ communication round by merging $C_{\nu_i^{k+1}}$ and $C_{\nu_j^{k+1}}$ according with the interpretation function $\psi_{\ell_{\nu_i^{k+1} \nu_j^{k+1}}} \in \Psi^{k+1}_{\nu_i^{k+1} \nu_j^{k+1}}$ is also a configuration in $\mathcal{U}_{\mathcal{N}}(M_{\nu_i^1 \nu_j^1}, \mathcal{O}_n)$.

We have that running the distributed algorithm Algorithm 9 the two agents detect the cycles that repeat the marking.

Since any configuration $C(M_{\nu_i^1 \nu_j^1}, \mathcal{O}_n) \in \mathcal{C}(M_{\nu_i^1 \nu_j^1}, \mathcal{O}_n)$ contains a finite number of condition nodes and a finite number of event nodes and any configuration $C_{\ell_{\nu_i^{k+1} \nu_j^{k+1}}(\nu_i^k \nu_j^k)}$ that is obtained extending $C_{\ell_{\nu_i^1 \nu_j^1}}$ at the $k+1$ communication is a configuration in $\mathcal{U}_{\mathcal{N}}(M_{\nu_i^1 \nu_j^1}, \mathcal{O}_n)$ we have that there exists a K_{max} finite s.t. after the K_{max}^{th} communication round the agents send the message *stop*. This is trivial since otherwise the local agents would generate configurations that contain an infinite number of condition-nodes or event-nodes that would contradict the assumption $viii$) in the setting.

ii) is proved as follows. We have that there are no circuits that contain places of $\mathcal{N}_i$ and $\mathcal{N}_j$ and K_c is the maximum number of how many times an unobservable oriented path $\wp$ that starts in a place in $\mathcal{P}_i$ or $\mathcal{P}_j$ crosses successively $\mathcal{N}_j$ and $\mathcal{N}_i$.

Assume that the distributed algorithm terminates after K_{max} communications rounds and $K_{max} > K_c$. Since at each communication round the agents sends a non-empty set of tokens that are newly available at the border places it means that there exists at least a configuration $C_{\ell_{\nu_i^{K_{max}} \nu_j^{K_{max}}}}$ that can be obtained only after K_{max} communication rounds. It means that $C_{\ell_{\nu_i^{K_{max}} \nu_j^{K_{max}}}}$ contains an unobservable path that alternates K_{max} times condition-nodes that correspond with $\mathcal{P}_i$ and $\mathcal{P}_j$ respectively. Since $C_{\ell_{\nu_i^{K_{max}} \nu_j^{K_{max}}}}$ is a configuration in $\mathcal{U}_{\mathcal{N}}(M_{\nu_i^1 \nu_j^1}, \mathcal{O}_n)$ we have that there exists an unobservable oriented path $\wp$ in $\mathcal{N}$ s.t. $c(\wp) > K_c$ that contradicts that $K_c = \mathtt{max}_{\wp \in \mathcal{N}}(c_{\wp})$

We have considered an arbitrary pair of local configurations $(C_{\nu_i^1}, C_{\nu_j^1}) \in \mathcal{C}_i^1(\mathcal{O}^i_{\theta_{com}}) \times \mathcal{C}_j^1(\mathcal{O}^j_{\theta_{com}})$. Thus the distributed algorithm terminates for all the pairs in the cartesian product $\mathcal{C}_i^1(\mathcal{O}^i_{\theta_{com}}) \times \mathcal{C}_j^1(\mathcal{O}^j_{\theta_{com}})$. This completes the proof. □

Let $C^k_{ij}(\mathcal{O}_{\theta_{com}})$ be the set of global configurations derived when the distributed algorithm *DD_Algo_2* achieves a fix-point.

Denote by $\mathcal{C}^{gcon}_{ij}(\mathcal{O}_{\theta_{com}})$ the set of global configuration that are consistent:

$$\mathcal{C}^{gcon}_{ij}(\mathcal{O}^i_{\theta_{com}}) = \Big\{ C_{\upsilon_i^{k+1}\upsilon_j^{k+1}} \in \mathcal{C}^{k+1}_{ij}(\mathcal{O}^i_{\theta_{com}}) \mid C_{\upsilon_i^{k+1}\upsilon_j^{k+1}} \text{ is consistent } \Big\}$$

and then denote by $\mathcal{C}^{gcon}_{i}(\mathcal{O}^i_{\theta_{com}})$ the set of local configurations in component i that are part of consistent global configurations

$$\mathcal{C}^{gcon}_{i}(\mathcal{O}^i_{\theta_{com}}) = \big\{ C_{\upsilon_i} \mid C_{\upsilon_i\upsilon_j} \in \mathcal{C}^{gcon}_{ij}(\mathcal{O}^i_{\theta_{com}}), C_{\upsilon_i\upsilon_j} = (C_{\upsilon_i}, C_{\upsilon_j}, \psi_{\upsilon_i\upsilon_j}) \big\}$$

Theorem 3. *Consider the set of global configurations $\mathcal{C}(\mathcal{O}_{\theta_{com}})$ in the global model $\langle \mathcal{N}, M_0 \rangle$ derived by a centralized agent having the overall plant observation $\mathcal{O}_{\theta_{com}}$ and the set of consistent global configurations $\mathcal{C}^{gcon}_{ij}(\mathcal{O}^i_{\theta_{com}})$ derived by a local agent Ag_i when the distributed algorithm* DD_Algo_2 *achieves the fix-point. We have that:*

i) for any global consistent configuration $C_{\nu_i\nu_j} \in \mathcal{C}^{gcon}_{ij}(\mathcal{O}_{\theta_{com}})$ there exists a global configuration $C_\nu \in \mathcal{C}(\mathcal{O}_{\theta_{com}})$ s.t. $C_{\nu_i\nu_j} = C_\nu$.

ii) for any global configuration $C_\nu \in \mathcal{C}(\mathcal{O}_{\theta_{com}})$ there exists a global consistent configuration $C_{\nu_i\nu_j} \in \mathcal{C}^{gcon}_{ij}(\mathcal{O}_{\theta_{com}})$ s.t. $C_\nu = C_{\nu_i\nu_j}$ where $C_{\nu_i\nu_j} = (C_{\nu_i}, C_{\nu_j}, \psi_{\nu_i\nu_j})$.

that together imply that:

$$\mathcal{C}(\mathcal{O}_{\theta_{com}}) = \mathcal{C}^{gcon}_{ij}(\mathcal{O}_{\theta_{com}})$$

Proof. First we prove i).

Consider a global configuration $C_{\nu_i^k\nu_j^k} \in \mathcal{C}^{gcon}_{ij}(\mathcal{O}_{\theta_{com}})$, that was found consistent at the k^{th} communication round ($k \geq 1$) and let $C_{\nu_i^k\nu_j^k}$ be obtained by extending a pair of preliminary local configurations $(C_{\nu_i^1}, C_{\nu_j^1})$.

Since $C_{\nu_i^k\nu_j^k}$ is consistent we have that $B^{ua}_{\nu_i^k\nu_j^k}(IN_i) = \emptyset$ and $B^{ua}_{\nu_j^k\nu_j^k}(IN_j) = \emptyset$ that is there are no border-conditions that correspond to the input places $\mathcal{P}_{IN_i}$ and $\mathcal{P}_{IN_j}$ that have no predecessors in $C_{\nu_i^k\nu_j^k}$.

We have that $C_{\nu_i^k\nu_j^k}$ is a configuration in $\mathcal{U}_\mathcal{N}(M_{\nu_i^1\nu_j^1}, \mathcal{O}_{\theta_{com}})$, that is the prefix of net unfolding of $\langle \mathcal{N}, M_{\nu_i^1\nu_j^1} \rangle$ that corresponds with the observation $\mathcal{O}_{\theta_{com}}$.

Then we have that $M_{\nu_i^1\nu_j^1} \geq M_0$. If $M_{\nu_i^1\nu_j^1} = M_0$ the proof is trivial. Consider that $M_{\nu_i^1\nu_j^1} > M_0$. Since $C_{\nu_i^k\nu_j^k}$ is such that all the border-conditions have predecessors its means that the tokens that were initially assumed on the border places in $M_{\nu_i^1\nu_j^1}$ are not used for deriving $C_{\nu_i^k\nu_j^k}$. Thus $C_{\nu_i^k\nu_j^k}$ is a configuration in $\mathcal{U}_\mathcal{N}(M_0, \mathcal{O}_{com})$ that is $C_{\nu_i^k\nu_j^k} \in \mathcal{C}(\mathcal{O}_{\theta_{com}})$.

The proof of ii) is straightforward since we have made calculations in $\mathcal{U}_\mathcal{N}(M_{\nu_i^1\nu_j^1}, \mathcal{O}_{\theta_{com}})$ with $M_{\nu_i^1\nu_j^1} \geq M_0$. □

Theorem 4. *Consider a distributed description of the plant comprising two components and two local agents and a distributed observation $\mathcal{O}_{\theta_{com}} = \mathcal{O}^i_{\theta_{com}} \otimes^{gc} \mathcal{O}^j_{\theta_{com}}$. The global consistent local diagnosis $\mathcal{LD}^{gcon}_i$ derived by the agent Ag_i at the time θ_{com} by running the distributed algorithm* DD_Algo_2 *is such that:*

i) if K_c is infinite then $\exists\ K_{max} \in \mathbb{N}^+$ finite s.t. after $K_{\max}$ communication rounds $\mathcal{LD}^{gcon}_i(\mathcal{O}^i_{\theta_{com}}) = \mathcal{D}_i(\mathcal{O}_{\theta_{com}})$ where K_{max} depends on both the PN topology and the initial marking M_0.

ii) if K_c is finite then after $K_{max} \leq K_c$ communication rounds, the consistent local diagnosis result $\mathcal{LD}^{gcon}_i(\mathcal{O}^i_{\theta_{com}})$ recovers the centralized diagnosis result of component i, i.e. $\mathcal{LD}^{gcon}_i(\mathcal{O}^i_{\theta_{com}}) = \mathcal{D}_i(\mathcal{O}_{\theta_{com}})$

Proof. The proof is straightforward by Theorem 3. The diagnosis is obtained projecting equal sets on to the set of fault events. □

4.3.4 A particular case - $K_c = 1$

In this section we make the assumption that the PN model under investigation satisfies the following additional condition.

Assumption 5. *We assume in the following that $K_{c_{ij}} \leq 1$ that is any oriented path $\wp_i$ in $\mathcal{N}_i$ that starts in a place $p_{0_i} \in \mathcal{P}_{IN_i}$ and terminates in a place $p_{m_i} \in \mathcal{P}_{OUT_i}$ contains at least one observable event $t^o_{q_i} \in \mathcal{T}_{o_i}$ and identically for component j.*

At the time θ_{com} when the first communication session is allowed consider that the local agents Ag_i and Ag_j have derived the set of local configurations $\mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ and respectively $\mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$ (where the upper script 1 is dropped).

Consider two local configurations $\mathcal{C}_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$, $\mathcal{C}_{\nu_j} \in \mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$ and the family of interpretation functions of their border-conditions:

$$\Psi_{\nu_i\nu_j} = \left\{\psi_{\ell_{\nu_i\nu_j}} \mid \ell_{\nu_i\nu_j} \in \mathcal{V}_{\nu_i\nu_j}\right\}$$

Let a global configuration $C_{\ell_{\nu_i\nu_j}}$ be obtained merging the border conditions of the local configurations $\mathcal{C}_{\nu_i}$ and $\mathcal{C}_{\nu_j}$ according with the interpretation function $\psi_{\ell_{\nu_i\nu_j}} \in \Psi_{\nu_i\nu_j}$:

$$C_{\ell_{\nu_i\nu_j}} = (\mathcal{C}_{\nu_i}, \mathcal{C}_{\nu_j}, \psi_{\ell_{\nu_i\nu_j}})$$

Since any oriented path $\wp_i$ in $\mathcal{N}_i$ that starts in a place $p \in \mathcal{P}_{IN_i}$ and terminates in $p' \in \mathcal{P}_{OUT_i}$ contains at least one observable event we have that any unobservable extension of $C_{\ell_{\nu_i\nu_j}}$ in component i (e.g. $C_{\nu_i(\ell_{\nu_i\nu_j})}$) and component j (e.g. $C_{\nu_j(\ell_{\nu_i\nu_j})}$) does not "produce" new output border conditions that is $B_{C_{\nu_i(\ell_{\nu_i\nu_j})}}(OUT_i) = B_{C_{\nu_i}}(OUT_i)$ and $B_{C_{\nu_j(\ell_{\nu_i\nu_j})}}(OUT_i) = B_{C_{\nu_j}}(OUT_j)$.

It means that the distributed algorithm *DD_Algo_2* achieves a fix-point after one communication round. Moreover we have that there is enough to derive the existence of an interpretation function $\psi_{\ell_{\nu_i \nu_j}}$ such that $C_{\ell_{\nu_i \nu_j}} = (C_{\nu_i}, C_{\nu_j}, \psi_{\ell_{\nu_i \nu_j}})$ is a consistent global configuration that is also sufficient for continuing to derive the future diagnosis of the plant. In other words, for the future calculations the way the tokens on the border are matched is not important.

Given two local configurations $\mathcal{C}_{\nu_i} \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ and $\mathcal{C}_{\nu_j} \in \mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$ the existence of an interpretation function $\psi_{\ell_{\nu_i \nu_j}} \in \Psi_{\nu_i \nu_j}$ can be checked easily as follows.

Let the observation received in component i up to the time θ_{com} be $\mathcal{O}^i_{\theta_{com}}$ where $\mathcal{O}^i_{\theta_{com}}$ comprises n_i observed events.

For $k_i = 1_i, \ldots, n_i$ denote by $B^{k_i}_{C_{\nu_i}}(IN_i)$ the set of input-border conditions required to have been satisfied before the time $\theta^o_{t_{k_i}}$ when the observable event $t^o_{k_i}$ happened in the plant.

$$B^{k_i}_{C_{\nu_i}}(IN_i) = \left\{ b_{IN_i} \mid \theta_{b_{IN_i}} \leq \theta^o_{t_{k_i}} \right\}$$

Denote by $B^{k_i}_{C_{\nu_i}}(OUT_i)$ the set of output-border conditions that can be provided before the time $\theta^o_{k_i}$ when the observable event $t^o_{k_i}$ happened in the plant:

$$B^{k_i}_{C_{\nu_i}}(OUT_i) = \left\{ b_{OUT_i} \mid \theta_{b_{OUT_i}} \geq \theta^o_{k_i} \right\}$$

Let $M^{k_i}_{\nu_i}(IN_i)$ and $M^{k_i}_{\nu_i}(OUT_i)$ be the markings that correspond via ϕ with $B^{k_i}_{C_{\nu_i}}(IN_i)$ and $B^{k_i}_{C_{\nu_i}}(OUT_i)$ respectively .

Similarly denote by $M^{k_j}_{\nu_j}(IN_j)$ and $M^{k_j}_{\nu_j}(OUT_j)$ the markings that correspond to $B^{k_j}_{C_{\nu_j}}(IN_j)$ and respectively $B^{k_j}_{C_{\nu_j}}(OUT_j)$ via ϕ where $B^{k_j}_{C_{\nu_j}}(IN_j)$ and respectively $B^{k_j}_{C_{\nu_j}}(OUT_j)$ are derived considering that the local observation in component j comprises n_j observed events and $k_j = 1_j \ldots n_j$.

Proposition 16. $\forall (C_{\nu_i}, C_{\nu_j}) \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}}) \times \mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$ *we have that* $\Psi_{ij} \neq \emptyset$ *iff* *all the inequalities in Eq.4.17 are satisfied:*

$$\begin{cases} \sum\limits_{q_i=1_i}^{q_i=k_i} M^{q_i}_{\nu_i}(IN_i) \leq \sum\limits_{q_j=0_j}^{q_j=k_j} M^{q_j}_{\nu_j}(OUT_j) \quad \textit{with } \theta_{k_j} < \theta_{k_i} < \theta_{k_{j+1}} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \textit{for } k_i = 1_i \dots n_i \\ \sum\limits_{q_j=1_j}^{q_j=k_j} M^{q_j}_{\nu_j}(IN_j) \leq \sum\limits_{q_i=0_i}^{q_i=k_j} M^{q_i}_{\nu_i}(OUT_i) \quad \textit{with } \theta_{k_i} < \theta_{k_j} < \theta_{k_{i+1}} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \textit{for } k_j = 1_j \dots n_j \end{cases} \tag{4.17}$$

Proof. The proof is straightforward. □

In words Proposition 16 says that two local configurations C_{ν_i} and C_{ν_j} are consistent if for any input place of $\mathcal{N}_i$, resp. $\mathcal{N}_j$ the number of tokens required to have entered a local site i (resp. j) for explaining the local observations $t^o_{1_i}, \dots t^o_{n_i}$ (resp. $t^o_{1_j}, \dots t^o_{n_j}$) is lower than the number of tokens that could have exited component j (resp. component i).

If the inequalities in 4.17 are satisfied and moreover $\sum_{q_i=1_i}^{q_i=n_i} M^{q_i}_{\nu_i}(IN_i) < \sum_{q_j=0_j}^{q_j=n_j} M^{q_j}_{\nu_j}(OUT_j)$ or $\sum_{q_j=1_j}^{q_j=n_j} M^{q_j}_{\nu_j}(IN_j) < \sum_{q_i=0_i}^{q_i=n_i} M^{q_i}_{\nu_i}(OUT_i)$ then C_{ν_i} or C_{ν_j} can be extended further generating new consistent pairs.

Notice that if it is required to detect the faults that for sure have happened, then one can skip the calculation of the new generated pairs. Moreover, we can forget the timing information encoded to $B^{new}_{\nu_i}(IN_i)$ and $B^{new}_{\nu_j}(IN_j)$ because there is required to achieve consistency at the time θ_{com} when the information exchange takes place.

4.3.5 The case of three or more components

As presented for the case of two agents, the information exchange comprises information about the border-conditions of the two agents. As it will shown in the following for more than two agents this is in general not enough.

Since the goal of the distributed algorithm is to recover the results of the centralized agents by local results that are consistent, in the case when the plant comprises more than two components we have two notions of consistency namely *local consistency* and *global consistency* [SW04].

To illustrate this consider $\mid J \mid > 2$ and that each local agent Ag_i, $i \in J$ has derived before communicating with its neighbours the set of preliminary configurations $\mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$.

Remark 15. *Notice that as in the case of two agents the set of preliminary configurations $\mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$, $i \in J$ are matched and extended. Since we have not presented* *yet a distributed algorithm for more than two agents we use the preliminary local calculations for the following definitions and explanations.*

For two components i and j consider an arbitrary pair of local configurations $(C_{\nu_i}, C_{\nu_j}) \in \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}}) \times \mathcal{C}_j(\mathcal{O}^j_{\theta_{com}})$.

We say that C_{ν_i} and C_{ν_j} are *local consistent* if there exists an interpretation function $\psi_{\ell_{\nu_i\nu_j}} \in \Psi_{\nu_i\nu_j}$ s.t. $C_{\ell_{\nu_i\nu_j}} = (C_{\nu_i}, C_{\nu_j}, \psi_{\ell_{\nu_i\nu_j}})$ does not contain input border conditions that correspond with the border places of component i and component j that have no predecessors, that is there are no input border conditions that remain unassigned: $B^{ua}_{\ell_{\nu_i\nu_j}}(IN_{ij}) = \emptyset$ and $B^{ua}_{\ell_{\nu_i\nu_j}}(IN_{ji}) = \emptyset$.

Consider two arbitrary subsets of local preliminary configurations derived for component i and component j respectively e.g. $\mathcal{C}'_i(\mathcal{O}^i_{\theta_{com}}) \subseteq \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ and $\mathcal{C}'_j(\mathcal{O}^j_{\theta_{com}}) \subseteq \mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$.

We say that $\mathcal{C}'_i(\mathcal{O}^i_{\theta_{com}})$ and $\mathcal{C}'_j(\mathcal{O}^i_{\theta_{com}})$ are *local consistent* if:

1. $\forall C_{\nu'_i} \in \mathcal{C}'_i(\mathcal{O}^i_{\theta_{com}}) \Rightarrow \exists C_{\nu'_j} \in \mathcal{C}'_j(\mathcal{O}^j_{\theta_{com}})$ such that $C_{\nu'_i}$ and $C_{\nu'_j}$ are local consistent.
2. $\forall C_{\nu'_j} \in \mathcal{C}'_j(\mathcal{O}^j_{\theta_{com}}) \Rightarrow \exists C_{\nu'_i} \in \mathcal{C}'_i(\mathcal{O}^i_{\theta_{com}})$ such that $C_{\nu'_i}$ and $C_{\nu'_j}$ are local consistent.

The subsets of local configurations $\mathcal{C}'_1(\mathcal{O}^i_{\theta_{com}}), \ldots, \mathcal{C}'_{|J|}(\mathcal{O}^i_{\theta_{com}})$ are local consistent if for any two components that have a common border (i.e. $\forall i, j$ s.t. $\mathcal{P}_{ij} \neq \emptyset$) we have that $\mathcal{C}'_i(\mathcal{O}^i_{\theta_{com}})$ and $\mathcal{C}'_j(\mathcal{O}^j_{\theta_{com}})$ are local consistent.

Consider a $\mid J \mid$-tuple of local configurations

$$(C_{\nu'_1}, \ldots, C_{\nu'_{|J|}}) \in \mathcal{C}'_1(\mathcal{O}^1_{\theta_{com}}) \times \ldots \times \mathcal{C}'_{|J|}(\mathcal{O}^{|J|}_{\theta_{com}})$$

We say that the $\mid J \mid$-tuple of local configurations $(C_{\nu'_1}, \ldots, C_{\nu'_{|J|}})$ is *global consistent* if for any two components that have a common border ($\forall i, j \in J, \mathcal{P}_{ij} \neq \emptyset$) the local configuration C'_i and C'_j are local consistent.

The subsets of local configurations $\mathcal{C}'_1(\mathcal{O}^1_{\theta_{com}}), \ldots, \mathcal{C}'_{|J|}(\mathcal{O}^{|J|}_{\theta_{com}})$ are *global consistent* if $\forall i \in J$ and $\forall C_{\nu'_i} \in \mathcal{C}'_i(\mathcal{O}^i_{\theta_{com}})$ there exists a $\mid J \mid$-tuple of local configurations

$$(C_{\nu'_1}, \ldots, C_{\nu'_i}, \ldots, C_{\nu'_{|J|}}) \in \mathcal{C}'_1(\mathcal{O}^1_{\theta_{com}}) \times \ldots \times \mathcal{C}'_i(\mathcal{O}^i_{\theta_{com}}) \times \ldots \times \mathcal{C}'_{|J|}(\mathcal{O}^{|J|}_{\theta_{com}})$$

s.t. $(C_{\nu'_1}, \ldots, C_{\nu'_i}, \ldots, C_{\nu'_{|J|}})$ is global consistent.

We have that it does not hold in general that if the $\mid J \mid$-tuple of local configurations

$$C_J \in \mathcal{C}'_1(\mathcal{O}^1_{\theta_{com}}) \times \ldots \times \mathcal{C}'_i(\mathcal{O}^i_{\theta_{com}}) \times \ldots \times \mathcal{C}'_{|J|}(\mathcal{O}^{|J|}_{\theta_{com}})$$

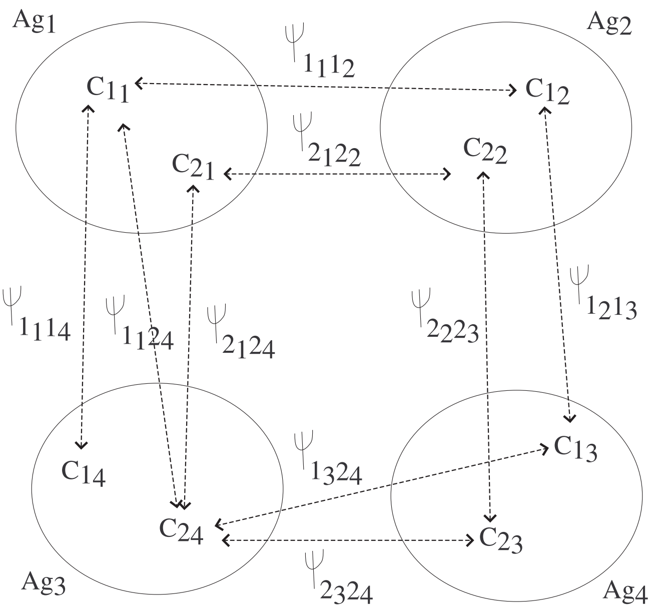


Figure 4.14:

is local consistent then C_J it is global consistent but if C_J is global consistent then C_J is local consistent ($C_J = (C_{\nu'_1}, \ldots, C_{\nu'_i}, \ldots, C_{\nu'_{|J|}})$).

To illustrate this consider in Fig. 4.14 the case of 4 components having the sub-sets of preliminary local configurations:

$$\mathcal{C}'_i(\mathcal{O}^i_{\theta_{com}}) = \{C_{1_i}, C_{2_i}\}, i = 1, 2, 3, 4.$$

We have that $\mathcal{C}'_1(\mathcal{O}^1_{\theta_{com}}), \mathcal{C}'_2(\mathcal{O}^2_{\theta_{com}}), \mathcal{C}'_3(\mathcal{O}^3_{\theta_{com}}), \mathcal{C}'_4(\mathcal{O}^4_{\theta_{com}})$ are local consistent but not global consistent. This is because there is not any $4 - tuple$ of local configurations that includes C_{1_4}. The only global consistent $4 - tuples$ are $(C_{1_1}, C_{1_2}, C_{1_3}, C_{2_4})$ and $(C_{2_1}, C_{2_2}, C_{2_3}, C_{2_4})$.

Hence if two agents Ag_i and Ag_j exchange for a pair of local configurations $(C_{\nu'_i}, C_{\nu'_j}) \in \mathcal{C}'_i(\mathcal{O}^i_{\theta_{com}}), \mathcal{C}'_j(\mathcal{O}^j_{\theta_{com}})$ only information about their local border-places without communicating the names (codes) of the local configurations in the neighboring components to whom $C_{\nu'_i}$ and $C_{\nu'_j}$ have been already checked consistent then the agents can achieve only the local consistency of their results.

If the goal of the distributed algorithm is to recover the results of a centralized agent then two local agents Ag_i and Ag_j should exchange for a pair of local configurations $(C_{\nu'_i}, C_{\nu'_j}) \in \mathcal{C}'_i(\mathcal{O}^i_{\theta_{com}}), \mathcal{C}'_j(\mathcal{O}^j_{\theta_{com}})$ beside the information about their local border-places also information about the names (codes) of the local configurations in the neighbouring components to whom $C_{\nu'_i}$ and $C_{\nu'_j}$ have been already found consistent. In this way Ag_i receives codes of local configurations from components to whom it has no common border but this is natural since otherwise the local agents cannot achieve in a distributed way the global consistency of their results.

Notice that it does not hold in general that if a $\mid J \mid$-tuple of local configurations:

$$(C_{\nu'_1}, \ldots, C_{\nu'_{|J|}}) \in \mathcal{C}'_1(\mathcal{O}^1_{\theta_{com}}) \times \ldots \times \mathcal{C}'_{|J|}(\mathcal{O}^{|J|}_{\theta_{com}})$$

is global consistent then there exists a global configuration $C_\nu \in \mathcal{C}(\mathcal{O}_{\theta_{com}})$ such that $C_{\nu'_1}, \ldots, C_{\nu'_{|J|}}$ are the maximal sub-nets of C_ν that include conditions and events that correspond with $\mathcal{N}_1, \ldots, \mathcal{N}_{|J|}$.

This is because of the circular dependencies (cycles) in the overall PN model that cover more than two components and may be intuitively understood as follows. A $\mid J \mid$-tuple of local configurations $(C_{\nu'_1}, \ldots, C_{\nu'_{|J|}})$ is global consistent if for any two components $i, j \in J$ the local configurations $C_{\nu'_i}, C_{\nu'_j}$ are local consistent that is there exists an interpretation function $\psi_{\ell_{\nu'_i\nu'_j}}$ of the border-conditions of component i and j s.t. $C_{\ell_{\nu'_i\nu'_j}} = (C_{\nu'_i}, C_{\nu'_j}, \psi_{\ell_{\nu'_i\nu'_j}})$ does not have input border conditions without successors and $C_{\ell_{\nu'_i\nu'_j}}$ is acyclic.

Thus by exchanging only information about their local border-conditions the local agents cannot detect the circular dependencies between the local traces that are contained in the $\mid J \mid$-tuple of local configurations $(C_{\nu'_1}, \ldots, C_{\nu'_{|J|}})$ that is found global consistent.

To illustrate this consider the case of a plant comprising 4 components as illustrated in Fig. 4.15.

We have that:

1. p_1, p_2 are output places of $\mathcal{N}_1$ and input places of $\mathcal{N}_2$;
2. p_3, p_4 are output places of $\mathcal{N}_2$ and input places of $\mathcal{N}_1$;
3. p_5 is output place of $\mathcal{N}_4$ and input place of $\mathcal{N}_1$;
4. p_6 is output place of $\mathcal{N}_1$ and input place of $\mathcal{N}_4$;
5. p_7 is output place of $\mathcal{N}_2$ and input place of $\mathcal{N}_4$;
6. p_8 is output place of $\mathcal{N}_3$ and input place of $\mathcal{N}_4$;
7. the dotted lines represent the unobservable oriented paths that contain transition of more than one component:
    (a) $\wp_{uo} := p_9 \ldots p_3 \ldots p_2 \ldots p_4 \ldots p_6 \ldots p_{10}$ (unobservable path)

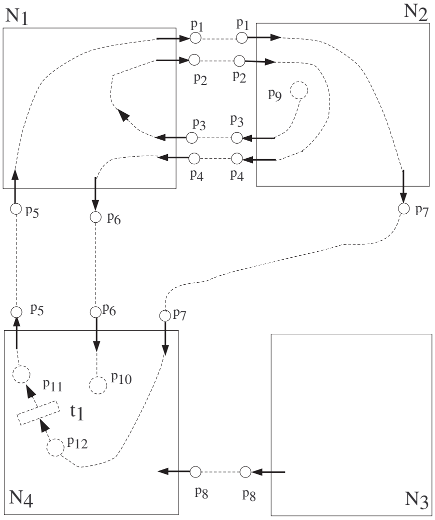


Figure 4.15:

(b) $\zeta_{uo} := p_{11} \dots p_5 \dots p_1 \dots p_7 \dots p_{12} t_1$ (unobservable circuit)

Consider that the local agents Ag_1, Ag_2, Ag_3 and Ag_4 have derived the sets of local preliminary configurations $\mathcal{C}_1^1(\mathcal{O}^1_{\theta_{com}}), \mathcal{C}_2^1(\mathcal{O}^2_{\theta_{com}}), \mathcal{C}_3^1(\mathcal{O}^3_{\theta_{com}})$ and $\mathcal{C}_4^1(\mathcal{O}^4_{\theta_{com}})$ respectively.

Moreover consider that two local agents exchange information only about the border conditions of their local border places and consider that first Ag_1 communicates with Ag_2.

We have that $K_{c_{12}} = 3$ since the oriented path $\wp$ alternates 3 times places of $\mathcal{N}_2$ and $\mathcal{N}_1$. Thus after 3 communications rounds and without intermediate communications with their neighbouring agents Ag_1 and Ag_2 stop their information exchange. For the sake of simplicity consider in the following that Ag_1 and Ag_2 achieve local consistency after one communication round. This allows us in the following to remove the indexes that refer to the communication round between two agents.

For a pair $(C_{\nu_1}, C_{\nu_2}) \in \mathcal{C}_1(\mathcal{O}^1_{\theta_{com}}) \times \mathcal{C}_2(\mathcal{O}^2_{\theta_{com}})$ and an interpretation function $\psi_{\ell_{\nu_1\nu_2}} \in \Psi_{\nu_1\nu_2}$ we have that $C_{\ell_{\nu_1\nu_2}} = (C_{\nu_1}, C_{\nu_2}, \psi_{\nu_1\nu_2})$ is a sub-net of a global configuration if $C_{\ell_{\nu_1\nu_2}}$ is acyclic. This is trivial since every global configuration is acyclic thus any sub-net of a global configuration should be acyclic.

Consider in the following for simplicity the trivial extensions of $C_{\ell_{\nu_1\nu_2}}$ in component 1, respectively component 2 namely $C_{\nu_1(\ell_{\nu_1\nu_2})}$ and $C_{\nu_2(\ell_{\nu_1\nu_2})}$.

Then consider that Ag_4 communicates with Ag_2. For a local configuration

$C_{\nu_4} \in \mathcal{C}_4(\mathcal{O}^4_{\theta_{com}})$ let an interpretation function $\psi_{\ell_{\nu_4 \ell_{\nu_1 \nu_2}}} \in \Psi_{\nu_4 \ell_{\nu_1 \nu_2}}$ be such that $C_{\ell_{\nu_4 \ell_{\nu_1 \nu_2}}}$ is acyclic.

Similarly consider the trivial extensions of $C_{\ell_{\nu_4 \ell_{\nu_1 \nu_2}}}$ in component 4 namely $C_{\nu_4(\ell_{\nu_4 \ell_{\nu_1 \nu_2}})}$.

Then consider that Ag_1 communicates with Ag_4 and for the pair of local configurations $(C_{\nu_1(\ell_{\nu_1 \nu_2})}, C_{\nu_4(\ell_{\nu_4 \ell_{\nu_1 \nu_2}})})$ and the interpretation function $\phi_{\ell_{\nu_1(\ell_{\nu_1 \nu_2})\nu_4(\ell_{\nu_4 \ell_{\nu_1 \nu_2}})}} \in \Psi_{\nu_1(\ell_{\nu_1 \nu_2})\nu_4(\ell_{\nu_4 \ell_{\nu_1 \nu_2}})}$ consider the configuration $C_{\ell_{\nu_1(\ell_{\nu_1 \nu_2})\nu_4(\ell_{\nu_4 \ell_{\nu_1 \nu_2}})}}$.

$C_{\ell_{\nu_1(\ell_{\nu_1 \nu_2})\nu_4(\ell_{\nu_4 \ell_{\nu_1 \nu_2}})}}$ may not be acyclic. This is because no any agent has knowledge about all the border places that are contained in the cycle if they communicate only information about their local border, i.e.

- Ag_1 knows $\psi_{\ell_{\nu_1 \nu_2}}$, $\psi_{\ell_{\nu_1(\ell_{\nu_1 \nu_2})\nu_4(\ell_{\nu_4 \ell_{\nu_1 \nu_2}})}}$, and $\wp_{uo_1} = p_5 \dots p_1$
- Ag_2 knows $\psi_{\ell_{\nu_1 \nu_2}}$, $\psi_{\ell_{\nu_4 \ell_{\nu_1 \nu_2}}}$, and $\wp_{uo_2} = p_1 \dots p_7$
- Ag_4 knows $\psi_{\ell_{\nu_4 \ell_{\nu_1 \nu_2}}}$, $\phi_{\ell_{\nu_1(\ell_{\nu_1 \nu_2})\nu_4(\ell_{\nu_4 \ell_{\nu_1 \nu_2}})}}$, and $\wp_{uo_4} = p_7 \dots p_{12} t_1 p_{11}$

where $\wp_{uo_1}, \wp_{uo_2}$ and respectively $\wp_{uo_4}$ denote the part of the circuit ζ_{uo} that comprises places and transitions in $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_4$ respectively.

Thus Ag_1 must send to Ag_4 also $\psi_{\ell_{\nu_1 \nu_2}}(p_1)$ while Ag_4 must send to Ag_1 also $\psi_{\ell_{\nu_4 \ell_{\nu_1 \nu_2}}}(p_7)$ where $\psi_{\ell_{\nu_1 \nu_2}}(p_1)$ and $\psi_{\ell_{\nu_4 \ell_{\nu_1 \nu_2}}}(p_7)$ are the interpretation of the border places in the local configurations that correspond with the places p_1 and p_7 respectively.

It means that to achieve the global consistency of the local results all the agents must exchange *all the information that they gathered about the border places that correspond with places of the circuit ζ_{uo} in the global model.*

But this has the inconvenience that the agents exchange information about border-conditions that correspond with places that are not included in the plant description that they know.

We have the following result:

Proposition 17. *Consider for each component $i \in J$ the local preliminary configurations $\mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$ derived by Ag_i for the local observation $\mathcal{O}^i_{com}$ and consider that for each component there exists a non-empty subset of local configurations $\mathcal{C}'_i(\mathcal{O}^i_{\theta_{com}}) \subseteq \mathcal{C}'_i(\mathcal{O}^i_{\theta_{com}})$ ($i \in J$) such that $\mathcal{C}'_1(\mathcal{O}^1_{\theta_{com}}), \dots, \mathcal{C}'_{|J|}(\mathcal{O}^{|J|}_{\theta_{com}})$ are global consistent. If the plant description is such that any unobservable circuit ζ_{uo} in $\mathcal{N}$ contains transitions of at most two components and if a $\mid J \mid$-tuple:*

$$(C_{\nu'_1}, \dots, C_{\nu'_{|J|}}) \in \mathcal{C}'_1(\mathcal{O}^1_{\theta_{com}}) \times \dots \times \mathcal{C}'_{|J|}(\mathcal{O}^{|J|}_{\theta_{com}})$$

is global consistent then there exists a global configuration $C_\nu \in \mathcal{C}(\mathcal{O}_{\theta_{com}})$ s.t. $C_{\nu'_1}, \dots, \dots, C_{\nu'_{|J|}}$ are the maximal sub-nets of C_ν that include conditions and events that correspond with $\mathcal{N}_1, \dots, \mathcal{N}_{|J|}$.

Proof. We have that $(C_{\nu'_1}, \ldots, C_{\nu'_{|J|}})$ is global consistent. Thus $\forall i, j \in J$, there exists an interpretation function $\psi_{\ell_{\nu_i \nu_j}}$ for the local configuration $C_{\nu'_i}$ and $C_{\nu'_j}$ s.t. for $C_{\ell_{\nu'_i \nu'_j}} = (C_{\nu'_i}, C_{\nu'_j}, \psi_{\ell_{\nu'_i \nu'_j}})$ we have that:

1. there are no input border conditions that correspond with the border places of comp. i and comp. j that have no predecessors in $C_{\ell_{\nu'_i \nu'_j}}$
2. $C_{\ell_{\nu'_i \nu'_j}}$ is acyclic

Abusing notation let $\psi_{\ell'_J}$ be the set of all the border interpretation functions of the local configurations:

$$\psi_{\ell'_J} = \left\{ \psi_{\ell_{\nu'_i \nu'_j}} \mid i, j \in J \text{ and } \psi_{\ell_{\nu'_i \nu'_j}} \in \Psi_{\nu'_i \nu'_j} \right\}$$

and denote by $C_{\ell'_J}$ the configuration obtained by merging the local configurations according with the $\psi_{\ell'_J}$.

$$C_{\ell'_J} = (C_{\nu'_1}, \ldots, C_{\nu'_{|J|}}, \psi_{\ell'_J})$$

Since the plant description is such that any unobservable circuit contains transitions of at most two components we have that $C_{\ell'_J}$ is acyclic. Then $C_{\ell'_J}$ has not input-border places that have no predecessors. Hence $C_{\ell'_J}$ is a global configuration. □

Thus if the overall PN model is such that any unobservable circuit ζ_{uo} in $\mathcal{N}$ comprises places and transitions of at most two components the local agents can derive global configurations by deriving global consistent pairs of local configurations.

Assumption 6. *Unless otherwise stated we assume in the remaining of this section that the overall PN model is such that any unobservable circuit ζ_{uo} in $\mathcal{N}$ comprises places and transitions of at most two components.*

In the following we present a distributed algorithm that guarantees that upon its completion the local agents recover the results of a centralized agent by exchanging limited information where the information that is exchanged between two local agents comprises only information about their local border places and the codes of how their local results where previously checked consistent with their neighbours.

By Theorem 4 we have that two neighboring agents Ag_i, Ag_j achieve local consistency after finitely communication rounds.

Without affecting the generality we assume in the following a communication protocol s.t. at the time θ_{com} when a communication session is allowed, two neighboring agents complete a number of communication rounds until they have become local consistent.

Moreover we require that the information exchange is *fair* [FBHJ05] i.e. any local agent is disallowed to communicate infinitely often with some neighbors ignoring to communicate with some other neighboring agents.

The distributed algorithm presented bellow for a local agent $Ag_i, i \in J$ comprises two parts. First the local agents derive their preliminary results in a similar way as presented for the case of two agents.

Denote by $Neighbour(i)$ the set of neighbouring components of component i. Then denote by $\mathcal{MSG}_{i \to j}$ the message that must be sent by Ag_i to its neighbour Ag_j where the information that is sent comprises information about their common border conditions for all the configurations that Ag_i has derived as well as the codes of the local configurations in the neighbouring components to whom the local configurations of component i have been already checked consistent.

Denote by $\mathcal{MSG}_i$ the set of messages prepared by Ag_i for all its neighbours:

$$\mathcal{MSG}_{to\_send_i} = \{\mathcal{MSG}_{i \to j} \mid j \in Neighbour(i)\}$$

and then denote by $\mathcal{MSG}_{receive_i}$ the set of messages received by Ag_i that are not yet processed:

$$\mathcal{MSG}_{received_i} = \{\mathcal{MSG}_{j \to i} \mid j \in Neighbour(i)\}$$

If for a neighbouring agent j, $\mathcal{MSG}_{i \to j} \neq \emptyset$ there is a non-empty set of input border conditions and output border conditions for at least a local configuration in component i that needs to be communicated to j then Ag_j can be chosen to check consistency. This is simply implemented running the distributed algorithm $DD\_Algo\_2(Ag_i, Ag_j)$ presented for the case of two agents (Algorithm 9).

The distributed algorithm $DD\_Algo\_2(Ag_i, Ag_j)$ is also run if Ag_i is chosen by its neighbour Ag_j to check consistency of their local results. Notice that once initiated the consistency check of the local results of Ag_i and Ag_j can not be interrupted by the arriving of a message sent by some other neigbouring agents of Ag_i and Ag_j.

If the consistency check with a neighbouring agent Ag_i is terminated then the set of messages that have to be sent is updated with the information that have become newly available to Ag_i after the communication with Ag_j that is new output border conditions and new codes for the local configurations.

If $\mathcal{MSG}_{to\_send_i} = \emptyset$ and $\mathcal{MSG}_{receive_i} = \emptyset$ then Ag_i emits the message that it is in a *local stop*. This message is propagated to all the agents and if a local agent receives the acknowledgment that all the local agents are in a local stop then a message is broadcast to all the agents indicating that a fix-point was achieved.

At the time when the distributed computation achieves a fix-point the local agents have derived the set of global configurations

$$\mathcal{C}_J(\mathcal{O}_{\theta_{com}}) = \{C_\nu = (C_{\nu_1}, \ldots, C_{\nu_i}, \ldots, C_{\nu_{|J|}}) \mid \nu_i \in \mathcal{V}_i, i \in J\}$$

Notice that a local agent i knows only its local configurations and codes for the configurations in all the other components:

$$\mathcal{C}_i(\mathcal{O}_{\theta_{com}}) = \{C_\nu = (\nu_1, \ldots, C_{\nu_i}, \ldots, \nu_{|J|}) \mid \nu_j \in \mathcal{V}_j, j \in J\}$$

In the second part of the algorithm the local agents discard the $\mid J \mid$-tuples that contains input-border conditions that have no predecessors. For any configuration $C_\nu \in \mathcal{C}(\mathcal{O}_{\theta_{com}})$ if C_{ν_i} has an input border condition that is not assigned Ag_i sends a message to all its neighboirs to acknowledge that any global pair that contains ν_i should be deleted. If Ag_i receives a message from a neighbour that all the $\mid J \mid$-tuples that contain a certain code should be deleted, then Ag_i deletes the corresponding $\mid J \mid$-tuples and also propagates further this information.

When all the $\mid J \mid$-tuples are deleted the agents obtain the set of global configurations $\mathcal{C}^{gcon}(\mathcal{O}_{\theta_{com}})$.

Proposition 18. *The distributed algorithm Algorithm 10 implemented to a set of agents J terminates and the set of global configurations $\mathcal{C}^{gcon}(\mathcal{O}_{\theta_{com}})$ that is derived by the local agents when $Algorithm 10$ ends is the set of configurations $\mathcal{C}(\mathcal{O}_{\theta_{com}})$ derived by a centralized agent that has the overall plant knowledge and observation.*

Proof. First we prove that Algorithm 10 terminates.

For $i = 1, \ldots, \mid J \mid$ denote by $M_{\nu_i^1}(IN_i)$ the marking of the input places $\mathcal{P}_{IN_i}$ in $C_{\nu_i^1}$ that correspond with the set of input border conditions considered by Ag_i for a preliminary local configuration $C_{\nu_i^1}$, i.e. $M_{\nu_i^1}(IN_i) = \phi(B_{C_{\nu_i^1}}(IN_i))$.

Denote by M_J the marking obtained from the sub-vector markings $M_{0_1}, \ldots, M_{0_{|J|}}, M_{\nu_1^1}(IN_1), \ldots, M_{\nu_{|J|}^1}(IN_{|J|})$:

$$M_J = M_{0_1} \uplus \ldots \uplus M_{0_{|J|}} \uplus M_{\nu_1^1}(IN_1) \uplus \ldots \uplus M_{\nu_{|J|}^1}(IN_{|J|})$$

where $\uplus$ denotes the union with addition of two multi-sets and a marking is considered a multi-set of tokens.

Consider $\langle \mathcal{N}, M_J \rangle$ that is the overall plant model $\mathcal{N}$ with the initial marking M_J. By Proposition 13 we have that M_J is finite and then by assumption $viii)$ in the setting we have that $\langle \mathcal{N}, M_J \rangle$ is bounded w.r.t. the unobservable evolution.

Denote by $\mathcal{U}_\mathcal{N}(M_J)$ the net-unfolding of $\langle \mathcal{N}, M_J \rangle$. Consider the (finite) sequence of observed events $\mathcal{O}_n$ and denote by $\mathcal{U}_\mathcal{N}(M_J, \mathcal{O}_n)$ the prefix of $\mathcal{U}_\mathcal{N}(M_J)$ that correspond with the observation. Denote by $\mathcal{C}(M_J, \mathcal{O}_n)$ the set of configurations that obey the observation $\mathcal{O}_n$.

Algorithm 10 DD_Algo (Ag_i considered)

```
Require: $\langle \mathcal{N}_i; M_{0_i} \rangle; \mathcal{O}^i_{\theta_{com}}$
Ensure: $\mathcal{C}^{gcon}_i(\mathcal{O}^i_{\theta_{com}})$
while fix_point=false do
    if $\mathcal{MSG}_{to\_send_i} = \emptyset$ and $\mathcal{MSG}_{receive_i} = \emptyset$ then
        send $\mathcal{MSG}_{i \to j} = \{$ local_stop $i\}$ to the neighbouring agents
    else if $\mathcal{MSG}_i \neq \emptyset$ then
        choose a neighbour $Ag_j$ s.t. $\mathcal{MSG}_{i \to j} \neq \emptyset$ or $\mathcal{MSG}_{j \to i} \neq \emptyset$
        run $DD\_Algo\_2$ for $Ag_i, Ag_j$
        update $\mathcal{MSG}_{to\_send_i}$
    end if
    if local_stop for all $j \in J$ then
        send to all neighbours $\mathcal{MSG}_{i \to j} = \{$ fix_point $\}$
         fix_point=true
    end if
end while
$\mathcal{C}_J(\mathcal{O}_{\theta_{com}}) = \{C_\nu = (\nu_1, \ldots, C_{\nu_i}, \ldots, \nu_{|J|}) \mid \nu_j \in \mathcal{V}_j, j \in J\}$
for all $C_\nu \in \mathcal{C}(\mathcal{O}_{\theta_{com}})$ do
    if $B^{ua}_{C_\nu}(IN_i) \neq \emptyset$ then
        delete $C_\nu$ from $\mathcal{C}(\mathcal{O}_{\theta_{com}})$
        send to all neighbours $\mathcal{MSG}_{i \to j} = \{C_\nu$ not consistent $\}$
    end if
    if $\mathcal{MSG}_{j \to i} = \{C'_\nu$ not consistent $\}$ then
        delete $C'_\nu$ from $\mathcal{C}(\mathcal{O}_{\theta_{com}})$
        send to all the neighbours other than j: $\mathcal{MSG}_{i \to q} = \{C'_\nu$ not consistent $\}$
    end if
end for
$\mathcal{C}^{gcon}(\mathcal{O}_{\theta_{com}}) = \mathcal{C}_J(\mathcal{O}_{\theta_{com}})$
```

All unobservable cycles in $\langle \mathcal{N}, M_J \rangle$ that repeats the marking are either executed only once or are not at all executed.

Since we have a finite number of observed events we have that any configuration $C(M_J, \mathcal{O}_n) \in \mathcal{C}(M_J, \mathcal{O}_n)$ contains a finite number of condition nodes and a finite number of event nodes.

Denote by $C_{\ell_J} = (C_{\ell_i}, \ldots, C_{\ell_{|J|}}, \psi_{\ell_J})$ the global configuration that is obtained by merging the local configurations.

We have that C_{ℓ_J} is a configuration in $\mathcal{U}_{\mathcal{N}}(M_J)$. Since any configuration $C(M_J, \mathcal{O}_n) \in \mathcal{C}(M_J, \mathcal{O}_n)$ contains a finite number of condition nodes and a finite number of event nodes then the local agents cannot communicate infinitely often.

Notice that any unobservable circuit contains transitions of at most two components. Thus the cycles that repeat the marking are detected as in Algorithm 9 for two agents.

We prove that $\mathcal{C}^{gcon}(\mathcal{O}_{\theta_{com}}) = \mathcal{C}(\mathcal{O}_{\theta_{com}})$ by proving that $\mathcal{C}^{gcon}(\mathcal{O}_{\theta_{com}}) \subseteq$

$\mathcal{C}(\mathcal{O}_{\theta_{com}})$ and then that $\mathcal{C}^{gcon}(\mathcal{O}_{\theta_{com}}) \supseteq \mathcal{C}(\mathcal{O}_{\theta_{com}})$

($\Rightarrow$) Consider a global configuration $C_{\ell_J} \in \mathcal{C}^{gcon}(\mathcal{O}_{\theta_{com}})$ where for adequate interpretation functions:

$$C_{\ell_J} = (C_{\ell_1}, \ldots, C_{\ell_{|J|}})$$

Since all the input border-conditions in C_{ℓ_J} have predecessors then C_{ℓ_J} is a configuration in $\mathcal{U}_{\mathcal{N}}(M_0)$.

($\Rightarrow$) The proof is straightforward.

□

Theorem 5. *Consider a distributed description of the plant and an arbitrary distributed observation $\mathcal{O}_{\theta_{com}} = \otimes^{gc}_{i \in J} \mathcal{O}^i_{\theta_{com}}$. The algorithm* DD_Algo *terminates in finite time and the local diagnosis $\mathcal{LD}^{gcon}_i$ ($i \in J$) derived by each local d-agent Ag_i ($i \in J$) is the diagnosis a centralized agent would have obtained for component i having the entire plant observation $\mathcal{O}_{\theta_{com}}$ and the knowledge of the overall plant.*

$$\forall i \in J, \quad \mathcal{D}_i(\mathcal{O}^i_{\theta_{com}}) = \mathcal{LD}^{gcon}_i(\mathcal{O}^i_{\theta_{com}})$$

Proof. The proof results straightforward from Proposition 18 by projecting two equal sets onto the set of fault events. □

Remark 16. *When the agents complete the communication protocol at the time θ_{com} they continue to locally monitor the components until a new communication session is allowed. The only difference is that instead of a single initial marking M_{0_i} as was the case for deriving the preliminary calculation before the first communication session, a local agent will have for a component a set of initial markings, that are the local markings that were found consistent after the completion of the communication protocol at the last communication session, i.e. instead $M_{0_1}, \ldots, M_{0_{|J|}}$ the local agents will consider the set of local markings:*

$$\mathcal{M}_{0_{\theta_{com}}} = \left\{(M_{0_{\nu_1}}, \ldots, M_{0_{\nu_{|J|}}}) \mid M_{0_{\nu_i}} = mark(C_{\nu_i}) \wedge C_\nu \in \mathcal{C}(\mathcal{O}_{\theta_{com}}) \wedge i \in J\right\}$$

where $C_\nu = (C_{\nu_1}, \ldots, C_{\nu_{|J|}})$.

Consider in the following the case when the plant comprises a large number of components. The amount of information that is necessary to be exchanged between the local agents regarding indexes of configurations in all the components may be considerably large. In what follows we show that under normal assumption this information can be reduced.

Consider a component $i \in J$ and denote by J_i the subset of components $J_i \subseteq J$ such that $j \in J_i$ implies that there exists an unobservable oriented path that starts in a place in component i and ends in a place of component j. A $\mid J_1 \mid$-tuple $(C_{\nu_{j_1(i)}}, \ldots, C_{\nu_{j_{|J_1|}(i)}})$ is pairwise consistent if for any two components in J_1 that have a common border, $C_{\nu_{j_u(i)}}$ and $C_{\nu_{j_v(i)}}$ are local consistent.

Proposition 19. *Consider a* $\mid J \mid$*-tuple of local configurations* $(C_{\nu_1}, \ldots, C_{\nu_{|J|}})$*. We have that* $(C_{\nu_1}, \ldots, C_{\nu_{|J|}})$ *is global consistent iff* $\forall i \in J$*, the* $\mid J_1 \mid$*-tuple* $(C_{\nu_{j_1(i)}}, \ldots, C_{\nu_{j_{|J_1|}(i)}})$ *is pairwise consistent*

Proof. Trivial. □

Based on Proposition 19 we have that a local agent i needs to know only indexes for configurations of the subset of components J_i. Assume that every local agent $i \in J$ knows for each of its neighbours $j \in J$ ($\mathcal{P}_{ij} \neq \emptyset$) the sub-set J_j of components, then the information exchanged between Ag_i to Ag_j includes codes only for configurations of the components that belong to J_i and J_j. Thus instead of sending the codes of J components two local agents exchange codes only for the components in the sub-set $J_{ij} = J_i \cap J_j$.

Chapter 5

Diagnosis for Time PNs

5.1 Introduction

In this chapter we extend the results of Chapter 4 by assuming that the PN models include information on the time delay between satisfying an enabling condition (tokens entering a place) and executing a transition. The PN models that we consider in the following consider the time as a quantifiable and continuous parameter whereas in an untimed PN model the time is taken into account only via the partial order relation between the transitions that are executed in the plant.

Two main time extensions of Petri Nets are Time Petri Nets [Mer74] and Timed Petri Nets [Ram74]. The difference between the two is that in Time Petri Nets a transition can be fired after a delay within a given time-interval. The execution takes no time to complete. In Timed Petri Nets a transition fires as soon as possible (without delay) but its execution requires a certain amount of time to complete. In this thesis we have chosen Time Petri Nets for modeling our systems since this formalism is convenient for expressing most of the temporal constraints regarding the execution and the duration of the events.

Thus in a Time Petri Net each transition $t \in \mathcal{T}$ has attached to it a time interval, called the static interval $I^s(t) = [L_t^s, U_t^s]$ that represents the set of all the possible delays between enabling and execution associated to transition t. Times L_t^s and U_t^s are relative to the time θ_t^{en} when transition t becomes enabled and U_t^s may be infinite. If transition t becomes enabled at the time θ_t^{en} then t cannot fire at a time before $\theta_t^{en} + L_t^s$ and is forced to be executed at the time $\theta_t^{en} + U_t^s$ unless it is disabled before by the firing of another transition. The execution of a transition takes no time to complete. A state in a TPN at the time θ is represented by the marking (the untimed state) and by the firing domains of the enabled transitions, i.e. the time intervals in which the enabled transitions can be executed.

As for untimed PNs all the interesting problems for the analysis of TPN models can be reduced to reachability analysis. Since a transition in a TPN can fire at any time in its firing domain, TPN models have in general infinite state spaces because a state may have an infinite number of successor states. Methods based on grouping states that are equivalent under a certain equivalence relation into so called *state classes* were proposed in [BM83], [YR98], [BV02] where it was shown that for bounded PNs the state class graph is finite. Thus the potentially infinite state spaces of a TPN can be finitely represented and thus the analysis of TPN models can be reduced to a decidable problem.

The goal of this chapter is to design an on-line model-based algorithm that derives the plant diagnosis at time θ based on the known plant model and on the observation received up to time θ. We assume a perfect knowledge of the initial state and consider that the model is correct and there are no delays in receiving the plant observation.

The setting that we consider assumes that the plant observation is given via a subset of transitions (observable transitions) whose occurrence is reported. Moreover we consider that the execution time of an observed event is also reported and is measured with perfect accuracy w.r.t. a global clock. The faults that should be detected are modeled by a subset of the unobservable (silent) transitions.

In the first part of this chapter we present in Section 5.4 the centralized analysis of TPN models under partial observation. The centralized analysis of the plant based on state classes [BM83] is presented in Section 5.4.1. In order to represent more accurately the plant behavior we consider the plant analysis based on the atomic state class graph [BV02], [YR98]. The atomic state class graph is a refinement of the linear state class graph that is based on the observation that a state in a linear state class may contain states that do not have successors in all the successor state classes.

The centralized on-line monitoring algorithm that we design in this section can be briefly described as follows. When the process starts we derive paths in the atomic state class graph up to the first observable event. If either the observable event is not observed in the output of the plant or if it is executed sooner than allowed by the characteristic system of the final atomic state class, then the path is deleted. Otherwise an equality relation is added to the characteristic system to express the fact that the observable event occurred at the time given by the received observation. Adding equality relation destroys in general the atomicity property of the state classes and thus it must be restored by splitting up existing state classes in subintervals (subdomains).

Since the TPN analysis based on state-classes becomes computationally unfeasible for models of reasonable size because of the state space explosion (due to the interleaving of the unobservable concurrent events) we propose in Section 5.5 a diagnosis algorithm based on time-configurations (time-processes [AL97]). A time-configuration is an untimed configuration with a valuation of the execution time for its events. A time-configuration is valid if there is a time trace in the original TPN that can be obtained from a linearization of the events of the configuration where the occurrence time of the transitions in the trace are identical with the valuation of their images in the time-configuration. To check whether a time-configuration is valid or not requires to solve a $(max, +) - linear$ system of inequalities called the characteristic system of the configuration.

Since the number of valid time-configurations is uncountable we introduce the time-interval configurations to finitely represent the set of all possible valid time-configurations. The idea is simple. The set of all solutions of the characteristic system of a configuration (the set of all valid times) is represented as a cover of subsets of solutions such that each sub-set of solutions has the time independence property for the concurrent events in the configuration. The time independence property of a subset of solutions of the characteristic system of a configuration can be intuitively understood as follows: *given any set of concurrent events in the configuration and fixing the execution times of their predecessors, their executions times belong to a hyper-rectangle in high dimensional space*. The execution time-intervals for the events in the configuration are obtained from the smallest hyperbox (of dimension equal with number of events in the configuration) that includes a given subset of solutions of the characteristic system.

We present an efficient algorithm to derive such a partition of the solution set of the characteristic system of a configuration and show how the method can handle the addition of extra inequalities (constrains) that are due to the received observation.

Then as a preamble to Section 5.7 (were we design a distributed algorithm for TPN models) we present in Section 5.6 the (centralized) backward analysis of Time Petri Nets. The backward analysis is needed in the distributed algorithm in order to meet the requirement that a local agent derives a local preliminary diagnosis of a component before communicating with its neighbors, and hence without knowing how many tokens can enter the border places.

The distributed diagnosis algorithm of TPN models is designed in Section 5.7 considering the same setting and the same requirements that were considered in Section 4.3 for the distributed diagnosis of untimed PN models. The differences here are that only two components are considered, that the overall plant model is 1-safe free-choice Time Petri Net and that in order to ensure that the algorithm converges in a finite number of iterations, we must assume that any oriented path that starts in an input border-place and ends in an output border place of the same component includes at least one observable event. These assumptions are necessary to prove desirable properties of the distributed fault diagnosis. In particular in Section 5.7.5 we prove that the distributed algorithm that we design for TPN models is such that the local agents recover the diagnosis result that would have been derived by a centralized agent knowing the overall plant model and having the overall plant observation.

The distributed diagnosis algorithm for TPN models comprises the same procedures as presented for the untimed models namely a procedure for performing the preliminary local calculation (Section 5.7.2), a procedure for information exchange (Section 5.7.3), and a procedure for updating a local calculation to incorporate the received information (Section 5.7.4).

5.2 Time Petri Nets

A Time Petri Net (TPN) $\mathcal{N}^\theta = (\mathcal{P}, \mathcal{T}, F, I^s)$, consists of an (untimed) Petri Net $\mathcal{N} = (\mathcal{P}, \mathcal{T}, F)$ (called the untimed support of $\mathcal{N}^\theta$) and the static time interval function $I^s : \mathcal{T} \to \mathcal{I}(\mathbb{Q}^+)$, $I^s(t) = [L_t^s, U_t^s]$, $L_t^s, U_t^s \in \mathbb{Q}^+$, representing the set of all possible time delays associated to transition $t \in \mathcal{T}$.

In a TPN $\mathcal{N}^\theta$ we say that a transition t becomes enabled at the time θ_t^{en} (according to a global clock [WD00]) then the clock attached to t is started and the transition t can and must fire at some time $\theta_t \in [\theta_t^{en} + L_t^s, \theta_t^{en} + U_t^s]$, provided t did not become disabled because of the firing of another transition. Notice that t is forced to fire if it is still enabled at the time $\theta_t^{en} + U_t^s$.

The following definitions are borrowed from [BM83] with the difference that here the time is counted according to a global clock relative with the time the process starts that is assumed 0 for simplicity.

Definition 51. *A state at the time θ (according to a global clock) of a TPN $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ is a pair $S_\theta = (M, FI)$ where:*

i) M is a marking and

ii) FI is a firing interval function that associates an interval to each enabled transition in M ($FI : \mathcal{T} \to \mathcal{I}(\mathbb{Q}^+)$)

We write $(M, FI) \xrightarrow{\langle t,\theta_t \rangle} (M', FI')$ or simply $S \xrightarrow{\langle t,\theta_t \rangle} S'$ if $\theta_t \in \mathbb{Q}^+$ and:

1. $(M \geq Pre(\cdot, t) \wedge \theta_t \geq \theta_t^{en} + L_t^s) \wedge (\forall t' \in \mathcal{T}$ *s.t.* $M \geq Pre(\cdot, t') \Rightarrow \theta_t \leq \theta_{t'}^{en} + U_{t'}^s)$

2. $M' = M - Pre(\cdot, t) + Post(t, \cdot)$

3. $\forall t'' \in \mathcal{T}$ *s.t.* $M' \geq Pre(\cdot, t'')$ *we have:*

   (a) *if $t'' \neq t$ and $M \geq Pre(\cdot, t'')$ then*
   $FI(t'') = [\mathtt{max}(L_{t''}^s + \theta_{t''}^{en}, \theta_t), \theta_{t''}^{en} + U_{t''}^s]$

(b) else $\theta_{t''}^{en} = \theta_t$ *and* $FI(t'') = [\theta_{t''}^{en} + L_{t''}^{s}, \theta_{t''}^{en} + U_{t''}^{s}]$

Condition (1) above assures that a transition t that fires at the time θ_t is enabled by the marking $M \geq Pre(\cdot, t)$ and that it fires in its temporal interval $FI(t)$ unless it is disabled by another enabled transition that is executed sooner. Condition (2) represents the marking transformation rule while (3) gives the rule how the firing intervals are modified that is:

(3.a) if a transition t'' remains enabled after firing t then its lower limit remains unaffected if $\theta_t \leq \theta_{t''}^{en} + L_{en}^{s}$ while if $\theta_t \geq \theta_{t''}^{en} + L_{t''}^{s}$ then the earliest firing time becomes θ_t

(3.b) while for a newly enabled transition t'' its firing interval $FI(t'')$ is obtained by adding the time $\theta_{t''}^{en}$ transition t'' has become enabled ($\theta_t = \theta_{t''}^{en}$) to its static interval $(L_{t''}^{s}, U_{t''}^{s})$

A time trace τ^θ is represented by a sequence that alternates the time progress updates and the occurrence of some events. Thus starting from the initial state S_0 first the time interval (denoted $\epsilon, [0, \theta_{t_1})$) lasts until the time θ_{t_1} when the first transition t_1 fires. Notice that the occurrence of a transition takes no time to complete that is the tokens from the input places of a transition t_ι are removed and added to the output places of t_ι instantaneously at the time θ_{t_ι} when t_ι fires.

$$\tau^\theta = S_0 \xrightarrow{\epsilon,[0,\theta_{t_1})} S_1' \xrightarrow{\langle t_1,\theta_{t_1}\rangle} S_1'' \xrightarrow{\epsilon,(\theta_1,\theta_{t_2})} S_2' \xrightarrow{\langle t_2,\theta_{t_2}\rangle} S_2'' \ldots S_\upsilon' \xrightarrow{\langle t_\upsilon,\theta_{t_\upsilon}\rangle} S_\upsilon \tag{5.1}$$

Definition 52. *A time trace τ^θ in a TPN $\mathcal{N}^\theta$ is defined as follows:*

i) $\tau^\theta = S_0 \xrightarrow{\langle t_1,\theta_{t_1}\rangle} S_1 \xrightarrow{\langle t_2,\theta_{t_2}\rangle} \ldots \xrightarrow{\langle t_\upsilon,\theta_{t_\upsilon}\rangle} S_\upsilon$

ii) $\forall \iota = 0, \upsilon - 1, \exists \theta_{t_{\iota+1}}$ *s.t.* $S_\iota \xrightarrow{\langle t_{\iota+1},\theta_{t_{\iota+1}}\rangle} S_{\iota+1}$ *according to Definition 51.*

iii) $\tau = M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} \ldots \xrightarrow{t_\upsilon} M_\upsilon$ *is the untimed support of* τ^θ

Definition 53. *Denote $\xrightarrow{*}$ for the reflexive and transitive closure of $\longrightarrow$. The state graph of a TPN $\mathcal{N}^\theta$ is $SG = (\mathcal{S}, \xrightarrow{*}, S_0)$ where:*

i) $\mathcal{S} = \left\{ S \mid S_0 \xrightarrow{*} S \right\}$ *is the set of reachable states from the initial state* S_0

ii) $S_0 = (M_0, FI_0)$ *where* $FI_0(t) = I^s(t)$ *for all transitions t s.t.* $M_0 \geq Pre(\cdot, t)$ *otherwise $FI_0(t)$ is not defined*

Remark 17. *Notice that in Definition 51 and Definition 53 above the assumption made is that there is a start-up transition that fires only once at the time zero producing the tokens considered by the initial marking. If this is not suitable for the model* *under investigation then consider for each token in the initial marking a fictitious transition that can fire only once and its static interval represents the* date of birth *of the tokens in the initial marking [CJ05]. The time the process starts is translated to the time when all the fictitious transitions become enabled (one token is produced simultaneously in the input places of the fictitious transitions).*

Notation 1. In the following for a time-trace τ^θ we use the notation τ to denote its untimed support. For the initial state S_0 we use also the notation M_0^θ. Denote by $\mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0^\theta)$ the set of all possible time-traces that can be executed in $\langle \mathcal{N}^\theta, M_0^\theta \rangle$. We call $\mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0^\theta)$ the time-language of the TPN $\langle \mathcal{N}^\theta, M_0^\theta \rangle$.

Then denote by $\mathcal{L}_{\mathcal{N}^\theta}(M_0^\theta)$ the untimed support language of $\mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0^\theta)$ i.e.:

$$\mathcal{L}_{\mathcal{N}^\theta}(M_0^\theta) = \left\{ \tau \mid \exists \tau^\theta \in \mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0^\theta) \right\}$$

For a time trace τ^θ we use the simplified notation $\tau^\theta = \theta_{t_1} \dots \theta_{t_\upsilon}$ and recall that $\mathcal{L}_{\mathcal{N}}(M_0)$ is the untimed language derived for the untimed support PN $\langle \mathcal{N}, M_0 \rangle$ of the TPN $\langle \mathcal{N}^\theta, M_0^\theta \rangle$.

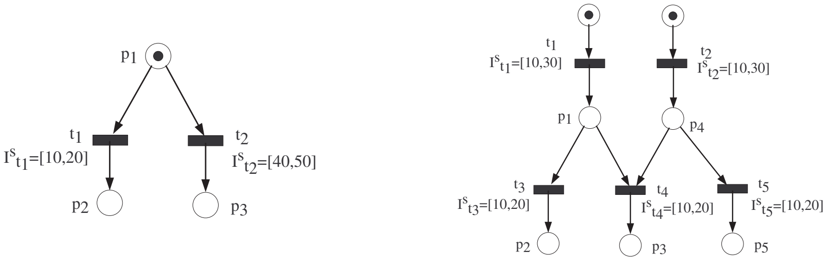


Figure 5.1:

Example 23. *Consider the TPN $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ displayed in Fig.5.1-left. The two transitions t_1 and t_2 are both enabled in the initial marking and the execution of either one disables the execution of the other one. Since $\theta^{en}_{t_1} = \theta^{en}_{t_2}$ and $U^s_{t_1} < L^s_{t_2}$ then it means that t_2 cannot be executed.*

For TPN $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ displayed in Fig.5.1-right, if a token arrives in p_1 then transition t_3 becomes enabled whereas transition t_4 becomes enabled when its input places p_1 and p_4 are both marked. Thus $\theta^{en}_{t_3} \leq \theta^{en}_{t_4}$ meaning that t_4 can be executed only if $L^s_{t_4} \leq U^s_{t_3}$. Moreover if t_2 fires more than 10 time units after t_1 ($\theta_{t_1} + 10 < \theta_{t_2}$) then transition t_4 cannot be executed because transition t_3 is forced to fire before.

To avoid trivialities we consider in the remaining of this thesis that given a TPN $\mathcal{N}^\theta$ for any two transitions $\forall t_1, t_2 \in \mathcal{T}$ we have that :

1. if $^\bullet t_1 = {}^\bullet t_2$ then $U^s_{t_1} = U^s_{t_2}$

2. if $^\bullet t_1 \subset {}^\bullet t_2$ then $U^s_{t_2} \leq U^s_{t_1}$

For this TPN model the following holds:

Proposition 20. *Given a TPN $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ we have that:*

$$\mathcal{L}_{\mathcal{N}^\theta}(M_0^\theta) \subseteq \mathcal{L}_{\mathcal{N}}(M_0)$$

that is the untimed support language of a TPN is included in the language of its untimed support PN. In other words the timing information reduces the set of possible untimed traces in a given PN model.

Then if $\langle \mathcal{N}, M_0 \rangle$ is a free-choice PN (Def.13) we have moreover that:

$$\mathcal{L}_{\mathcal{N}^\theta}(M_0^\theta) \equiv_{\Sigma_\mu} \mathcal{L}_{\mathcal{N}}(M_0) \tag{5.2}$$

that is for any untimed trace $\forall \tau \in \mathcal{L}_{\mathcal{N}}(M_0)$ derived in the untimed model there exists a timed-trace τ'^θ s.t. τ and τ' have the same Parikh vector ($\Sigma_\mu(\tau) = \Sigma_\mu(\tau')$).

Proof. The net is free-choice thus any transition that becomes enabled can be executed. This is true since $\forall t, t \in \mathcal{T}$ s.t. ${}^\bullet t = {}^\bullet t' \Rightarrow U_t^s = U_{t'}^s$. □

In words Proposition 20 says that the timing information eliminates some untimed traces because some transitions that are concurrent in the untimed model are forced to be executed in a certain order in the timed model. The timing constraints of the time model may also eliminate some choices that would be possible without timing constraints but are made impossible by timing constraints.

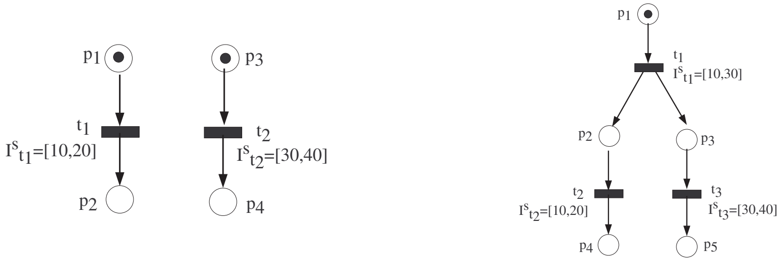


Figure 5.2:

Example 24. *Consider the TPN $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ displayed in Fig.5.2-left. The two transitions t_1 and t_2 are enabled in the initial marking and in the untimed PN $\langle \mathcal{N}, M \rangle$, t_1 and t_2 are concurrent thus they can be executed in any order, i.e. $\tau_1 = t_1 t_2$ and $\tau_2 = t_2 t_1$ are both legal. However this is not true in $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ since there is no legal time-trace that has the untimed support $\tau_2 = t_2 t_1$ since t_2 cannot be executed before t_1. This is because both t_1 and t_2 become enabled at the very same time while the earliest time when transition t_2 can be executed ($\theta_2^{en} + L_{t_2}^s$) is greater than the latest time time when transition t_1 must be executed ($\theta_1^{en} + U_{t_1}^s$). Thus $\theta_{t_1} < \theta_{t_2}$. Similarly in Fig.5.2-right we have $\theta_{t_2} < \theta_{t_3}$.*

Consider a TPN $\mathcal{N}^\theta$ and two initial markings M_0^θ and $M_0'^\theta$ s.t. $M_0^\theta \leq M_0'^\theta$ where all the tokens that appear in M_0^θ have the same date of birth in $M_0'^\theta$ as in M_0^θ.

Let $\mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0^\theta)$ and $\mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0'^\theta)$ be the timed languages of the TPN $\langle \mathcal{N}^\theta, M_0^\theta \rangle$, respectively $\langle \mathcal{N}^\theta, M_0'^\theta \rangle$. Then the following holds:

Proposition 21. *If $\mathcal{N}$ is free-choice and $M_0^\theta \leq M_0'^\theta$ then we have that:*

$$\mathcal{L}_{\mathcal{N}^\theta}(M_0^\theta)_{/\equiv_{\Sigma_\nu}} \sqsubseteq \mathcal{L}_{\mathcal{N}^\theta}(M_0'^\theta)_{/\equiv_{\Sigma_\nu}} \tag{5.3}$$

that is:

1. $\forall \tau^\theta \in \mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0^\theta) \Rightarrow \exists \tau'^\theta \in \mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0'^\theta)$ *s.t.* $\Sigma_\mu(\tau) \subseteq \Sigma_\mu(\tau')$
2. $\forall \tau'^\theta \in \mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0'^\theta) \Rightarrow \exists \tau^\theta \in \mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0'^\theta)$ *s.t.* $\Sigma_\mu(\tau') \supseteq \Sigma_\mu(\tau)$

Proof. To prove this it is enough to examine what effect an extra token can have in a free-choice net. We have that the presence of a token can not disable the execution of a transition. The only effects are the firing of some extra transitions or to speed up the execution of a transition. □

Unfortunately for a general TPN we do not have any monotonicity property because of the forcing of a transition to be executed when the time reaches its upper limit of its firing time interval.

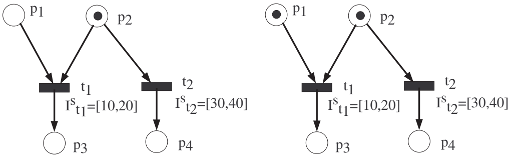


Figure 5.3:

Example 25. *Consider in Fig.5.3 a TPN with two initial markings e.g. $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ (Fig.5.3-left) and $\langle \mathcal{N}^\theta, M_0'^\theta \rangle$ (Fig.5.3-right). As one can see in $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ only t_2 can be executed whereas in $\langle \mathcal{N}^\theta, M_0'^\theta \rangle$ only t_1 is possible.*

Given a global time ξ denote by $\mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0^\theta, \xi)$ the set of all time traces that can be executed up to the time ξ that is:

$$\mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0^\theta, \xi) = \Big\{ \tau^\theta_\xi = \theta_{t_1} \ldots \theta_{t_\nu} \mid \exists \tau^\theta \in \mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0) \text{ s.t. } \tau^\theta = \theta_{t_1} \ldots \theta_{t_\nu} \theta_{t_{\nu+1}} \ldots \\ \wedge\ \theta_{t_\nu} \leq \xi \ \wedge\ \theta_{t_{\nu+1}} > \xi \Big\}$$

Remark 18. *In a rigorous notation we have that:*

$$\tau^\theta_\xi := S_0 \xrightarrow{[0,\theta_{t_1})} S_1' \xrightarrow{\theta_{t_1}} S_1'' \xrightarrow{(\theta_{t_1},\theta_{t_2})} \ldots \xrightarrow{(\theta_{t_{\nu-1}},\theta_{t_\nu})} S_\nu' \xrightarrow{\theta_{t_\upsilon}} S_\nu'' \xrightarrow{(\theta_{t_\upsilon},\xi]} S_\xi \tag{5.4}$$

where it was assumed that $\theta_{t_\nu} < \xi$ otherwise $\theta_{t_\nu} = \xi$, $S_\xi = S''_\nu$ and $\xrightarrow{(\theta_{t_\nu},\xi]}$ should be removed.

5.3 Diagnosis of TPN - general setting

We consider the following plant description:

1. the TPN model $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ is untimed 1-safe i.e. the untimed PN support $\langle \mathcal{N}, M_0 \rangle$ of the TPN $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ is 1-safe

2. $\mathcal{T} = \mathcal{T}_o \cup \mathcal{T}_{uo}$ where $\mathcal{T}_o$ is the set of observable events and $\mathcal{T}_{uo}$ is the set of unobservable (silent) events

3. l_o is the observation labeling function $l_o : \mathcal{T} \to \Omega_o \cup \{\epsilon\}$ where Ω_o is a set of labels and ϵ is the empty label. $l_o(t) = \epsilon$ iff $t \in \mathcal{T}_{uo}$ and $l_o(t) \in \Omega_o$ iff $t \in \mathcal{T}_o$. l_o is not necessary injective in Ω_o that is $\exists t_1, t_2 \in \mathcal{T}_o$ such that $t_1 \neq t_2$ and $l_o(t_1) = l_o(t_2)$.

4. when an observable transition $t^o \in \mathcal{T}_o$ is executed in the plant the label $l_o(t^o)$ is emitted together with the global time $\theta_{l_o(t^o)}$ when this execution of t^o took place.

5. the observation is always correct and the execution time of an observed event is measured with perfect accuracy according to a global clock

6. the observation is always received (there is no loss of observation) and there are no delays in receiving the observation

7. the execution of an unobservable event does not emit anything (is silent)

8. $\mathcal{T}_f \subset \mathcal{T}_{uo}$ is the set of faulty transitions. $\mathcal{T}_f$ is partitioned as $\mathcal{T}_f = \mathcal{T}_{\mathtt{F}_1} \cup \ldots \cup \mathcal{T}_{\mathtt{F}_\upsilon}$ where $\mathtt{F}_\iota$ is the subset of fault transitions that models a fault of kind $\mathtt{F}_\iota$.

9. the faults are unpredictable, i.e. $\forall t \in \mathcal{T}_f$, $\exists t' \in \mathcal{T} \setminus \mathcal{T}_f$ s.t. $i)$ ${}^\bullet t' \subseteq {}^\bullet t$ and $ii)$ $L^s_{t'} \leq U^s_t$.

Remark 19. *As for the untimed models (see Assumption 2) we assume at item* 9) *in the setting that the faults are unpredictable otherwise the diagnosis algorithm must include also a prognosis module since faults may be detected that will happen for sure. Notice that the condition at item* 9) *is a sufficient condition for the faults in the TPN to be unpredictable.*

Formal description of the problem

Given the plant model as described above design an on-line algorithm that derives the diagnosis of plant based on the model and the received observation. The exact meaning of diagnosis is defined as follow.

Assumption 7. *We make the assumption that the time diverges when an infinite number of transitions are executed (a cycle that contains only unobservable transitions that can be executed infinitely often contains at least one transition that has an non-zero lower limit of its static interval).*

Denote the observation received up to a global time ξ by:

$$\mathcal{O}^{\theta}_{\xi} = \langle obs_1, \theta_{obs_1}\rangle, \ldots, \langle obs_n, \theta_{obs_n}\rangle$$

where $obs_1, \ldots, obs_n \in \Omega_o$ are the labels that are emitted and $\theta_{obs_1}, \ldots, \theta_{obs_n}$ are the times at which the corresponding events occur.

Notice that by the assumptions that we have made in the setting there are no delays in receiving the observation and there are no observations that are lost. In other words the diagnoser agent receives the message $\langle obs_n, \theta_{obs_n}\rangle$ at time θ_{obs_n} for each n. Thus we have that $\theta_{obs_n} \leq \xi$ and the next observed event, if any, will be executed at a time after ξ.

Abusing notation we denote in the remaining of this thesis by $\mathcal{O}^{\theta}_{n}$ all the plant observation up to the time when the n^{th} observed event has occurred.

$\mathcal{L}^{\theta}_{\mathcal{N}^{\theta}}(M^{\theta}_0, \mathcal{O}^{\theta}_{\xi})$ is the set of all time traces that are feasible in $\langle \mathcal{N}^{\theta}, M^{\theta}_0\rangle$ up to the time ξ and that obey the received observation $\mathcal{O}^{\theta}_{\xi}$.

We say that a time-trace τ^{θ} obeys the observation $\mathcal{O}^{\theta}_{\xi}$ if $\tau^{\theta} \in \mathcal{L}^{\theta}_{\mathcal{N}^{\theta}}(M^{\theta}_0, \xi)$ and:

1. $l_o(\tau) = \mathcal{O}_{\xi}$ (the untimed support τ of the legal trace τ^{θ} obeys the untimed observation support trace $\mathcal{O}_{\xi}$)

2. and for $k = 1, \ldots, n$, $\theta_{t^o_k} = \theta_{obs_k}$ with $l_o(t^o_k) = obs_k$ and n the number of observed events in $\mathcal{O}^{\theta}_{\xi}$ (the execution time $\theta_{t^o_k}$ of each observable transition t^o_k ($l(t^o_k) = obs_k$) in τ^{θ} is equal with the time θ_{obs_k} that was reported)

In the following we use for $\mathcal{L}^{\theta}_{\mathcal{N}^{\theta}}(M^{\theta}_0, \mathcal{O}^{\theta}_{\xi})$ the simplified notation $\mathcal{L}^{\theta}(\mathcal{O}^{\theta}_{\xi})$ where the lower index $\mathcal{N}^{\theta}$ and the index M^{θ}_0 are dropped since it is clear from the context that we are analyzing $\langle \mathcal{N}^{\theta}, M^{\theta}_0\rangle$. Similarly for $\mathcal{L}^{\theta}_{\mathcal{N}^{\theta}}(M^{\theta}_0, \xi)$ (the set of time-traces derived up to ξ) we use in what follows the simplified notation $\mathcal{L}^{\theta}(\xi)$.

Denote by $\mathcal{D}(\mathcal{O}^{\theta}_{\xi})$ the plant diagnosis at time ξ based on the received observation $\mathcal{O}^{\theta}_{\xi}$. $\mathcal{D}(\mathcal{O}^{\theta}_{\xi})$ comprises the time-strings that are obtained from the projection of the time-traces contained in $\mathcal{L}^\theta(\mathcal{O}_\xi^\theta)$ on the set of fault transitions $\mathcal{T}_f$:

$$\mathcal{D}(\mathcal{O}_\xi^\theta) = \{\tau_f^\theta \mid \tau_f^\theta = \Pi_{\mathcal{T}_f}(\tau^\theta) \wedge \tau^\theta \in \mathcal{L}^\theta(\mathcal{O}_\xi^\theta)\} \tag{5.5}$$

Notice that the projection above applies to the untimed support τ of the time trace τ^θ and the obtained string τ_f^θ preserves the information about the time the fault transitions in τ^θ are executed.

If we have the set of fault events given by the partition $\mathcal{T}_f = \mathcal{T}_{\mathtt{F}_1} \cup \ldots \cup \mathcal{T}_{\mathtt{F}_\upsilon}$ then the detection of a fault of kind $\mathtt{F}_\iota$ is obtained by further projecting the strings of $\mathcal{D}(\mathcal{O}_\xi^\theta)$ on $\mathcal{T}_{\mathtt{F}_\iota}$:

$$\mathcal{D}_{\mathtt{F}_\iota}(\mathcal{O}_\xi^\theta) = \{\tau_{\mathtt{F}_\iota}^\theta \mid \tau_{\mathtt{F}_\iota}^\theta = \Pi_{\mathcal{T}_{\mathtt{F}_\iota}}(\tau_f^\theta) \wedge \tau_f^\theta \in \mathcal{D}(\mathcal{O}_\xi^\theta)\} \tag{5.6}$$

For a given kind of fault, say $\mathtt{F}_\iota$, we have that the diagnosis result of the plant w.r.t. faults of kind $\mathtt{F}_\iota$ at time ξ with received observation $\mathcal{O}_\xi^\theta$ is:

$$\mathcal{DR}_{\mathtt{F}_\iota}(\mathcal{O}_\xi^\theta) = \begin{cases} \mathtt{F}_{\mathtt{F}_\iota} & \text{iff } \epsilon \notin \mathcal{D}_{\mathtt{F}_\iota}(\mathcal{O}_\xi^\theta) \\ \mathtt{N}_{\mathtt{F}_\iota} & \text{iff } \{\epsilon\} = \mathcal{D}_{\mathtt{F}_\iota}(\mathcal{O}_\xi^\theta) \\ \mathtt{UF}_{\mathtt{F}_\iota} & \text{iff } \{\epsilon\} \subsetneq \mathcal{D}_{\mathtt{F}_\iota}(\mathcal{O}_\xi^\theta) \end{cases} \tag{5.7}$$

where similarly as defined for the untimed case in [SSL$^+$95] we have that:

1. $\mathtt{F}_{\mathtt{F}_\iota}$ means that a fault of kind $\mathtt{F}_\iota$ did necessarily happen in the plant:

$$\forall \tau_f^\theta \in \mathcal{D}_{\mathtt{F}_\iota}(\mathcal{O}_\xi^\theta) \Rightarrow \Pi_{\mathcal{T}_{\mathtt{F}_\iota}}(\tau_f) \neq \epsilon$$

2. $\mathtt{N}_{\mathtt{F}_\iota}$ means that a fault of kind $\mathtt{F}_\iota$ did not happen in the plant:

$$\forall \tau_f^\theta \in \mathcal{D}_{\mathtt{F}_\iota}(\mathcal{O}_\xi^\theta) \Rightarrow \Pi_{\mathcal{T}_{\mathtt{F}_\iota}}(\tau_f) = \epsilon$$

3. $\mathtt{UF}_{\mathtt{F}_\iota}$ means that it is uncertain whether a fault of kind $\mathtt{F}_\iota$ happened or not in the plant that is, there exist two legal time-strings $\tau_f^\theta, \tau_f'^\theta \in \mathcal{D}_{\mathtt{F}_\iota}(\mathcal{O}_\xi^\theta)$ s.t. $\Pi_{\mathcal{T}_{\mathtt{F}_\iota}}(\tau_f) \neq \epsilon$ and $\Pi_{\mathcal{T}_{\mathtt{F}_\iota}}(\tau_f') = \epsilon$.

5.4 Centralized diagnosis of TPN models

The set of firing times of a transition t in a TPN is in general uncountable since the time θ_t when t fires is given in general by a certain time-interval $[L(t), U(t)]$ where $L(t) < U(t)$. Thus the set of time traces $\mathcal{L}_{\mathcal{N}^\theta}^\theta(M_0^\theta)$ and the set of reachable states in a TPN are uncountable sets. In order to have a finite representation of the set of reachable states of a TPN [BM83] proposed the analysis of the TPN models based on state classes. The state class graph explicitly represents the untimed support language of $\langle\mathcal{N}^\theta, M_0^\theta\rangle$.

This is important since the diagnosis result at a given time ξ based on the received observation $\mathcal{O}^{\theta}_{\xi}$ (see Definition 5.7) requires only to calculate the untimed support of the set of time-traces $\mathcal{L}^{\theta}(\mathcal{O}^{\theta}_{\xi})$ that are possible in the TPN model and obey the received observation.

5.4.1 Analysis of Time Petri Nets based on state classes

Given a state $S = (M, FI)$ an enabled transition t may fire at any time in its temporal interval $FI(t)$, hence in general a state has infinitely many successors. For analyzing the system's behavior we must find a finite representation of the set of all possible reachable states by grouping the states into sets called state classes [BM83].

Definition 54. *Consider a time trace τ^{θ} in a TPN $\langle \mathcal{N}^{\theta}, M_0^{\theta} \rangle$ with $\tau^{\theta} = S_0 \xrightarrow{\theta_{t_1}} S_1 \xrightarrow{\theta_{t_2}} \ldots S_{\upsilon-1} \xrightarrow{\theta_{t_\upsilon}} S_\upsilon$. The times at which $\{t_1, t_2, \ldots, t_\upsilon\}$ are executed $\{\theta_{t_1}, \theta_{t_2}, \ldots \theta_{t_\upsilon}\}$ are called the path variables and are denoted by the vector Θ. The path variables keep track of the past evolution of the system, from the initial state S_0 up to the state S_υ that is reached after the execution of the last transition in τ^{θ}.*

The state variables are represented by the possible firing times (denoted ϑ_t) of the enabled transitions $t \in \mathcal{T}, M_\upsilon \geq Pre(\cdot, t)$ and are represented by the vector $\mathbf{v}$ where $\vartheta_t \in FI(t)$.

The path variables Θ and the state variables $\mathbf{v}$ are related by a system of inequalities (the characteristic system of τ^{θ}) denoted K_τ and having the shape:

1) $A \cdot \Theta \leq \mathbf{a}$

2) $\Theta^{en} = B \cdot \Theta$

3) $\Theta^{en} + \mathbf{L} \leq \mathbf{v} \leq \Theta^{en} + \mathbf{U}$

where: A is $m \times m$ matrix (with m the number of transitions that were executed) that relates the firing times of the past transitions; $\mathbf{a}$ is a vector of constant rational numbers. B is a $m \times q$ matrix (with q the number of enabled transitions) that determines the enabling times of the enabled transitions; Θ^{en} is a vector of dimension q having the component ι equal to the global time $\theta^{en}_{t_\iota}$ when t_ι has become enabled; $\mathbf{L}, \mathbf{U}$ are vectors of dimension m with the components specifying the earliest respectively and the latest global time a transition can be executed after it has become enabled.

Notice that by the assumption that $\langle \mathcal{N}, M_0 \rangle$ is 1-safe we have that the *parents* of a transition t_ι (the previous transitions whose occurrence made t_ι enabled) are uniquely defined for any trace τ^{θ}. We have that $\Theta^{en}_{t_\iota} = \mathtt{max}_{t_\gamma \in \bullet\bullet t_\iota}(\theta_{t_\gamma})$ if the input places of t_ι are not marked by tokens from the initial marking. If a transition t_ι is enabled by the tokens from the initial marking then $\theta^{en}_{t_\iota} = 0$ since the marking is assumed produced by a a fictitious transition that fires at the time 0.

Algorithm (Computing characteristic systems)

1 $K_\epsilon = \{L^s_t \le \vartheta_t \le U^s_t \mid t \in \mathcal{T} \wedge M_0 \ge Pre(\cdot, t)\}$

2 assume $S_0 \xrightarrow{\tau^\theta} S_\nu$. Then $t_{\nu+1}$ is fireable from S_ν iff:

2.1 $M_\nu \ge Pre(\cdot, t_{\nu+1})$

2.2 $K_\tau \;\wedge\; \left\{\vartheta_{t_{\nu+1}} \le \vartheta_{t_\iota} \mid t_\iota \ne t_{\nu+1} \;\wedge\; M_\nu \ge Pre(\cdot, t_\iota)\right\}$ is consistent (the solution set is not empty).

3 if $t_{\nu+1}$ is fireable then $K_{\tau t_{\nu+1}}$ is computable from K_τ as:

3.1 the fireability constraints for $t_{\nu+1}$ given by 2.2 above are added to the characteristic system K_τ

3.2 the variable $\vartheta_{t_{\nu+1}}$ is renamed into a path variable $\theta_{t_{\nu+1}} = \vartheta_{t_{\nu+1}}$

3.3 for each newly enabled transition t_λ at $M_{\nu+1}$, a new variable ϑ_{t_λ} is introduced and then:

3.3.1 for all the transitions t_ι that remained enabled after firing $t_{\nu+1}$ ($\forall t_\iota,\ M - Pre(\cdot, t_{\nu+1}) \ge Pre(\cdot, t_\iota)$) we have that:

$$\mathtt{max}(\theta^{en}_{t_\iota} + L^s_{t_\iota}, \theta_{\nu+1}) \le \theta_{t_\iota}$$

and $\theta_{t_\iota} \le \mathtt{max}(\theta^{en}_{t_\iota} + U^s_{t_\iota}, \theta_{\nu+1})$ remains the same.

3.3.2 for the newly enabled transitions t_λ we have $\theta^{en}_{t_\lambda} = \theta_{t_{\nu+1}}$ and $\theta^{en}_{t_\lambda} + L^s_{t_\lambda} \le \vartheta_{t_\lambda} \le \theta^{en}_{t_\lambda} + U^s_{t_\lambda}$

3.4 the variable $\vartheta_{\nu+1}$ is eliminated

Linear state classes

Originally state classes were introduced in [BM83] for finitely representing the state graphs of bounded TPNs. Linear state classes are constructed based on the following fact: consider two time traces τ^θ and τ'^θ (having the same untimed support) that lead the system from the initial state S_0 into a marking M and the characteristic systems K_τ and $K_{\tau'}$ have equal solution sets after projection on their state variables $\mathbf{v}$ ($K_\tau \mid_{\mathbf{v}} = K_{\tau'} \mid_{\mathbf{v}}$). Then the subtrees of the state graph SG rooted in the states given by K_τ and by $K_{\tau'}$ are isomorphic. Linear state classes are characteristic systems considered modulo this equivalence.

Definition 55. *The linear state class graph of a TPN $\langle \mathcal{N}^\theta, M^\theta_0\rangle$ is:*

$$LSCG = (\mathcal{SC}_{/\cong}, \overset{*}{\rightarrow}, [\{S_0\}]_{\cong})$$

where $\mathcal{SC}$ is a cover of $\mathcal{S}$ (a set of subsets of $\mathcal{S}$ whose union includes $\mathcal{S}$) inductively defined as:

1. $\mathcal{SC} = \bigcup_{\tau \in \mathcal{L}_{\mathcal{N}}(M_0)} SC_\tau$*, with* $SC_\epsilon = \{S_0\}$*,*
   $SC_{\tau t} = \left\{ S' \mid \exists S \in SC_\tau \text{ s.t. } S \xrightarrow{\langle t, \theta_t \rangle} S' \right\}$

2. $SC \cong SC'$ *iff* $(\exists S \in SC)(\exists S' \in SC')$ *s.t.* $M(S) = M(S')$ *and*

$$\bigcup_{S \in SC} FI(S) = \bigcup_{S' \in SC'} FI(S')$$

3. $SC \xrightarrow{t} SC''$ *iff* $\exists S \in SC$ *and* $\exists S'' \in SC''$ *s.t.* $S \xrightarrow{t} S''$

Each state class SC_τ contains all states reachable from the initial state by firing time-traces of untimed support τ. A state S may belong to more than one state class. All the states contained in a state class have the same marking while the firing domain of a state class is the union of the firing domains of each state included in the state class. We use the notation $\mathcal{FI}(SC)$ to denote the firing domain of a state class.

The linear state classes are obtained from the characteristic systems after elimination of path variables $\boldsymbol{\Theta}$.

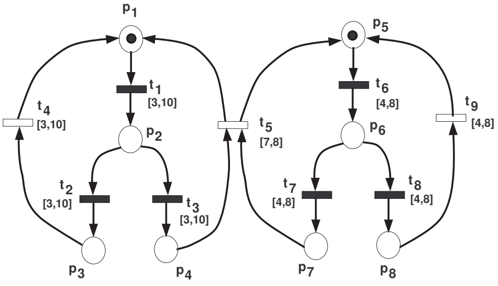


Figure 5.4:

Example 26. *Consider the TPN in Fig. 5.4 were the static time intervals are:* $I^s(t_\iota) = [3, 10]$ *for* $\iota = 1, \ldots, 4$*;* $I^s(t_\lambda) = [4, 8]$ *for* $\lambda = 6, \ldots, 9$ *while the synchronizing transition* t_5 *has its static time interval* $I^s(t_5) = [7, 8]$*.*

The linear state class graph is constructed as follows. In the initial marking $M_0 = \{M(p_1) = 1, M(p_5) = 1\}$ *we have* t_1 *and* t_6 *as enabled transitions, and we assume that the tokens arrived in* p_1 *respectively* p_5 *at the time* 0 *(the analysis starts at the global time* 0*).*

$$SC_0 \qquad K_\epsilon = \begin{cases} 3 \leq \vartheta_{t_1} \leq 10 \\ 4 \leq \vartheta_{t_6} \leq 8 \end{cases} \tag{5.8}$$

Consider now that t_1 fires first. The new marking is $M_1 = \{M(p_2) = 1, M(p_5) = 1\}$, the set of newly enabled transition is $\{t_2, t_3\}$ while t_6 remains enabled after firing t_1. The characteristic system K_{t_1} of the state class SC_1 is:

$$SC_1 \qquad K_{t_1} = \begin{cases} 3 \leq \theta_{t_1} \leq 10 \\ 4 \leq \vartheta_{t_6} \leq 8 \\ \theta_{t_1} \leq \vartheta_{t_6} \\ \theta_{t_1} + 3 \leq \vartheta_{t_2} \leq \theta_{t_1} + 10 \\ \theta_{t_1} + 3 \leq \vartheta_{t_3} \leq \theta_{t_1} + 10 \end{cases} \tag{5.9}$$

If t_6 fires first then the new marking is $M_2 = \{M(p_1) = 1, M(p_6) = 1\}$ and the set of newly enabled transition is $\{t_7, t_8\}$ while t_1 remains enabled after firing t_6. The characteristic system K_{t_6} of the state class SC_2 is:

$$SC_2 \qquad K_{t_6} = \begin{cases} 4 \leq \theta_{t_6} \leq 8 \\ 3 \leq \vartheta_{t_1} \leq 10 \\ \theta_{t_6} \leq \vartheta_{t_1} \\ \theta_{t_6} + 4 \leq \vartheta_{t_7} \leq \theta_{t_6} + 8 \\ \theta_{t_6} + 4 \leq \vartheta_{t_8} \leq \theta_{t_6} + 8 \end{cases} \tag{5.10}$$

Consider now that t_1 has fired first and then t_2 fires. The new marking is $M_3 = \{M(p_3) = 1, M(p_5) = 1\}$ an the set of newly enabled transition is $\{t_4\}$ while t_6 remains enabled after firing t_2. The characteristic system $K_{t_1 t_2}$ of the state class SC_3 is:

$$SC_3 \qquad K_{t_1 t_2} = \begin{cases} 3 \leq \theta_{t_1} \leq 10 \\ \theta_{t_1} + 3 \leq \theta_{t_2} \leq \theta_{t_1} + 10 \\ 4 \leq \vartheta_6 \leq 8 \\ \theta_{t_1} \leq \vartheta_{t_6} \\ \theta_{t_2} \leq \vartheta_{t_6} \\ \theta_{t_2} + 3 \leq \vartheta_{t_4} \leq \theta_{t_2} + 10 \end{cases} \tag{5.11}$$

Consider now that t_1 has fired first and then t_6 fires. The new marking is $M_4 = \{M(p_2) = 1, M(p_6) = 1\}$ an the set of newly enabled transition is $\{t_7, t_8\}$ while t_2, t_3 remain enabled after firing t_6. The characteristic system $K_{t_1 t_6}$ of the state class SC_4 is:

$$SC_4 \qquad K_{t_1 t_6} = \begin{cases} 3 \le \theta_{t_1} \le 10 \\ 4 \le \theta_{t_6} \le 8 \\ \theta_{t_1} \le \theta_{t_6} \\ \theta_{t_6} \le \vartheta_{t_2} \\ \theta_{t_6} \le \vartheta_{t_3} \\ \theta_{t_1} + 3 \le \vartheta_{t_2} \le \theta_{t_1} + 10 \\ \theta_{t_1} + 3 \le \vartheta_{t_3} \le \theta_{t_1} + 10 \\ \theta_{t_2} + 4 \le \vartheta_{t_7} \le \theta_{t_2} + 8 \\ \theta_{t_2} + 4 \le \vartheta_{t_8} \le \theta_{t_2} + 8 \end{cases} \tag{5.12}$$

The linear state class tree rooted in SC_0 is displayed in Fig. 5.5 where for $\nu = 0 \dots 4$ the linear state classes are $SC_\nu = (M_\nu, \mathcal{FI}_\nu)$. M_ν is the marking for each of the cases described before and $\mathcal{FI}_0 = K_\epsilon \mid_\vartheta$, $\mathcal{FI}_1 = K_{t_1} \mid_\vartheta$, $\mathcal{FI}_2 = K_{t_6} \mid_\vartheta$, $\mathcal{FI}_3 = K_{t_1 t_2} \mid_\vartheta$ and $\mathcal{FI}_4 = K_{t_1 t_6} \mid_\vartheta$.

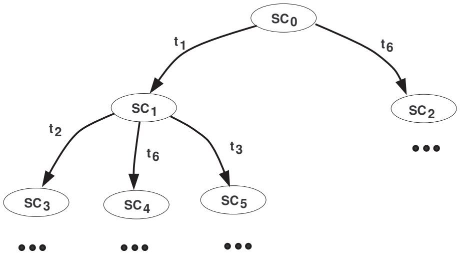


Figure 5.5:

Now consider that t_1 fires at the time $\theta_{t_1} = 3$ leading the plant in to the new state $S_1 \in SC_1$ ($S_0 \xrightarrow{\langle t_1, 3 \rangle} S_1$). It is easy to check that S_1 has successors in the successor state classes of SC_1 namely $S_3 \in SC_3$, $S_4 \in SC_4$, and $S_5 \in SC_5$ respectively, where for $\theta_{t_2} = 6$, $\theta_{t_6} = 7$ and $\theta_{t_3} = 6$ we have $S_1 \xrightarrow{\langle t_2, 6 \rangle} S_3$, $S_1 \xrightarrow{\langle t_6, 7 \rangle} S_4$, $S_1 \xrightarrow{\langle t_3, 6 \rangle} S_5$.

Then if t_1 fires at $\theta_{t_1} = 6$ we obtain the state $S_1 \in SC_1$ that has successors only in SC_4 and not in SC_3 and SC_5.

Thus we have that in general not all the states within a state class have successors in a successor state class. In order to assure the atomicity property that all the states within a state class have successors in all the successor state classes the linear state classes have to be split [YR98].

The procedure to refine the linear state class graph $LSCG$ such that the atomicity property is satisfied was proposed in [YR98] in the context of CTL*

Model Checking and then improved in [BV02]. In the following without going into all the details we present the concept of atomic state classes and sketch the construction of the atomic state class graph.

Atomic state classes

Definition 56. *A state class SC_ι is atomic if, for each state class SC_λ, whenever a state $S_\iota \in SC_\iota$ has a successor S_λ in SC_λ ($\exists S_\lambda \in SC_\lambda$ s.t. $S_\iota \xrightarrow{\langle t,\theta_t\rangle} S_\lambda$) then all the states of SC_ι have successors in SC_λ ($\forall S'_\iota \in SC_\iota \Rightarrow \exists S'_\lambda \in SC_\lambda$ s.t. $S'_\lambda \xrightarrow{\langle t,\theta'_t\rangle} S'_\lambda$).*

In order to generate atomic state classes one must impose some linear constraints for splitting the linear state classes until the atomicity is obtained.

Example 27. *For imposing the atomicity property one must split the state classes by adding linear constrains of the form κ and $\neg\kappa$. For instance for the LSCG derived in Example 26 κ is expressed as $\theta_{t_1} \leq U^s_{t_6} - L^s_{t_2}$ meaning that κ,$\neg\kappa$ split SC_1 into SC'_1 and SC''_1 (all the states of SC'_1 obey κ and all the states of SC''_1 obey $\neg\kappa$). $\theta_{t_1} \leq U^s_{t_6} - L^s_{t_2}$ means that if t_2 fires at its earliest time being enabled from the time θ_{t_1} then it can fire before t_6 (where the latest time is considered).*

Notice that splitting a state class may in general cause as well the splitting of the predecessor state classes. For instance if a linear constraint $\kappa_{\nu+1}$ is imposed to state class $SC_{\nu+1}$ to determine the state class region that contains states having successors in $SC_{\nu+2}$ then for preserving the atomicity the predecessor state classes $SC_\nu, SC_{\nu-1}\ldots$ of $SC_{\nu+1}$ must be refined as well imposing whenever necessary, adequate linear constraints $\kappa_\nu, \kappa_{\nu-1}, \ldots$.

Without going into details we assume that following the algorithm proposed in [YR98] we have obtained the atomic state class graph:

$$ASCG = (\mathcal{ASC}, \xrightarrow{*}, ASC_0)$$

where $ASC_0 = \{S_0\}$ is the initial state $S_0 = (M_0, FI_0)$ and $\mathcal{ASC}$ is the set of atomic state classes that is $\forall ASC_\iota, ASC_\lambda \in \mathcal{ASC}$ we have that $ASC_\iota \xrightarrow{t} ASC_\lambda$ iff $\forall S_\iota \in ASC_\iota \Rightarrow \exists S_\lambda \in ASC_\lambda$ s.t. $S_\iota \xrightarrow{\langle t,\theta_t\rangle} S_\lambda$.

Denote by ρ a path in the atomic state class graph $ASCG$ where:

$$\rho := ASC_0 \xrightarrow{t_1} ASC_1 \ldots ASC_{\upsilon-1} \xrightarrow{t_\upsilon} ASC_\upsilon$$

and denote $\tau(\rho)$ the untimed support trace that corresponds with ρ. Then denote by $\mathcal{L}^\theta(ASCG)$ the set of timed traces that are represented by $ASCG$ and denote by $\mathcal{L}(ASCG)$ the untimed support of $\mathcal{L}^\theta(ASCG)$ where:

$$\mathcal{L}(ASCG) = \{\tau(\rho) \mid \rho \in ASCG\}$$

Proposition 22. *Given a Time Petri Net $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ and the atomic state class graph $ASCG(\langle \mathcal{N}^\theta, M_0^\theta \rangle)$ we have that $\mathcal{L}^\theta_{\mathcal{N}^\theta}(M_0^\theta) = \mathcal{L}^\theta(ASCG)$ and $\mathcal{L}_{\mathcal{N}^\theta}(M_0^\theta) = \mathcal{L}(ASCG)$.*

Proof. Trivial. □

5.4.2 Centralized diagnosis of TPNs based on atomic state class graph

In this section we present an on-line algorithm that derives the plant diagnosis based on the calculation of the atomic state class graph. The on-line diagnosis-algorithm that we design works as follows:

1. when the process starts set initial condition $\mathcal{O}_0^\theta = \epsilon, n = 1$

2. compute the atomic state class graph $ASCG(\mathtt{obs_n} \mid \mathcal{O}_{n-1}^\theta)$ whose paths correspond with untimed traces that contain only unobservable transitions except for the last transition that is observable

3. for each path ρ_ν of $ASCG(\mathtt{obs_n} \mid \mathcal{O}_{n-1}^\theta)$ let $\widehat{\theta}_\nu$ be the latest global time when the estimated observable transition t_ν^o can be executed ($\widehat{\theta}_\nu = U_\nu(t_\nu^o)$) where in the solution of the characteristic system $Sol(K_{\tau(\rho_\nu)})$ we have that $\theta_{t_\nu^o} \leq U_\nu(t_\nu^o)$.

4. with the global time progress $\xi = \xi + \delta\xi$ we have the following cases:

   (a) if no observation is received by the time ξ and $\xi > \widehat{\theta}_\nu$ then delete the path ρ_ν

   (b) if at time ξ, $\langle obs_n, \theta_{obs_n} \rangle$ is received ($\mathcal{O}_\xi^\theta = \mathcal{O}_{n-1}^\theta \langle obs_n, \theta_{obs_n} \rangle$ and $\mathcal{O}_{\xi-\delta\xi}^\theta = \mathcal{O}_{n-1}^\theta$) then:

      i. for all paths ρ_ν s.t. $l_o(t_{n_\nu}^o) \neq obs_n$ delete ρ_ν

      ii. for all paths ρ_ν s.t. $l_o(t_{n_\nu}^o) = obs_n \wedge \theta_{obs_n} \notin [L_\nu(t_{n_\nu}^o), U_\nu(t_{n_\nu}^o)]$ delete ρ_υ

      iii. for all paths ρ_ν s.t. $l_o(t_n^o) = obs_n \wedge \theta_{obs_n} \in [L_\nu(t_{n_\nu}^o), U_\nu(t_{n_\nu}^o)]$

         A. impose the constraint that $t_{n_\nu}^o$ occurred at the time θ_{obs_n}: $\kappa := \left\{ \theta_{t_{n_\nu}^o} = \theta_{obs_n} \right\}$

         B. and then restore the atomicity property

5. denote $ASCG(\mathcal{O}_\xi^\theta)$ the atomic state class graph obtained in this way where $\xi = \theta_{obs_n}$

6. $n := n + 1$, return to 2

In the following we use the notation $ASCG(\texttt{obs}_\texttt{k} \mid \mathcal{O}^\theta_{k-1})$ for the atomic state class graph that estimates the k^{th} event that will be observed ($\texttt{obs}_\texttt{k}$) given the received observation $\mathcal{O}^\theta_{k-1}$. Notice that $\mathcal{O}^\theta_0 = \epsilon$ and $\mathcal{O}^\theta_k = \mathcal{O}^\theta_{k-1}\langle obs_k, \theta_{obs_k}\rangle$.

For $\xi = \theta_{obs_n}$ consider the atomic state class graph $ASCG(\mathcal{O}^\theta_\xi)$ where:

$$ASCG(\mathcal{O}^\theta_\xi) = (\mathcal{ASC}_\xi, \overset{*}{\rightarrow}, ASC_0)$$

with $ASC_0 = \{S_0\}$ initial atomic state class and $\mathcal{ASC}$ the set of atomic state classes derived after imposing the constrains due to the received observation $\mathcal{O}^\theta_\xi$ and the constraints that restore the atomicity property.

Denote by $\mathcal{ASC}_\xi$ the set of atomic state classes that are obtained at the time ξ after the execution of the last observed event and then denote by ρ_ν a complete path in $ASCG(\mathcal{O}^\theta_\xi)$ where:

$$\rho_\nu = ASC_0 \xrightarrow{t^o_{1_\nu}} ASC_{1_\nu} \dots ASC_{q-1_\nu} \xrightarrow{t^o_{q_\nu}} ASC_{q_\nu} \tag{5.13}$$

Denote by $\tau_\nu = t_{1_\nu} \dots t_{q_\nu}$ the untimed support of ρ_ν. For τ_ν we have that:

1. $l_o(\tau_\nu) = \mathcal{O}_\xi$
2. and for $k = 1, \dots, n \quad l_o(t^o_{k_\nu}) = obs_k$ and $\theta_{t^o_{k_\nu}} = \theta_{obs_k}$

Let $K_{\rho_\nu}(\mathcal{O}^\theta_\xi)$ be the characteristic system associated with the path ρ_ν in $ASCG(\mathcal{O}^\theta_\xi)$ where $K_{\rho_\nu}(\mathcal{O}^\theta_\xi)$ includes the constraints due to the received observation and the constraints that were imposed for preserving the atomicity property.

Denote by $\mathcal{L}^\theta(ASCG(\mathcal{O}^\theta_\xi))$ the set of time-traces up to the time ξ that explain the received observation:

$$\mathcal{L}^\theta(ASCG(\mathcal{O}^\theta_\xi)) = \left\{\tau^\theta_\nu \mid \rho_\nu \in ASCG(\mathcal{O}^\theta_\xi) \wedge \Theta_\nu \text{ is a solution of } \; K_{\rho_\nu}\right\} \tag{5.14}$$

Let $\mathcal{L}(ASCG(\mathcal{O}^\theta_\xi))$ be the untimed language that comprises the untimed support of $\mathcal{L}^\theta(ASCG(\mathcal{O}^\theta_\xi))$.

Theorem 6. *Given the observation generated by the plant $\mathcal{O}^\theta_\xi$ we have that:*

$$\mathcal{L}^\theta(ASCG(\mathcal{O}^\theta_\xi)) = \mathcal{L}^\theta_{\mathcal{N}^\theta}(M^\theta_0, \mathcal{O}^\theta_\xi)$$

where ξ is the occurrence time of the last observed event in $\mathcal{O}^\theta_\xi$.

Proof. ($\Leftarrow$) First we prove that $\mathcal{L}^\theta(ASCG(\mathcal{O}^\theta_\xi)) \subseteq \mathcal{L}^\theta_{\mathcal{N}^\theta}(M^\theta_0, \mathcal{O}^\theta_\xi)$.

Consider a state $S_\xi \in ASC_{n_\iota}$ ($ASC_{n_\iota} \in \mathcal{ASC}_\xi$). Thanks to the atomicity property we have that there exists a state in the predecessor atomic state classes s.t. $S_0 \xrightarrow{\tau^\theta} S_\xi$. Then we have that τ^θ obeys the observation because of the constraints that were imposed.

$\Rightarrow$ By definition the union of the sets that are included in the state classes of $ASCG(\mathcal{O}^\theta_\xi)$ is a cover of the set of states that can be obtained at the time ξ given the received observation. Since each state is part of a legal time-trace then we have that $\mathcal{L}^\theta(ASCG(\mathcal{O}^\theta_\xi)) \supseteq \mathcal{L}^\theta_{\mathcal{N}^\theta}(M^\theta_0, \mathcal{O}^\theta_\xi)$.

□

Th. 6 shows that the plant diagnosis can be derived using $\mathcal{L}^\theta(ASCG(\mathcal{O}^\theta_\xi))$ and $\mathcal{L}(ASCG(\mathcal{O}^\theta_\xi))$ instead of $\mathcal{L}^\theta(\mathcal{O}^\theta_\xi)$ respectively $\mathcal{L}(\mathcal{O}^\theta_\xi)$ in Eq. 5.5, Eq. 5.6, and Eq. 5.7.

The set of estimated states of the plant at the time ξ when the last observed event was received is given by the union of atomic state classes that are obtained in $ASCG(\mathcal{O}^\theta_\xi)$ considering the execution of the last observed event on a path ρ_ν:

$$\mathcal{S}_\xi = \left\{ S_\xi \mid S_\xi \in ASC_{n_\nu} \wedge ASC_0 \xrightarrow{\rho_\nu} ASC_{n_\nu} \wedge \rho_\nu \in ASCG(\mathcal{O}^\theta_\xi) \right\}$$

Example 28. *Consider again the TPN in Fig. 5.4 where $\mathcal{T}_o = \{t_4, t_5, t_9\}$ and $l_o(t_4) = l_o(t_5) = l_o(t_9)$.*

We illustrate the construction of $ASCG(\texttt{obs}_1 \mid \mathcal{O}^\theta_0)$ considering the untimed support $\tau = t_1 t_2 t_6 t_4$. The linear state class that corresponds to τ is as follows:

$$K_\tau = \begin{cases} 3 \le \theta_{t_1} \le 10 \\ 4 \le \theta_{t_6} \le 8 \\ \theta_{t_1} + 3 \le \theta_{t_2} \le \theta_{t_1} + 10 \\ \theta_{t_2} + 3 \le \theta_{t_4} \le \theta_{t_2} + 10 \\ \theta_{t_1} \le \theta_{t_6} \\ \theta_{t_2} \le \theta_{t_6} \\ \theta_{t_6} \le \theta_{t_3} \\ \theta_{t_6} + 4 \le \vartheta_{t_7} \le \theta_{t_6} + 8 \\ \theta_{t_6} + 4 \le \vartheta_{t_8} \le \theta_{t_6} + 8 \\ \theta_{t_4} \le \vartheta_{t_7} \\ \theta_{t_4} \le \vartheta_{t_8} \end{cases} \tag{5.15}$$

After simple manipulations we obtain:

$$K_\tau = \begin{cases} 3 \leq \theta_{t_1} \leq 8 \\ 4 \leq \theta_{t_6} \leq 8 \\ 6 \leq \theta_{t_2} \leq 8 \\ 9 \leq \theta_{t_4} \leq 16 \\ 10 \leq \vartheta_{t_7} \leq 16 \\ 10 \leq \vartheta_{t_8} \leq 16 \\ \theta_{t_1} \leq \theta_{t_6} \\ \theta_{t_2} \leq \theta_{t_6} \\ \theta_{t_6} \leq \theta_{t_4} \\ \theta_{t_4} \leq \vartheta_{t_7} \\ \theta_{t_4} \leq \vartheta_{t_8} \end{cases} \tag{5.16}$$

Given the state class SC_{t_1} the state $S_1 \in SC_{t_1}$ that corresponds with the firing of t_1 at the time $\theta_{t_1} = 8$ has no successors in SC_τ since in τ, t_2 is assumed executed and t_2 is executed at least 3 time units after the execution of t_1.

By adding the constraints to preserve the atomicity property we have the atomic state classes ASC_τ and ASC_{t_1} obtained from K_τ^{at} and respectively $K_{t_1}^{at}$ by projecting the solution set on to the state variables:

$$K_{t_1}^{at} = \begin{cases} 3 \leq \theta_{t_1} \leq 5 \\ 4 \leq \vartheta_{t_6} \leq 8 \\ \theta_{t_1} \leq \vartheta_{t_6} \\ 6 \leq \vartheta_{t_2} \leq 8 \\ 6 \leq \vartheta_{t_3} \leq 8 \end{cases} \tag{5.17}$$

$$K_\tau^{at} = \begin{cases} 3 \leq \theta_{t_1} \leq 5 \\ 6 \leq \theta_{t_6} \leq 8 \\ 6 \leq \theta_{t_2} \leq 8 \\ 9 \leq \theta_{t_4} \leq 16 \\ 10 \leq \vartheta_{t_7} \leq 16 \\ 10 \leq \vartheta_{t_8} \leq 16 \\ \theta_{t_1} \leq \theta_{t_6} \\ \theta_{t_2} \leq \theta_{t_6} \\ \theta_{t_6} \leq \theta_{t_4} \\ \theta_{t_4} \leq \vartheta_{t_7} \\ \theta_{t_4} \leq \vartheta_{t_8} \end{cases} \tag{5.18}$$

For the path $\rho_\nu \in ASCG(\texttt{obs}_1 \mid \mathcal{O}_0^\theta)$ that corresponds to τ the expected time-interval the first observation of t_4 will be received is $\theta_{t_4} \in [9, 16]$.

Then we have that:

1. *if no observation is received until the global clock becomes 16 then ρ_ν is deleted from $ASCG(\mathcal{O}_1^\theta)$.*
2. *if an observation is received before time 9 then also ρ_ν should be deleted*
3. *if the label $l_o(t_4)$ is observed at the time $\theta_{l_o(t_4)} = 10$, $\theta_{t_4} = 10$ is added to K_τ^{at}; restore the atomicity property obtaining for t_2 and t_1 that $6 \leq \theta_2 \leq 7$ and respectively $3 \leq \theta_1 \leq 4$. We obtain $ASC_{\tau,\xi} = (M, \mathcal{FI})$ where $\mathcal{FI} = \{FI(t_1) = (10, 23], FI(t_7) = (10, 16], FI(t_8) = (10, 16]\}$*

Notice in this example that the atomic state class graph derived before receiving the first observation will have paths corresponding with all the untimed traces that are legal in the untimed support PN $\langle \mathcal{N}, M_0 \rangle$ and contain only one observable event that is executed as the last event in the trace.

In this example the time information does not reduce the number of untimed traces that are derived before the received observation. As one can see the consideration of all the possible interleaving between $\{t_1, t_2, t_3\}$ and $\{t_6, t_7, t_8\}$ leads very quickly to state space explosion.

We conclude this section with the following remarks:

1. the diagnosis algorithm based on atomic state classes is an adaptation of the state class method for TPNs where the plant behavior is constrained so that the plant behavior obeys the received observation. Moreover we have presented an on-line algorithm that computes the next observation estimation and then refines the calculations based on the received observation.
2. this approach suffers from the state space explosion due to the interleaving of the concurrent transitions. Even for TPN models having a reasonable size the plant computation becomes practically impossible.
3. this method is not suitable for distributed applications where a local agent performs the local site analysis and exchange information with its neighbors. The reason is twofold: first the plant calculation requires at any time the global state and secondly there is no easy way to include the new information that is available to a local agent after the information exchange takes place.

In order to overcome the state space explosion when the plant exhibits a high degree of concurrency we present in the next section the analysis of Time Petri Nets based on partial orders.

5.5 The analysis of TPNs based on time processes

As was already illustrated in Example 28 the timing information may not reduce the number of the interleavings of the concurrent (unobservable) events that are considered. The partial order reduction techniques developed for untimed PN [McM93], [Esp94] are shown in [HB95], [SY96], [AL97], [Lil98] applicable for TPN. Consider a configuration C in the unfolding $\mathcal{U}_{\mathcal{N}}(M_0)$ of the untimed PN support of a TPN (Section 2.5). Then consider a valuation Θ of the execution times at which the events $e \in E_C$ in the configuration C are executed that is for each $e \in E_C$ consider a time value $\theta_e \in \mathbb{T}$ ($\mathbb{T}$ the time axis) at which e occurs and Θ is an $\mid E_C \mid$-tuple comprising all the values at which all the events $e \in E_C$ are executed.

An untimed configuration C with a valuation $\Theta \in \mathbb{T}^{|E_C|}$ of the execution time for its events is called a time configuration (time process in [AL97]) of the TPN model.

A time configuration is valid if the execution times of the events in the configuration provide a solution of the system of inequalities called the characteristic system of the configuration.

A valid time configuration means that there exists a legal trace $\tau^\theta \in \mathcal{L}^\theta_{\mathcal{N}^\theta}(M^\theta_0)$ in the TPN $\langle \mathcal{N}^\theta, M^\theta_0 \rangle$ whose untimed τ support is a linearization (Def. 2) of the partial order relation of the events in the configuration (i.e. $\tau = \phi(\sigma)$ and $\sigma \in \langle E_C \rangle$) while the execution time θ_t of every transition t considered in the trace τ^θ is identical with the valuation of the event e to whom t is its image via ϕ.

Consider for each event e in the configuration C the static interval of its image transition $t = \phi(e)$ in the original TPN $\langle \mathcal{N}^\theta, M^\theta_0 \rangle$.

For a free-choice PN the characteristic system K_{C^θ} of a configuration C is the system of $(max, +) - linear$ inequalities obtained by considering for each event $e \in E_C$ in the configuration C a $(max, +)$ linear inequality expressing the fact that the execution time θ_e of each event $e \in E_C$ is within a delay in its static interval I^s_e ($I^s_e = I^s_t$, $\phi(e) = t$) after the time θ^{en}_e when e has become enabled.

Thus the characteristic system K_{C^θ} of the configuration considers for each event $e \in E_C$ two inequalities regarding the execution time θ_e of the event e having the form:

$$\mathtt{max}_{e' \in {}^{\bullet\bullet}e}(\theta_{e'}) + L^s_e \;\leq\; \theta_e \;\leq\; \mathtt{max}_{e' \in {}^{\bullet\bullet}e}(\theta_{e'}) + U^s_e \geq \theta_e$$

where $e' \in {}^{\bullet\bullet}e$ means that $e'^{\bullet} \cap {}^{\bullet}e \neq \emptyset$.

The reason why the characteristic system of a configuration in the unfolding of a free choice TPN contains variables only for the events that are executed in the configuration is that in a free choice PN any two transitions that are in conflict share the same input conditions. Thus any transition that becomes enabled in a TPN can fire at any time within its static interval.

The characteristic systems of a configuration contains inequalities expressed using $(max, +)$ algebra thus to check whether Θ is a solution of the characteristic system K_{C^θ} or not and moreover to derive all the valuations for which C is a valid time configuration requires enumeration of all the cases for a $(max, +)$ inequality that is for $e_1, e_2, e_3 \in E_C$ and $e_2, e_3 \in {}^{\bullet\bullet}e_1$ the inequalities regarding θ_{e_1} in K_{C^θ} can expressed as:

$$\begin{cases} \theta_{e_2} + L^s_{e_1} \leq \theta_{e_1} \\ \theta_{e_2} + U^s_{e_1} \geq \theta_{e_1} \\ \theta_{e_3} \leq \theta_{e_2} \\ \text{or} \\ \theta_{e_3} + L^s_{e_1} \leq \theta_{e_1} \\ \theta_{e_3} + L^s_{e_1} \geq \theta_{e_1} \\ \theta_{e_2} \leq \theta_{e_3} \end{cases} \tag{5.19}$$

If we do the same thing for all the concurrent events in configuration C that have common immediate predecessor events, the characteristic system K_{C^θ} is rewritten as a disjunction of systems of linear inequalities in a conjunctive form that can be brought to a canonical form (e.g. by Floyd-Warshall algorithm) and thus we can derive the set of all the time valuations $\Theta \in \mathbb{T}^{|E_C|}$ for which $C(\Theta)$ is a valid time configuration. Using the canonical conjunctive form we can derive the time intervals within which the events in the configuration are executed and thus derive valid time-interval configuration.

For a general TPN the characteristic system of a configuration should include some extra inequalities regarding the events in the unfolding $\mathcal{U}_{\mathcal{N}}(M_0)$ that are in conflict with events of the configuration C. An event $\breve{e} \in E$ is a conflicting event of a configuration C if $\breve{e} \in E$ is not included in the configuration C ($\breve{e} \notin E_C$), all its input conditions are contained in C (${}^\bullet\breve{e} \subseteq B_C$), and there is an event $e \in E_C$ to whom $\breve{e}$ is in conflict.

As already mentioned in a free-choice TPN any transition t that becomes enabled can fire at any time within its static interval. In a general TPN this is not true anymore because even though a transition t becomes enabled it is not guaranteed that t can be executed because some other transitions in cluster with which it shares only a part of its input conditions may be forced to be executed sooner than the earliest time when t can be executed thus disabling t.

Hence we should include in the characteristic system inequalities regarding the $no-execution$ of the conflicting events. These inequalities are in a disjunctive form expressing the fact that at least one event $e \in E_C$ with which the conflicting event $\breve{e}$ shares input conditions is executed before the time the conflicting event $\breve{e}$ reaches the upper limit of its static interval after it has become enabled, that is:

$$\bigwedge_{\breve{e}\in\breve{E}_C} \left\{ \bigvee_{\substack{e\sharp\breve{e}\\ e\in E_C}} \{\theta_e \leq \theta_{\breve{e}}^{en} + U_{\breve{e}}^{s}\} \right\} \tag{5.20}$$

where $\theta_{\breve{e}}^{en} = \mathtt{max}_{e' \in {}^{\bullet\bullet}\breve{e}}(\theta_{e'})$ and $\breve{E}_C$ is the set of conflicting events of configuration C.

The characteristic system K_{C^θ} of a configuration in a general TPN includes $(max, +)$ inequalities regarding the occurrence times of events that are executed in E_C as well as inequalities in a disjunctive form regarding the conflicting events that were not executed.

To derive the set of all valid time configurations we should bring the characteristic system K_{C^θ} in to a form of disjunction of systems of inequalities in conjunctive form. Let K_{C^θ} be expressed as $K_{C^\theta} = \bigvee_{\lambda\in\mathcal{V}}(K_{C^\theta}^{\lambda})$ where each system of inequalities $K_{C^\theta}^{\lambda}$, $\lambda \in \mathcal{V}'$ is in conjunctive form.

Each system of inequalities $K_{C^\theta}^{\lambda}$, $\lambda \in \mathcal{V}'$ can be brought in a canonical conjunctive form and then we have that $Sol(K_{C^\theta}) = \{Sol(K_{C^\theta}^{\lambda}) \mid \lambda \in \mathcal{V}''\}$ (obviously for the system of inequalities that have solutions).

As already mentioned we use time processes for designing on-line diagnosis algorithms for TPN models. The on-line monitoring algorithms that we design in this chapter can be briefly described as follows. When the process starts, time interval configurations are derived on-line up to the first discarding time when in absence of any observation the time-interval configuration can be discarded. The discarding time corresponds with the lowest upper limit of the execution time interval among the observable events that are included in the time interval configuration.

As in the previous section we assume for on-line monitoring that the plant observation includes the exact time at which an observable event occurs. The exact observation of the occurrence of an event is taken into account by adding to the characteristic system of a configuration an extra constraint (inequality) that specifies the time when the observed event was executed as well as adding constraints to express the fact that all the other observable events did not happen yet. This is because we assume that the observation is correct, there are no delays in receiving the plant observation and the time when a transition is executed is measured precisely.

Adding inequalities after receiving an observation requires to restore the canonical form of the characteristic system that includes the plant observation.

The on-line analysis continues by extending the time interval configurations that were obtained after imposing the constraints due to the received observation up to the next discarding time.

As presented above the on-line analysis of a TPN can be performed by deriving time-interval configurations translating the characteristic system of a configuration with the received observation into a disjunction of systems of inequalities in a conjunctive form that can be brought to a canonical form. However this approach is inconvenient because it requires the enumeration of the possible orders between concurrent events that have common predecessors events, as well as the enumeration for the general TPN of all the possible orders in which events of the configurations are executed before the conflicting events reach the upper limit of their firing intervals.

The idea behind the algorithm that we propose for deriving the entire set of solutions of the the characteristic system of a configuration is to use the causal relation encoded in the configuration itself for propagating the constraints among the execution times of the events in the configuration. The disjunctive inequalities that are encountered lead to the consideration of different cases where the main advantage is that it is not required to have an *a priori* enumeration of the order in which concurrent events are executed.

The set of all solutions of the characteristic system $Sol(K_{C^\theta})$ is obtained as a cover of subsets of the solution set e.g. $\{Sol_\nu(K_{C^\theta}) \mid \nu \in \mathcal{V}\}$ and $Sol(K_{C^\theta}) = \bigcup_{\nu\in\mathcal{V}} Sol_\nu(K_{C^\theta})$, where each subset of solutions $Sol_\nu(K_{C^\theta})$, $\nu \in \mathcal{V}$ has the *time independence property* for the concurrent events in the configuration. The time independence property of a subset of solutions of the characteristic system of a configuration can be intuitively understood as follows: *given any set of concurrent events in the configuration and fixing the execution times of their predecessors, their executions times belong to a hyper-rectangle in high dimensional space.*

We have that each sub-set of solutions $Sol(K^\lambda_{C^\theta})$, $\lambda \in \mathcal{V}''$ of the characteristic system K_{C^θ} that results as the solution set of a system of linear inequalities in canonical conjunctive form has the time independence property but a sub-set of solutions that we compute e.g. $Sol_\nu(K_{C^\theta})$, $\nu \in \mathcal{V}$ is not necessarily the solution set of a single conjunctive system of linear inequalities but it may be obtained as a union of different systems of inequalities in conjunctive form.

The time independence property of a subset of solutions of the characteristic system allows us to provide an adequate definition of the time-interval configuration. A time interval configuration is obtained by considering for each event e in C the execution time-intervals equal with the projection of the sub-set of solution $Sol_\nu(K_{C^\theta}) \subseteq Sol(K_{C^\theta})$ that has the time-independence property on to time-axis that correspond with the occurrence time of the events e.

For the sake of simplifying the presentation we consider in the following an arbitrary acyclic causal TPN that may be understood as a TPN obtained from an untimed configuration by attaching static time intervals to all its events. By doing this we avoid to discuss formally at this stage the on-line computation of a time interval-configuration.

5.5.1 Preliminaries

Definition 57. *A Petri Net $\widetilde{\mathcal{N}} = (\widetilde{\mathcal{P}}, \widetilde{\mathcal{T}}, \widetilde{F})$ is an acyclic causal net if:*

i) $\forall p \in \widetilde{\mathcal{P}}$ *(*$^\bullet p \leq 1$ *and* $p^\bullet \leq 1$*)*

ii) and $\widetilde{\mathcal{N}}$ *contains no circuits.*

Moreover assume that all the transition in $\widetilde{\mathcal{N}}$ *have non-empty sets of input places and output places:* $\forall t \in \widetilde{\mathcal{T}}$, $t^\bullet \neq \emptyset \wedge {}^\bullet t \neq \emptyset$.

Denote $\mathtt{min}(\widetilde{\mathcal{N}})$ *the set of places that have no input transitions:*

$$\mathtt{min}(\widetilde{\mathcal{N}}) = \left\{ p \in \widetilde{\mathcal{P}} \mid {}^\bullet p = \emptyset \right\}$$

Let $\widetilde{M_0}$ *be the initial marking of* $\widetilde{\mathcal{N}}$ *where* $\widetilde{M_0}(p) = 1$ *if* $p \in \mathtt{min}(\widetilde{\mathcal{N}})$ *and* $\widetilde{M_0}(p) = 0$ *otherwise.*

Then let I^s *be a function that associates to each transition* $t \in \widetilde{\mathcal{T}}$ *its static time interval:* $I^s : \widetilde{\mathcal{T}} \rightarrow \mathcal{I}_{\mathbb{Q}^+}$*, and denote* $\widetilde{\mathcal{N}}^\theta = (\widetilde{\mathcal{P}}, \widetilde{\mathcal{T}}, \widetilde{F}, I^s)$ *the TPN obtained in this way.*

Notice that if $\widetilde{\mathcal{N}}$ is a causal net then $\widetilde{F}$ is a partial order relation in $\widetilde{\mathcal{P}} \cup \widetilde{\mathcal{T}}$ and consequently the concurrence relation $\|$ and the predecessor relation $\preceq = \widetilde{F}$ are well-defined in $\widetilde{\mathcal{N}}$.

Consider for $\widetilde{\mathcal{N}}^\theta$ the initial marking $\widetilde{M_0^\theta}$ given by the tokens from the untimed marking $\widetilde{M_0}$ and consider $\widetilde{M_0^\theta}$ produced at the time 0.

It is easy to see that by definition $\langle \widetilde{\mathcal{N}}, \widetilde{M_0} \rangle$ is 1-safe thus it results that $\langle \widetilde{\mathcal{N}}^\theta, \widetilde{M_0^\theta} \rangle$ is also 1-safe. Since any reachable marking $\widetilde{M}$ in $\langle \widetilde{\mathcal{N}}^\theta, \widetilde{M_0^\theta} \rangle$ contains at most a token in each place we can interpret $\widetilde{M}$ as the set of places that contain a token.

Denote by $\widetilde{\mathcal{L}}_{\widetilde{\mathcal{N}}^\theta}(\widetilde{M_0^\theta})$ the set of all complete time-traces that can be executed in $\langle \widetilde{\mathcal{N}}^\theta, \widetilde{M_0^\theta} \rangle$ where a complete trace means that all transitions in $\widetilde{\mathcal{N}}^\theta$ are executed. This is possible since $\widetilde{\mathcal{N}}$ is acyclic, conflict free, and 1-safe. Hence any complete time-trace τ^θ contains $\mid \widetilde{\mathcal{T}} \mid$ transitions since each transition $t \in \widetilde{\mathcal{T}}$ is executed exactely once.

For $\upsilon = 1, \ldots, \mid \widetilde{\mathcal{T}} \mid$ denote by θ_{t_υ} the time when transition t_υ is executed in $\langle \widetilde{\mathcal{N}}^\theta, \widetilde{M_0^\theta} \rangle$. $\widetilde{\Theta} = (\theta_{t_1}, \theta_{t_2}, \ldots, \theta_{t_{|\widetilde{\mathcal{T}}|}})$ is the valuation of the execution times for all the transitions in $\widetilde{\mathcal{T}}$.

The following system of inequalities represents the characteristic system of $\langle \widetilde{\mathcal{N}}^\theta, \widetilde{M_0^\theta} \rangle$.

$$\widetilde{K}(\langle\widetilde{\mathcal{N}}^{\theta},\widetilde{M}_0^{\theta}\rangle)=\begin{cases}\mathtt{max}_{t_\upsilon\in{}^{\bullet\bullet}t_1}(\theta_{t_\upsilon})+L^s_{t_1} & \leq \quad \theta_{t_1} \quad \leq \quad \mathtt{max}_{t_\upsilon\in{}^{\bullet\bullet}t_1}(\theta_{t_\upsilon})+U^s_{t_1}\\ \mathtt{max}_{t_\upsilon\in{}^{\bullet\bullet}t_2}(\theta_{t_\upsilon})+L^s_{t_2} & \leq \quad \theta_{t_2} \quad \leq \quad \mathtt{max}_{t_\upsilon\in{}^{\bullet\bullet}t_2}(\theta_{t_\upsilon})+U^s_{t_2}\\ \quad\vdots & \\ \mathtt{max}_{t_\upsilon\in{}^{\bullet\bullet}t_{|\widetilde{\mathcal{T}}|}}(\theta_{t_\upsilon})+L^s_{t_{|\widetilde{\mathcal{N}}|}} & \leq\theta_{t_{|\widetilde{\mathcal{N}}|}}\leq\mathtt{max}_{t_\upsilon\in{}^{\bullet\bullet}t_{|\widetilde{\mathcal{T}}|}}(\theta_{t_\upsilon})+U^s_{t_{|\widetilde{\mathcal{T}}|}}\end{cases} \tag{5.21}$$

where in 5.21 ${}^{\bullet\bullet}t=\emptyset$ implies $\mathtt{max}_{t_\upsilon\in{}^{\bullet\bullet}t}(\theta_{t_\upsilon})=0$.

Denote by $Sol(\widetilde{K})$ the set of all solutions of $\widetilde{K}(\langle\widetilde{\mathcal{N}}^{\theta},\widetilde{M}_0^{\theta}\rangle)$:

$$Sol(\widetilde{K})=\left\{\widetilde{\Theta}\mid\widetilde{\Theta}\text{ is a solution of }\widetilde{K}(\langle\widetilde{\mathcal{N}},\widetilde{M}_0\rangle)\right\} \tag{5.22}$$

If $\widetilde{\Theta}$ is a solution of the characteristic system $\widetilde{K}(\langle\widetilde{\mathcal{N}}^{\theta},\widetilde{M}_0^{\theta}\rangle)$ then there exists a complete time-trace $\widetilde{\tau}^{\theta}\in\widetilde{\mathcal{L}}^{\theta}(\widetilde{M}_0^{\theta})$ s.t. the execution times of the transitions in $\widetilde{\tau}^{\theta}$ are given by Θ. Basically $\widetilde{\tau}^{\theta}$ is obtained by ordering the execution times $\theta_{t_1},\theta_{t_2},\ldots,\theta_{t_{|\widetilde{\mathcal{T}}|}}$ s.t. t_ι appears in τ^{θ} before t_λ if: *i*) $\theta_{t_\iota}\leq\theta_{t_\lambda}$ and *ii*) if $\theta_{t_\iota}=\theta_{t_\lambda}$ then t_ι is not a successor of t_λ (i.e. $t_\iota\|t_\lambda$ or $t_\iota\preceq t_\lambda$ where $\preceq$ is the partial order relation defined by $\widetilde{F}$).

Denote by $\widetilde{I}(t_\upsilon)=Proj_\upsilon(Sol(\widetilde{K}))$ the projection of the solution set $Sol(\widetilde{K})$ on the time-axis υ that corresponds with the execution time θ_{t_υ} of transition t_υ. In other words $\widetilde{I}(t_\upsilon)=[L(t_\upsilon),U(t_\upsilon)]$ is the time interval that gives the earliest global time $L(t_\upsilon)$ respectively the latest global time $U(t_\upsilon)$ during which transition t_υ can be executed in $\langle\widetilde{\mathcal{N}}^{\theta},\widetilde{M}_0^{\theta}\rangle$. Denote by $\widetilde{\mathbf{I}}=\widetilde{I}(t_1)\times\widetilde{I}(t_2)\times\ldots\times\widetilde{I}(t_{|\widetilde{\mathcal{T}}|})$ the smallest $\mid\widetilde{\mathcal{T}}\mid$-hyperbox that includes $Sol(\widetilde{K})$.

In general given a sub-set $Sol_\nu(\widetilde{K})$ of the solution set $Sol(\widetilde{K})$ denote $\widetilde{\mathbf{I}}_\nu$ the smallest $\mid\widetilde{\mathcal{T}}\mid$-hyperbox that includes $Sol_\nu(\widetilde{K})$.

Denote by $[t]^{\downarrow}$ the set of predecessor transitions of a transition $t\in\widetilde{\mathcal{N}}$:

$$[t]^{\downarrow}=\left\{t'\in\widetilde{\mathcal{T}}:t'\preceq t\text{ and }t'\neq t\right\}$$

Denote by $\mathcal{TCUT}(\widetilde{\mathcal{N}})$ the set of all the transition-cuts η where a transition-cut is a set of concurrent transitions in $\widetilde{\mathcal{N}}$ that is maximal w.r.t. set of inclusion. $[\eta]^{\downarrow}=\bigcup_{t_\iota\in\eta}[t_\iota]^{\downarrow}$ is the set of all the predecessor transitions of the transitions that are included in the transition-cut η.

Consider in the following a sub-set $Sol_\nu(\widetilde{K})$ of the solution set ($Sol_\nu(\widetilde{K})\subseteq Sol(\widetilde{K})$).

Let $\mathtt{K}_\eta$ be the set of indexes that correspond to the set of transitions in $[\eta]^{\downarrow}$ and then denote $Sol_\nu(\widetilde{K})\mid_{\mathtt{K}_\eta}$ the projection of $Sol_\nu(\widetilde{K})$ on to the $\mathtt{K}_\eta$ hyperplane that corresponds with the transitions in $\mathtt{K}_\eta$.

$Sol_\nu(\widetilde{K}) \ |_\eta$ is the projection of $Sol_\nu(\widetilde{K})$ on the hyper-plane that corresponds with the axes of the transitions contained in η.

Given $\Theta_{\mathtt{K}_\eta} \in Sol_\nu(\widetilde{K}) \ |_{\mathtt{K}_\eta}$ denote $Sol_\nu(\widetilde{K}) \ |_\eta^{\Theta_{\mathtt{K}_\eta}}$ the set of $| \ \eta \ | -points$ in $Sol_\nu(\widetilde{K}) \ |_\eta$ that correspond to Θ_K. For $\Theta_{\mathtt{K}_\eta} \in Sol_\nu(\widetilde{K}) \ |_{\mathtt{K}_\eta}$ denote by $\theta_{t_\upsilon}^{en}(\Theta_{\mathtt{K}_\eta})$ the enabling time of transition $t \in \eta$ given $\Theta_{\mathtt{K}_\eta}$:

$$\theta_t^{en}(\Theta_{\mathtt{K}_\eta}) = \mathtt{max}_{t' \in {}^{\bullet\bullet}t}(\theta_{t'}(\Theta_{\mathtt{K}_\eta}))$$

Denote $I(t \mid \Theta_{\mathtt{K}_\eta})$ the execution time-interval of a transition $t \in \eta$ given $\Theta_{\mathtt{K}}$:

$$I_\nu(t \mid \Theta_{\mathtt{K}_\eta}) = [\mathtt{max}(\theta_t^{en}(\Theta_{\mathtt{K}_\eta}) + L_t^s, L_\nu(t)), \mathtt{min}(\theta_t^{en}(\Theta_{\mathtt{K}_\eta}) + U_t^s, U_\nu(t))]$$

Definition 58 (Time Independence). *Consider a sub-set $Sol_\nu(\widetilde{K})$ of the solution set $Sol(\widetilde{K})$ and a transition-cut $\eta \in \mathcal{TCUT}(\widetilde{\mathcal{N}})$, $\eta = (t_1, \ldots, t_{|\eta|})$. The transitions in η are time-independent in $Sol_\nu(\widetilde{K})$ if:*

$$\forall \Theta_{\mathtt{K}_\eta} \in Sol_\nu(\widetilde{K}) \ |_{\mathtt{K}_\eta} \qquad Sol_\nu(\widetilde{K}) \ |_\eta^{\Theta_{\mathtt{K}_\eta}} = I(t_1 \mid \Theta_{\mathtt{K}_\eta}) \times \ldots \times I(t_{|\eta|} \mid \Theta_{\mathtt{K}_\eta}) \quad (5.23)$$

If $\forall \eta \in \mathcal{TCUT}(\widetilde{\mathcal{N}})$ we have that the transitions in η are time-independent then the sub-set $Sol_\nu(\widetilde{K})$ of the solution set has the time-independence property for the concurrent transitions.

In words we say that $Sol_\nu(\widetilde{K})$ has the time-independence property for the concurrent transitions if for any maximal set of concurrent transitions say $t_1, \ldots, t_{|\eta|}$ we have that:

if $t_1, \ldots, t_{|\eta|}$ can become enabled at the time $\theta_{t_1}^{en}, \ldots, \theta_{t_{|\eta|}}^{en}$

then their firing times $\theta_{t_1}, \ldots, \theta_{t_{|\eta|}}$ belong to an $| \ \eta \ |$-rectangle

Notice that $Sol(\widetilde{K})$ has the time-independence property for the concurrent transitions since $\widetilde{\mathcal{N}}$ is free-choice.

Consider below a temporal constraint κ_{t_ℓ} expressed as a linear inequality on some variable θ_{t_ℓ} ($t_\ell \in \widetilde{\mathcal{T}}$) (e.g. $\kappa_{t_\ell} := \{L'(t_\ell) \leq \theta_{t_\ell} \leq U'(t_\ell)\}$) that is imposed to $\langle \widetilde{\mathcal{N}}^\theta, \widetilde{M_0^\theta} \rangle$. The temporal constraint κ_{t_ℓ} may be intuitively understood as the observation of execution time of transition t_ℓ.

The plant behavior constrained by the temporal constraint κ_{t_ℓ} is given by $Sol(\widetilde{K} \wedge \kappa_{t_\ell})$ (where κ_{t_ℓ} is added to $\widetilde{K}$) and in general we have that $Sol(\widetilde{K} \wedge \kappa_{t_\ell})$ does not satisfy the time-independence property even if $Sol(\widetilde{K})$ satisfies time independence. Consequently we cannot use the smallest $| \ \widetilde{\mathcal{T}} \ |$-hyperbox $I(\kappa_{t_\ell})$ that includes $Sol(\widetilde{K} \wedge \kappa_{t_\ell})$ as an approximation for $Sol(\widetilde{K} \wedge \kappa_{t_\ell})$ since the concurrent transitions are not time-independent.

Intuitively to restore time-independence property we should find e.g. two temporal constraints κ_1 and κ_2 such that $\widetilde{K} \wedge \kappa_{t_\ell}$ can be rewritten as:

$$\widetilde{K} \wedge \kappa_{t_\ell} \Leftrightarrow (\widetilde{K} \wedge \kappa_1) \vee (\widetilde{K} \wedge \kappa_2)$$

such that $Sol(\widetilde{K} \wedge \kappa_1)$ and $Sol(\widetilde{K} \wedge \kappa_2)$ satisfy the time independence property and:

$$Sol(\widetilde{K} \wedge \kappa) = Sol(\widetilde{K} \wedge \kappa_1) \cup Sol(\widetilde{K} \wedge \kappa_2).$$

We need the time-independence property for the following reasons:

- Two transitions are concurrent whenever they can be executed in any order provided they are enabled. For TPN models it may be the case that two transitions are concurrent in the untimed case but their execution times are correlated in order to satisfy a certain temporal constraint.
- A configuration C in the untimed case is extended by appending the events that are enabled in $CUT(C)$. If the concurrent transitions are time independent it means that we can easily extend a time configuration since the tokens in $CUT(C)$ are produced independently.
- In a distributed setting this property will allow to exchange temporal-information only for the border-conditions of two local time configurations that are checked for consistency.

Consider for a transition $t_\ell \in \widetilde{\mathcal{T}}$ a temporal constraint on θ_{t_ℓ} given as:

$$\kappa_{t_\ell} \overset{def}{:=} \{L'(t_\ell) \leq \theta_{t_\ell} \leq U'(t_\ell) \ \wedge \ I'(t_\ell) \subset I(t_\ell)\}$$

where $I'(t_\ell) = [L'(t_\ell), U'(t_\ell)]$.

We have the set of solutions that obey the temporal constraint given as the solution set of the characteristic systems $\widetilde{K}' = \widetilde{K} \wedge \kappa_{t_\ell}$:

$$\widetilde{K} \wedge \kappa_{t_\ell} = \begin{cases} \mathtt{max}_{t_\gamma \in {}^{\bullet\bullet}t_\iota}(\theta_{t_\gamma}) + L^s_{t_\iota} \leq \theta_{t_\iota} \leq \mathtt{max}_{t_\gamma \in {}^{\bullet\bullet}t_\iota}(\theta_{t_\gamma}) + U^s_{t_\iota} \\ \hfill \text{for } \iota = 1, \ldots, \mid \widetilde{\mathcal{T}} \mid \\ L'(t_\ell) \leq \theta_{t_\ell} \leq U'(t_\ell) \end{cases} \tag{5.24}$$

In the following we present an algorithm that given $\langle \widetilde{\mathcal{N}}^\theta, M_0^\theta \rangle$ and a constraint κ_{t_ℓ} derives a set of $\mid \widetilde{\mathcal{T}} \mid$-hyperboxes $\Gamma(\kappa_{t_\ell}) = \left\{ \widetilde{\mathbf{I}}_\nu(\kappa_{t_\ell}) \mid \nu \in \mathcal{V}(\kappa_{t_\ell}) \right\}$ such that for $Sol_\nu(\widetilde{K} \wedge \kappa_{t_\ell}) \overset{def}{=} Sol(\widetilde{K} \wedge \kappa_{t_\ell}) \cap \widetilde{\mathbf{I}}_\nu(\kappa_{t_\ell})$ we have that:

1. $\bigcup_{\nu \in \mathcal{V}(\kappa_{t_\ell})} Sol_\nu(\widetilde{K} \wedge \kappa_{t_\ell}) = Sol(\widetilde{K} \wedge \kappa_{t_\ell})$
2. $\forall \nu \in \mathcal{V}(\kappa_{t_\ell}) \Rightarrow Sol_\nu(\widetilde{K} \wedge \kappa_{t_\ell})$ has the time independence property

The idea behind developing the algorithm for deriving $\Gamma(\kappa_{t_\ell})$ is that the constraint κ_{t_ℓ} applied to $\widetilde{I}(t_\ell)$ is translated into a disjunction of conjunctions of temporal constraints expressed over the input transition and output transitions of t_ℓ:

$$\kappa_{t_\ell} \Leftrightarrow \bigvee_{\nu_{t_\ell} \in \mathcal{V}_{t_\ell}} \mathcal{K}_{\nu_{t_\ell}} \quad \text{with} \quad \mathcal{K}_{\nu_{t_\ell}} = \bigwedge_{t_\iota \in Prox'_{t_\ell}} \kappa_{t_\iota}$$

where $Prox'_{t_\ell}$ is a subset of the set of input and output transitions of t_ℓ ($Prox'_{t_\ell} \subseteq Prox_{t_\ell}$ and $Prox_{t_\ell} = {}^{\bullet\bullet}t_\ell \cup t_\ell^{\bullet\bullet}$).

The maximum number of conjunctions of the form $\mathcal{K}_{\nu_{t_\ell}}$ is given by the maximum number of concurrent predecessor transitions of t_ℓ. Each conjunction of temporal constraints $\mathcal{K}_{\nu_{t_\ell}}$ is partitioned in a conjunction of constraints that are propagated *"forward"* from t_ℓ (to its output transitions) and a conjunction of constraints that are propagated *"backwards"* from t_ℓ (to input transitions of t_ℓ):

$$\mathcal{K}_{\nu_{t_\ell}} = \mathcal{K}^f_{\nu_{t_\ell}} \wedge \mathcal{K}^b_{\nu_{t_\ell}}$$

where $\mathcal{K}^f_{\nu_{t_\ell}} = \bigwedge_{t_\iota \in t_\ell^{\bullet\bullet}} \kappa_{t_\iota}$ and $\mathcal{K}^b_{\nu_{t_\ell}} = \bigwedge_{t_\iota \in {}^{\bullet\bullet}t_\ell} \kappa_{t_\iota}$.

A constraint $\kappa_{t_\ell} = I'(t_\ell)/I(t_\ell)$ propagates backwards in the following way:

1. the new upper limits of the input transitions of t_ℓ are:

$$U'(t_\iota) = \mathtt{max}\left\{(U'(t_\ell) - L^s_{t_\ell}), U(t_\iota)\right\} \text{ for } t_\iota \in {}^{\bullet\bullet}t_\ell$$

2. $Cases(t_\ell)$ is the set of input transitions of t_ℓ ($t_\iota \in {}^{\bullet\bullet}t_\ell$) s.t. we have $L'(t_\ell) - U^s_{t_\ell} > L(t_\iota)$:

$$Cases(t_\iota) = \left\{t_\iota \mid t_\iota \in {}^{\bullet\bullet}t_\ell \wedge (L(t_\iota) < L'(t_\ell) - U^s_{t_\ell})\right\}$$

3. then for each element t_ι of $Cases(t_\ell)$ we have a case (that corresponds to a conjunction of constraints $\mathcal{K}_{\nu_{t_\ell}}$) where:
   (a) the new lower limit for t_ι is $L'(t_\iota) = L(t_\ell) - U^s_{t_\ell}$
   (b) while all the other input transitions of t_ℓ keep their lower limits

The forward propagation of κ_{t_ℓ} to its successor transitions $t_\iota \in t_\ell^{\bullet\bullet}$ is straightforward by imposing the new lower and upper limits: $L'(t_\iota) = \mathtt{max}_{t_\gamma \in {}^{\bullet\bullet}t_\iota}(L'(t_\gamma)) + L^s_{t_\iota}$ and $U'(t_\iota) = \mathtt{max}_{t_\gamma \in {}^{\bullet\bullet}t_\iota}(U'(t_\gamma)) + U^s_{t_\iota}$

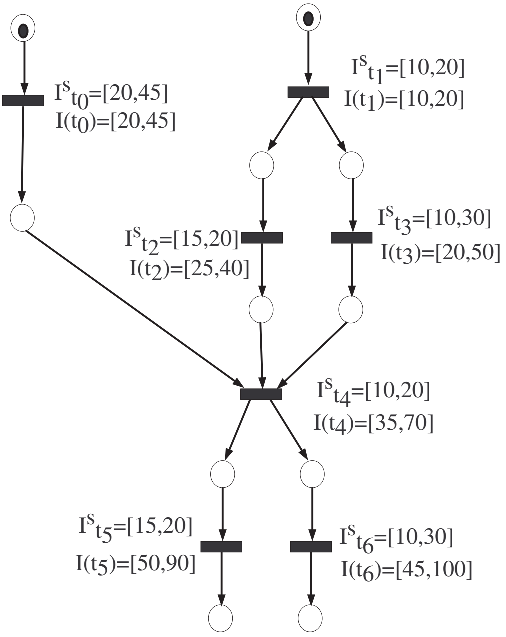


Figure 5.6:

Example 29. *Consider the TPN* $\langle \widetilde{\mathcal{N}}^{\theta}, \widetilde{M_0^{\theta}} \rangle$ *displayed in Fig. 5.6 where the static firing interval* $I^s_{t_\ell}$ *is displayed next to each transition* t_ℓ*,* $\ell = 1, \ldots, 6$ *as well as its execution time-interval* $I(t_\ell)$*.*

Assume that the execution of transition t_4 *was observed at the time* $\theta_{t_4} = 53$*. The characteristic system constrained by* $\kappa_{t_4} := \{\theta_{t_4} = 53\}$ *is:*

$$\widetilde{K} \wedge \kappa_{t_5} = \begin{cases} 20 \leq \theta_{t_0} \leq 45 \\ 10 \leq \theta_{t_1} \leq 20 \\ \theta_{t_2} + 15 \leq \theta_{t_2} \leq \theta_{t_2} + 20 \\ \theta_{t_2} + 10 \leq \theta_{t_3} \leq \theta_{t_2} + 30 \\ \mathtt{max}(\theta_{t_0}, \theta_{t_1}, \theta_2) + 10 \leq \theta_{t_4} \leq \mathtt{max}(\theta_{t_0}, \theta_{t_1}, \theta_{t_2}) + 20 \\ \theta_{t_5} + 15 \leq \theta_{t_5} \leq \theta_{t_5} + 20 \\ \theta_{t_5} + 10 \leq \theta_{t_6} \leq \theta_{t_5} + 30 \\ \theta_{t_4} = 53 \end{cases} \tag{5.25}$$

After simple calculation one can derive that the smallest 7*-hyperbox that includes* $Sol(\widetilde{K} \wedge \kappa_{t_4})$ *is:* $I(t_1) = [20, 45]$*,*$I(t_1) = [10, 20]$*,* $I(t_2) = [25, 40]$*,* $I(t_3) = [20, 43]$*,* $I(t_4) = 53$*,* $I(t_5) = [68, 73]$*,* $I(t_6) = [63, 93]$*.*

First we check if $Sol(\widetilde{K} \wedge \kappa_{t_4})$ has the property of time-independence of the concurrent transitions: Consider for instance $\eta = \{t_0, t_2, t_3\}$. We have that t_0,t_2 and t_3 have only a predecessor event that is $[t_0, t_2, t_3]^{\downarrow} = \{t_1\}$ thus $\mathtt{K}_\eta = \{t_1\}$.

Let $\theta_{t_1} = \Theta_{\mathtt{K}_\eta} = 10$. We have that $I(t_0 \mid \Theta_{\mathtt{K}_\eta}) = [20, 45]$, $I(t_2 \mid \Theta_{\mathtt{K}_\eta}) = [25, 30]$ and $I(t_3 \mid \Theta_{\mathtt{K}_\eta} = [20, 40]$. The time independence property requires that $I(t_0 \mid \Theta_{\mathtt{K}_\eta}) \times I(t_2 \mid \Theta_{\mathtt{K}_\eta}) \times I(t_3 \mid \Theta_{\mathtt{K}_\eta})$ is the projection of $Sol(\widetilde{K} \wedge \kappa_{t_4})$ on to the hyper-plane $\eta = (\theta_{t_0}, \theta_{t_1}, \theta_{t_2})$. To see that the time independence property is not satisfied for $Sol(\widetilde{K} \wedge \kappa_{t_4})$ consider $\theta_{t_0} = 20$, $\theta_{t_2} = 25$ and $\theta_{t_3} = 20$; we have that t_4 cannot be executed at the time 53.

Thus we should look for a cover $\left\{Sol_\nu(\widetilde{K} \wedge \kappa_{t_4}) \mid \nu \in \mathcal{V}\right\}$ of $Sol(\widetilde{K} \wedge \kappa_{t_4})$ s.t. $\forall Sol_\nu(\widetilde{K} \wedge \kappa_{t_4})$ $\nu \in \mathcal{V}$, we have that the time independence property is satisfied in $Sol_\nu(\widetilde{K} \wedge \kappa_{t_4})$.

First we should examine how κ_{t_4} constrains its input and output transitions. Constraint κ_{t_4} propagates forward on to its output transitions t_5 and t_6. Thus $\theta_{t_4} = 53$ implies $\theta_{t_5} \in [68, 73]$ and $\theta_{t_6} \in [63, 93]$. κ_{t_4} also propagates backwards on to its input transitions t_0, t_2 and t_3 as follows. Since t_4 is executed at the time 53 we have that t_4 becomes enabled in the time interval $[33, 43]$ that means that:

1. *neither θ_0 nor θ_2 nor θ_3 can be executed after the time 43*
2. *at least one input transition of t_4 is executed after 33 that is: $(\theta_{t_0} \geq 33)$ or $(\theta_{t_2} \geq 33)$ or $(\theta_{t_{33}} \geq 33)$*

We have three cases that correspond with splitting $Sol(\widetilde{K} \wedge \kappa_{t_4})$ on to $Sol_1(\widetilde{K} \wedge \kappa_{t_4})$, $Sol_2(\widetilde{K} \wedge \kappa_{t_4})$, and $Sol_3(\widetilde{K} \wedge \kappa_{t_4})$ that also correspond with re-writing κ_{t_4} as $\kappa_{t_4} = \mathcal{K}1_{t_4} \vee \mathcal{K}2_{t_4} \wedge \mathcal{K}3_{t_4}$ where:

1. $\mathcal{K}1_{t_4} = \kappa^1_{t_0} \wedge \kappa^1_{t_2} \wedge \kappa^1_{t_3} \wedge \kappa^1_{t_5} \wedge \kappa^1_{t_6}$
   - $\kappa^1_{t_0} = \{33 \leq \theta_{t_0} \leq 43\}$;
   - $\kappa^1_{t_2} = \{25 \leq \theta_{t_2} \leq 40\}$;
   - $\kappa^1_{t_3} = \{20 \leq \theta_{t_3} \leq 43\}$;
   - $\kappa^1_{t_5} = \{68 \leq \theta_{t_5} \leq 73\}$;
   - $\kappa^1_{t_6} = \{73 \leq \theta_{t_5} \leq 93\}$
2. $\mathcal{K}2_{t_4} = \kappa^2_{t_0} \wedge \kappa^2_{t_2} \wedge \kappa^2_{t_3} \wedge \kappa^2_{t_5} \wedge \kappa^2_{t_6}$
   - $\kappa^2_{t_0} = \{20 \leq \theta_{t_0} \leq 43\}$;
   - $\kappa^2_{t_2} = \{33 \leq \theta_{t_2} \leq 40\}$;
   - $\kappa^2_{t_3} = \{20 \leq \theta_{t_3} \leq 43\}$;
   - $\kappa^2_{t_5} = \{68 \leq \theta_{t_5} \leq 73\}$;
   - $\kappa^2_{t_6} = \{73 \leq \theta_{t_5} \leq 93\}$

3. $\mathcal{K}_{3_{t_4}} = \kappa^3_{t_0} \wedge \kappa^3_{t_2} \wedge \kappa^3_{t_3} \wedge \kappa^3_{t_5} \wedge \kappa^3_{t_6}$

- $\kappa^3_{t_0} = \{20 \leq \theta_{t_0} \leq 43\}$;
- $\kappa^3_{t_2} = \{25 \leq \theta_{t_2} \leq 40\}$;
- $\kappa^3_{t_3} = \{33 \leq \theta_{t_3} \leq 43\}$;
- $\kappa^3_{t_5} = \{68 \leq \theta_{t_5} \leq 73\}$;
- $\kappa^3_{t_6} = \{73 \leq \theta_{t_5} \leq 93\}$

For case 2 we have that $\kappa^2_{t_2} = \{33 \leq \theta_{t_2} \leq 40\}$ *propagates backwards onto* t_1 *and then we obtain* $\kappa^2_{t_1} = \{13 \leq \theta_{t_2} \leq 20\}$

Thus we obtain:

$\mathbf{I}_1(\kappa_{t_5})$*:* $I_0(t_1) = [33, 43]$, $I_1(t_1) = [10, 20]$, $I_1(t_2) = [25, 40]$, $I_1(t_3) = [20, 43]$, $I_1(t_4) = [53, 53]$, $I_1(t_5) = [68, 73]$, $I_1(t_6) = [63, 93]$

$\mathbf{I}_2(\kappa_{t_5})$*:* $I_0(t_1) = [20, 43]$, $I_2(t_1) = [13, 20]$, $I_2(t_2) = [33, 40]$, $I_2(t_3) = [20, 43]$, $I_2(t_4) = [53, 53]$, $I_2(t_5) = [68, 73]$, $I_2(t_6) = [63, 93]$

$\mathbf{I}_3(\kappa_{t_5})$*:* $I_0(t_1) = [20, 43]$, $I_3(t_1) = [10, 20]$, $I_3(t_2) = [25, 40]$, $I_3(t_3) = [33, 43]$, $I_3(t_4) = [53, 53]$, $I_3(t_5) = [68, 73]$, $I_3(t_6) = [63, 93]$

Algorithm 11 (below) imposes a given constraint κ_{t_ℓ} as follows:

- first we calculate the execution time intervals for the unconstrained behaviour of the TPN $\langle \widetilde{\mathcal{N}}^\theta, \widetilde{M}^\theta_0 \rangle$
- then for a constraint κ_{t_ℓ} applied to the execution time interval of transition t_ℓ
  - (a) first the upper limits of the predecessor transitions are recalculated (Algorithm 12)
  - (b) and then the lower limits of the predecessor transitions are recalculated (Algorithm 13) but this requires to split up analysis considering different cases.

We consider the following partitions of the set of transitions:

$\widetilde{\mathcal{T}}$ is partitioned in disjunct subsets (layers) from up to down as follows:

- $Layer^{\downarrow}[0] = \left\{ t_\iota \in \widetilde{\mathcal{T}} \mid {}^{\bullet\bullet}t_\iota = \emptyset \right\}$

- $Layer^{\downarrow}[y] = \left\{ t_\iota \in \widetilde{\mathcal{T}} \mid ({}^{\bullet\bullet}t_\iota \subseteq \bigcup_{z=0}^{y-1} Layer^{\downarrow}[z]) \wedge ({}^{\bullet\bullet}t_\iota \cap Layer^{\downarrow}[y-1] \neq \emptyset) \right\}$

Then given a transition $t_\ell \in \widetilde{\mathcal{T}}$ consider the set of its predecessor transitions $[t^{\downarrow}_\ell]$ partitioned in disjunct subsets (layers) as follows:

- $Layer^{\uparrow}_{t_\ell}[0] = \{t_\ell\}$

- $Layer^{\uparrow}_{t_\ell}[w+1] = \left\{t_\upsilon \in [t_\ell^{\downarrow}] \mid t_\upsilon^{\bullet\bullet} \subseteq \bigcup_{z=0}^{w} Layer^{\uparrow}_{t_\ell}[z] \wedge t_\upsilon^{\bullet\bullet} \cap Layer^{\uparrow}_{t_\ell}[w] \neq \emptyset\right\}$

Algorithm 11 Algorithm to impose a temporal constraint

```
Require: $\langle \widetilde{\mathcal{N}}^\theta, \widetilde{M}_0^\theta \rangle$, $\kappa_{t_\ell}$
Ensure: $\Gamma(\kappa_{t_\ell})$
for all $t_\iota \in Layer^{\downarrow}[0]$ do
    $I(t_\iota) = I^s_{t_\iota}$
end for
for $y = 1$ to $y_{max}$ do
    for all $t_\iota \in Layer^{\downarrow}[y]$ do
        $L(t_\iota) = \mathtt{max}_{t_\lambda \in {}^{\bullet\bullet}t_\iota}(L(t_\lambda)) + L^s_{t_\iota}$
        $U(t_\iota) = \mathtt{max}_{t_\lambda \in {}^{\bullet\bullet}t_\iota}(U(t_\lambda)) + U^s_{t_\iota}$
        $I(t_\iota) = [L(t_\iota), U(t_\iota)]$
    end for
end for
Propagate_constraint_U($\kappa_{t_\ell}$)
$\nu = \nu_{max} = 1$; $\mathbf{I}_\nu = \mathbf{I}$;
Propagate_constraint_L($\kappa_{t_\ell}$)
$\Gamma(\kappa_{t_l}) = \{\mathbf{I}_\nu \mid \nu = 1, \ldots, \nu_{max}\}$
```

Algorithm 12 Propagate_constraint_U

```
Require: $\kappa_{t_\ell}$
Ensure: $\mathbf{I}$
$U'(t_\ell) = \kappa_{t_\ell}(U)$
for $w = 0$ to $w_{max}$ do
    for all $t_\iota \in Layer^{\uparrow}_{t_\ell}[w]$ do
        $U'(t_\iota) = \mathtt{max}(U(t_\iota), \mathtt{max}_{t_\lambda \in t_\iota^{\bullet\bullet}}(U'(t_\lambda)) - L^s_{t_\iota})$
    end for
end for
```

Algorithm 13 Propagate_constraint_L_backwards

```
Require: $\kappa_{t_\ell}$
Ensure: $\Gamma(\kappa_{t_\ell})$
$Cases(Layer^{\uparrow}_{t_\ell}[0]) = \{\kappa_{t_\ell}(L)\}$
for $w = 0$ to $w_{max}$ do
  for all $\kappa_{t_\iota}(L) \in Cases(Layer^{\uparrow}_{t_\iota}[w])$ do
    for all $t_\lambda \in {}^{\bullet\bullet}t_\iota \cap Layer^{\uparrow}_{t_\ell}[w+1]$ do
      new_constraint=false
      if $L'(t_\iota) - U^s_{t_\iota} > L(t_\lambda)$ then
        $\kappa_{t_\lambda} := \left\{L'(t_\lambda) = L'(t_\iota) - U^s_{t_\iota}\right\}$
        add $(\kappa_{t_\lambda})$ to $Cases(Layer^{\uparrow}_{t_\ell}[w+1]$
        new_const=true
      end if
    end for
    if new_constraint=false then
      add $\mathbf{I}_\iota$ to $\Gamma(\kappa_{t_\ell})$
    end if
  end for
end for
for all $\nu \in \mathcal{V}_\ell$ do
  for all $y = 0$ to $y_{max}$ do
    for all $t_\upsilon \in Layer^{\downarrow}[y]$ do
      $L'_\nu(t_\upsilon) = \mathtt{max}(L'_\nu(t_\upsilon), \mathtt{max}_{t_\gamma \in {}^{\bullet\bullet}t_\upsilon}(L_\nu(t_\gamma) + L^s_{t_\upsilon}))$
      $U'_\nu(t_\upsilon) = \mathtt{max}(U'_\nu(t_\upsilon), \mathtt{max}_{t_\gamma \in {}^{\bullet\bullet}t_\upsilon}(U_\nu(t_\gamma) + U^s_{t_\upsilon}))$
    end for
  end for
end for
```

The maximum number of cases that can be obtained is bounded by the maximum number of transitions in a transition-cut in $\widetilde{\mathcal{N}}$. The complexity of the algorithm is $O(N^2)$ where N is the number of transitions in $\widetilde{\mathcal{N}}$.

Example 30. *Consider the TPN $\langle\widetilde{\mathcal{N}^\theta}, \widetilde{M^\theta_0}\rangle$ where the static firing intervals are displayed attached to each transition. Then we have* $\mathbf{I} = I(t_1) \times I(t_2) \times I(t_3) \times I(t_4) \times I(t_5) \times I(t_6)$ *where:* $I(t_1) = [10, 25]$; $I(t_2) = [50, 80]$; $I(t_2) = [40, 90]$; $I(t_4) = [90, 120]$; $I(t_5) = [110, 170]$; $I(t_6) = [130, 210]$.

In Fig. 5.8 we have displayed the projection of $Sol(\widetilde{K})$ on to different axes. As one can see the polygon that represents the projection of $Sol(\widetilde{K})$ on to the plane of $(\theta_{t_\iota}, \theta_{t_\lambda})$ has edges either $90°$ *or* $45°$.

Now consider that we are interested in deriving the behavior of $\langle\widetilde{\mathcal{N}^\theta}, \widetilde{M^\theta_0}\rangle$ such that t_4 is executed in the interval $\theta_{t_4} \in [100, 120]$. *Notice that this problem may arise in the case when the temporal constraint on t_4 is the result of an observation.*

Let $\kappa_{t_4} := 100 \le \theta_{t_4} \le 120$ *be the temporal constraint. We show how to derive a*

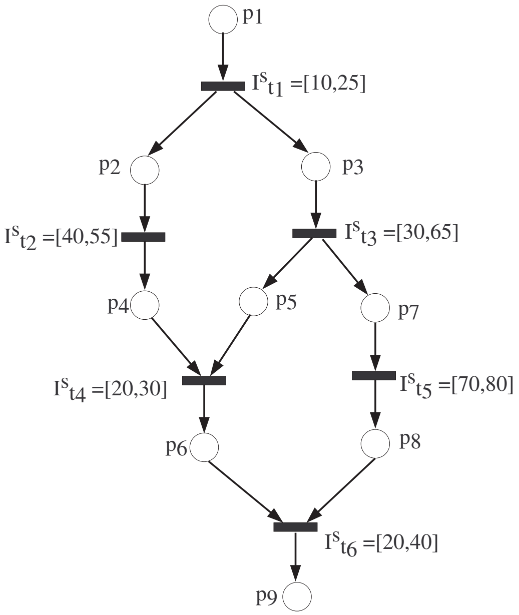


Figure 5.7:

set of $|\widetilde{\mathcal{T}}|$*-hyperboxes* $\Gamma(\kappa_{t_4})$ *s.t.* $\forall \mathbf{I}_\nu(\kappa_{t_4}) \in \Gamma(\kappa_{t_4})$ *we have that* $Sol_\nu(\widetilde{K} \wedge \kappa_{t_4}) = Sol(\widetilde{K} \wedge \kappa_{t_4}) \cap I_\nu(\kappa_{t_4})$ *satisfies the time independence property.*

Algorithm 11 works for κ_{t_4} *as follows:*

- $I'(t_4) = [100, 120] \Rightarrow \kappa_{t_4} \xrightarrow{b} \kappa_{t_2} \vee \kappa_{t_3}$ *where* $\kappa_{t_2} := \{70 \leq \theta_{t_2} \leq 80\}$ *and* $\kappa_{t_3} := \{70 \leq \theta_{t_3} \leq 90\}$.

- *for case 1:* $\mathcal{K}_1^b = \kappa_{t_2}$ *we have that* $\kappa_{t_2}[1] \xrightarrow{b} \kappa_{t_1}[1]$ *where* $\kappa_{t_1}[1] := \{15 \leq \theta_{t_1} \leq 25\}$. *Then we have* $\kappa_{t_1}[1] \xrightarrow{f} \kappa_{t_3}[1]$ *where* $\kappa_{t_3}[1] := \{45 \leq \theta_{t_3} \leq 90\}$ *and then* $\kappa_{t_3}[1] \xrightarrow{f} \kappa_{t_5}[1]$ *with* $\kappa_{t_5}[1] := \{115 \leq \theta_5 \leq 170\}$ *and* $\kappa_{t_5}[1] \xrightarrow{f} \kappa_{t_6}[1]$ *with* $\kappa_{t_6}[1] := \{135 \leq \theta_6 \leq 170\}$

- *for case 2:* $\mathcal{K}_2^b = \kappa_{t_3}$ *we have that* $\kappa_{t_3}[2] \xrightarrow{f} \kappa_{t_5}[2]$ *where* $\kappa_{t_5}[2] := \{140 \leq \theta_{t_5} \leq 170\}$ *and then we have* $\kappa_{t_5}[2] \xrightarrow{f} \kappa_{t_6}[2] := \{160 \leq \theta_{t_6} \leq 210\}$

Thus for $\nu = 1, 2$ *we have* $\mathbf{I}_\nu(\kappa_{t_4}) = I_\nu(t_1) \times I_\nu(t_2) \times I_\nu(t_3) \times I_\nu(t_4) \times I_\nu(t_5) \times I_\nu(t_6)$ *where respectively:*

1. $I_1(t_1) = [15, 25]$; $I_1(t_2) = [70, 80]$; $I_1(t_3) = [45, 90]$; $I_1(t_4) = [100, 120]$; $I_1(t_5) = [115, 170]$; $I_1(t_6) = [135, 210]$;

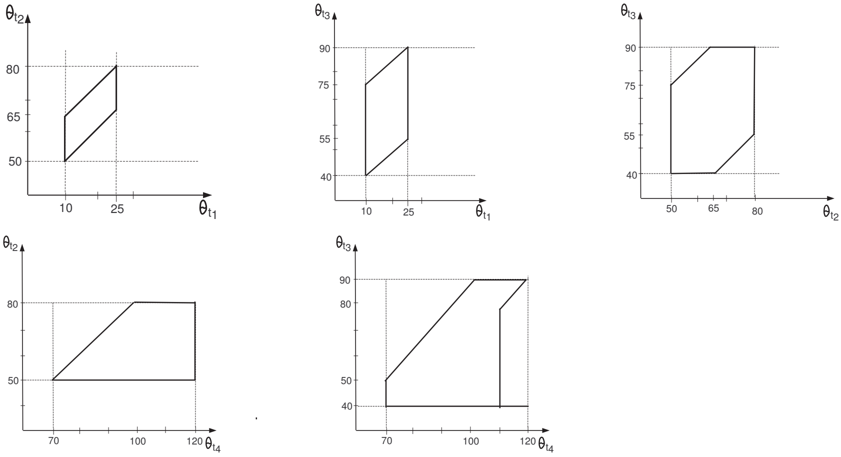


Figure 5.8:

2. $I_2(t_1) = [10, 25]$; $I_2(t_2) = [50, 80]$; $I_2(t_3) = [70, 90]$; $I_2(t_4) = [100, 120]$; $I_2(t_5) = [140, 170]$; $I_2(t_6) = [170, 210]$;

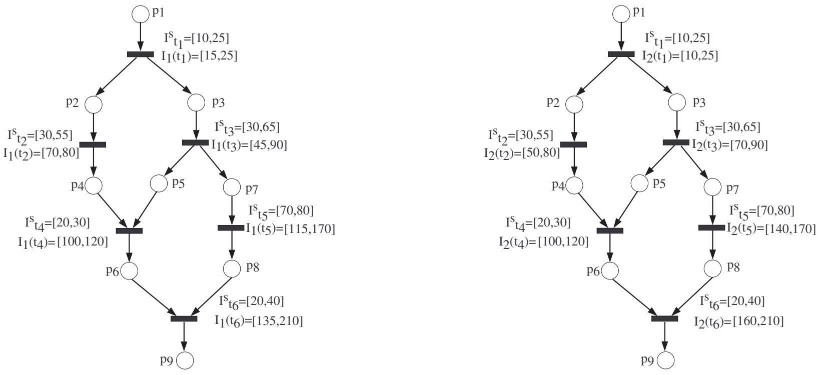


Figure 5.9:

In Fig. 5.9-left we display $\widetilde{\mathcal{N}}^{\theta}$ where attached to each transition t_ι is the time interval when transition t_ι can be executed considering case 1, while in Fig. 5.9-right there is displayed the time intervals corresponding to case 2.

5.5.2 Analysis of Time Petri Nets based on time processes - the case of free choice TPN

In this section we consider the case of free-choice PNs as a preamble of the next section where we treat the general case of TPNs. The reason for doing so is that a TPN model that has a free-choice PN support preserves the untimed language under the equivalence relation $\equiv_{\Sigma_\mu}$ (see Proposition 20) and this makes the presentation of the method for this subclass of models easier. Then, having already introduced the main concept we later generalize the method for a general PN model. Notice that in what follows the 1-safe assumption remains.

Definition 59. *Consider a free-choice TPN model $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ and then let $\mathcal{U}_\mathcal{N}(M_0)$ be the unfolding of its untimed PN support $\langle \mathcal{N}, M_0 \rangle$. Denote $\mathcal{C}$ the set of all configurations in $\mathcal{U}_\mathcal{N}(M_0)$. Consider a configuration $C = (B_C, E_C, \preceq)$ and denote $C^\theta = (B_C, E_C, \preceq, I^s)$ the TPN that has as its untimed support $C = (B_C, E_C, \preceq)$ while $I^s : E_C \to \mathcal{I}(\mathbb{Q}_+)$ is the function that gives the static interval for an event $e \in E_C$, where $I^s(e) = I^s(t)$ with $\phi(e) = t$.*

Denote K_{C^θ} the characteristic system of C^θ:

$$K_{C^\theta} = \begin{cases} \mathtt{max}_{e_v \in {}^{\bullet\bullet}e_1}(\theta_{e_v}) + L^s_{e_1} & \leq & \theta_{e_1} & \leq & \mathtt{max}_{e_v \in {}^{\bullet\bullet}e_1}(\theta_{e_v}) + U^s_{e_1} \\ \mathtt{max}_{e_v \in {}^{\bullet\bullet}e_2}(\theta_{e_v}) + L^s_{e_2} & \leq & \theta_{e_2} & \leq & \mathtt{max}_{e_v \in {}^{\bullet\bullet}e_2}(\theta_{e_v}) + U^s_{e_2} \\ \quad\vdots & & & & \\ \mathtt{max}_{e_v \in {}^{\bullet\bullet}e_{|E_C|}}(\theta_{e_v}) + L^s_{e_{|E_C|}} & \leq & \theta_{e_N} & \leq & \mathtt{max}_{e_v \in {}^{\bullet\bullet}e_N}(\theta_{e_v}) + U^s_{e_{|E_C|}} \end{cases} \tag{5.26}$$

and $Sol(K_{C^\theta})$ denotes the set of all solutions for C^θ.

Consider $Sol_\nu(K_{C^\theta})$ a subset of $Sol(K_{C^\theta})$ and then denote in the following by $\mathbf{I}_\nu$ the smallest $\mid E_C \mid$-hyperbox that includes $Sol_\nu(K_{C^\theta})$. If $Sol_\nu(K_{C^\theta})$ has the time independence property we say that $C(\mathbf{I}_\nu)$ is a time-interval configuration of $\langle \mathcal{N}^\theta, M_0^\theta \rangle$.

Given the untimed configuration $C \in \mathcal{C}$ consider its event-set E_C partitioned in disjunct subsets (layers) as follows:

- $Layer^{\downarrow}[0] = \{e_\iota \in E_C \mid {}^{\bullet\bullet}e = \emptyset\}$

- $Layer^{\downarrow}[y] = \Big\{e_\iota \in E_C \mid ({}^{\bullet\bullet}e_\iota \subseteq \bigcup_{z=0}^{y-1} Layer^{\downarrow}[z]) \wedge$
  $({}^{\bullet\bullet}e_\iota \cap Layer^{\downarrow}[y-1] \neq \emptyset)\}$

The following algorithm calculates an $\mid E_{C^\theta} \mid$-hyperbox $\mathbf{I}$ for a configuration C of the unfolding of the untimed support PN of a TPN model $\langle \mathcal{N}^\theta, M_0^\theta \rangle$.

Algorithm 14 Time Interval

```
Require: $C \in \mathcal{TC}$
Ensure: $C(\mathbf{I})$
for all $e_\iota \in Layer^{\downarrow}[0]$ do
    $I(e_\iota) = I^s_{e_\iota}$
end for
for $y = 1$ to $y_{max}$ do
    for all $e_\iota \in Layer^{\downarrow}[y]$ do
        $L(e_\iota) = \mathtt{max}_{e_\lambda \in {}^{\bullet\bullet}e_\iota}(L(e_\lambda)) + L^s_{e_\iota}$
        $U(e_\iota) = \mathtt{max}_{e_\lambda \in {}^{\bullet\bullet}e_\iota}(U(e_\lambda)) + U^s_{e_\iota}$
        $I(e_\iota) = [L(e_\iota), U(e_\iota)]$
    end for
end for
```

Proposition 23. *For a TPN $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ that has an untimed support $\mathcal{N}$ that is free choice we have that for any untimed configuration $\forall C \in \mathcal{C}$, $C(\mathbf{I})$ is a time interval configuration of $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ where $\mathbf{I}$ is obtained by running Algorithm 14.*

Proof. The proof is straight forward since $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ is untimed 1-safe and free choice thus if a transition becomes enabled it cannot be disabled by the execution of an untimed concurrent event. □

Consider a time interval configuration $C_\ell(\mathbf{I}_\ell)$ of an 1-safe free-choice TPN $\langle \mathcal{N}^\theta, M_0^\theta \rangle$. Then denote $EXT^1(C_\ell(\mathbf{I}_\ell))$ the set of extensions of $C_\ell(\mathbf{I}_\ell)$ by appending an event e that is enabled in $CUT(C_\ell)$:

$$EXT^1(C_\ell(\mathbf{I}_\ell)) = \{C_v(\mathbf{I}_v) \mid C_v(\mathbf{I}_v) = C_\ell(\mathbf{I}_\ell) \odot (e, I(e))\}$$

where $I_v(e) = [L_v(e), U_v(e)]$ with $L_v(e) = \mathtt{max}_{e' \in {}^{\bullet\bullet}e}(L_\ell(e')) + L^s_e$ and $U_v(e) = \mathtt{max}_{e' \in {}^{\bullet\bullet}e}(U_\ell(e')) + U^s_e$.

Notice that $C_v = C_\ell \odot e$ is an untimed configuration that extends C_ℓ by e while the vector of execution time-intervals $\mathbf{I}_v$ of the events in C_v is obtained from $\mathbf{I}_\ell$ by adding the element $I_v(e)$ that is the execution time of the event e that was appended.

$EXT^*(C_\ell(\mathbf{I}_\ell))$ is the set of all the finite extensions of a time configuration $C_\ell(\mathbf{I}_\ell)$ where recursively:

1. $EXT^0(C_\ell(\mathbf{I}_\ell)) = \{C_\ell(\mathbf{I}_\ell)\}$
2. $EXT^q(C_\ell(\mathbf{I}_\ell)) = \big\{C_v(\mathbf{I}_v) \mid \exists C_\iota(\mathbf{I}_\iota) \in EXT^{q-1}(C_\ell(\mathbf{I}_\ell)) \wedge C_v(\mathbf{I}_v) \in EXT^1(C_\iota(\mathbf{I}_\iota))\big\}$

Given an 1-safe free-choice TPN $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ and a time interval configuration $C_\ell(\mathbf{I})$ we have that $\forall C_v(\mathbf{I}_v) \in EXT^*(C_\ell(\mathbf{I}_\ell))$, $C_v(\mathbf{I}_v)$ is a time interval configuration.

Denote by $C^{\perp}(\mathbf{I}^{\perp})$ the *"initial"* time-interval configuration of $\langle \mathcal{N}^{\theta}, M_0^{\theta} \rangle$ where $B_{C^{\perp}} = \mathtt{min}(\mathcal{U}_{\mathcal{N}})$, $E_{C^{\perp}} = \emptyset$ and $\mathbf{I}^{\perp} = 0$ (assuming that the tokens in M_0^{θ} are produced at the time 0). Then $EXT^{*}(C^{\perp}(\mathbf{I}^{\perp}))$ denotes the set of all finite time-interval configurations of $\langle \mathcal{N}^{\theta}, M_0^{\theta} \rangle$.

5.5.3 Centralized diagnosis of free choice TPN based on time processes

In the following we design an on-line monitoring algorithm that derives the diagnosis of the plant at the time ξ based on the observation that is received up to the time ξ.

Consider that the plant monitoring starts at the time 0. Starting from the initial configuration $C^{\perp}(\mathbf{I}^{\perp})$ we derive time-interval configurations $C_{\upsilon}(\mathbf{I}_{\upsilon})$ up to a time (called discarding time and denoted $\widehat{\theta}_{\upsilon}$) when in absence of any received observation the time interval configuration $C_{\upsilon}(\mathbf{I}_{\upsilon})$ is discarded because it violates the observation, i.e.:

- starting from the initial time configuration $C^{\perp}(\mathbf{I}^{\perp})$ derive the set of time interval configurations that are obtained from $C^{\perp}(\mathbf{I}^{\perp})$ by appending an event e that has the minimal upper limit $U(e)$ among all the events that are enabled in $C^{\perp}(\mathbf{I}^{\perp})$.

- recursively derive further extensions by appending events having the minimal upper limit until the event e that is appended is an observable event e^{o} ($\phi(e^{o}) \in \mathcal{T}_{o}$).

- let $\widehat{\theta}_{\upsilon} = U_{\upsilon}(e^{o})$ be the discarding time for the time interval configuration $C_{\upsilon}(\mathbf{I}_{\upsilon})$. Then extend $C_{\upsilon}(\mathbf{I}_{\upsilon})$ by appending events until all the enabled events in an extension $C_{\iota}(\mathbf{I}_{\iota})$ of $C_{\upsilon}(\mathbf{I}_{\upsilon})$ have their lower bound greater than $\widehat{\theta}_{\upsilon}$ ($\forall e' \in ENABLED(C_{\iota})$, $L_{\iota}(e') > \widehat{\theta}_{\upsilon}$). Obviously all the events e in $C_{\iota}(\mathbf{I}_{\iota})$ have they lower bound $L_{\iota}(e)$ of their execution time interval smaller than the discarding time $\widehat{\theta}_{\upsilon}$, i.e.

$$\forall e \in E_{C_{\iota}(\mathbf{I}_{\iota})} \Rightarrow L_{\iota}(e) \leq \widehat{\theta}_{\upsilon}$$

In the following we use the *"hat"* symbol to denote the time-interval configurations that are derived w.r.t. a discarding time $\widehat{\theta}$.

Denote by $\widehat{\mathcal{TC}}(\mathtt{obs}_1 \mid \mathcal{O}_0^{\theta})$ the set of time interval configurations derived up to the first discarding time by running the Algorithm 15 presented below.

Algorithm 15

```
Require: $C^{\perp}(\mathbf{I}^{\perp})$
Ensure: $\widehat{\mathcal{TC}}(\mathtt{obs}_1 \mid \mathcal{O}_0^{\theta})$
$SET_1 = \left\{C^{\perp}(\mathbf{I}^{\perp})\right\}$
$SET_2 = \emptyset$; $\widehat{\mathcal{TC}}(\mathtt{obs}_1 \mid \mathcal{O}_0^{\theta}) = \emptyset$
while $SET_1 \neq \emptyset$ do
    pick $C_\ell(\mathbf{I}_\ell) \in SET_1$
    $Append_1(C_\ell(\mathbf{I}_\ell)) = \{e \mid \forall e' \in ENABLED(C_\ell), U_\ell(e) \leq U_\ell(e')\}$
    for all $e \in Append_1(C_\ell(\mathbf{I}_\ell))$ do
        $C_v(\mathbf{I}_v) = C_\ell(\mathbf{I}_\ell) \odot (e, I_v(e))$
        if $\phi(e) \in \mathcal{T}_o$ then
            $\widehat{\theta}_v = U_v(e)$
            $SET_2 = SET_2 \cup \{C_v(\mathbf{I}_v)\}$
        else
            $SET_1 = SET_1 \cup \left\{\widehat{C}_v(\mathbf{I}_v)\right\}$
        end if
    end for
    remove $C_\ell(\mathbf{I}_\ell)$ from $SET_1$
end while
while $SET_2 \neq \emptyset$ do
    choose $C_v(\mathbf{I}_v) \in SET_2$
    $Append_2(C_v(\mathbf{I}_v)) = \left\{e \in ENABLED(C_v) \mid L(e) \leq \widehat{\theta}_v\right\}$
    if $Append_2(C_v(\mathbf{I}_v)) \neq \emptyset$ then
        for all $e \in Append_2(\widehat{C}_v(\mathbf{I}_v))$ do
            $\widehat{C}_\iota(\mathbf{I}_\iota) = C_v(\mathbf{I}_v) \odot (e, I_\iota(e))$
            $SET_2 = SET_2 \cup \left\{\widehat{C}_\iota(\mathbf{I}_\iota)\right\}$
        end for
    end if
    remove $\widehat{C}_v(\mathbf{I}_v)$ from $SET_2$
end while
$\widehat{\mathcal{TC}}(\mathtt{obs}_1 \mid \mathcal{O}_0^{\theta}) = SET_2$
```

Formally the set of time interval configurations derived before the first observed event is:

$$\widehat{\mathcal{TC}}(\mathtt{obs}_1 \mid \mathcal{O}_0^{\theta}) = \left\{\widehat{C}_\ell(\mathbf{I}_\ell) \mid \ell \in \mathcal{V}_{\mathtt{obs}_1}\right\}$$

where $\widehat{C}_\ell(\mathbf{I}_\ell) \in \widehat{\mathcal{TC}}(\mathtt{obs}_1 \mid \mathcal{O}_0^{\theta})$ is such that:

1. $\widehat{\theta}_\ell = U_\ell(e)$ s.t. $\phi(e) \in \mathcal{T}_o$ and $\forall e' \in E_{C_\ell}$, $\phi(e') \in \mathcal{T}_o \Rightarrow \widehat{\theta}_\ell \leq U_\ell(e')$

2. $\forall e \in ENABLED(\widehat{C}_\ell) \Rightarrow L_\ell(e)) > \widehat{\theta}_\ell$

Then we have the following cases:

case 1 if there is no observation until the time $\widehat{\theta}_\ell$ or the observed label obs_1 is such that for any observable event $e^o \in E_{\widehat{C}_\ell}$ we have that $\ell(\phi(e^o)) \neq obs_1$ then discard $\widehat{C}_\ell(\mathbf{I}_\ell)$

case 2 the label obs_1 is observed and for all events $e^o \in E_{\widehat{C}_\ell}$ s.t. $\ell(\phi(e^o)) = obs_1$ we have that $\theta_{obs_1} < L_\ell(e^o)$ then discard $\widehat{C}_\ell(\mathbf{I}_\ell)$

case 3 the label obs_1 is observed at the time θ_{obs_1}; then for each e'^o s.t. $\ell(\phi(e'^o)) = obs_1$ and $\theta_{obs_1} \in [L_\ell(e'^o), U_\ell(e'^o)]$ update $\widehat{C}_\ell(I_\ell)$ according with the following rule:
impose the constraints:

$$\mathcal{K}_{e'^o} = \kappa_{e'^o} \wedge \mathcal{K}_{e''^o \neq e'^o}$$

where

$$\kappa_{e'^o} := \{\theta_{e'^o} = \theta_{obs_1}\} \tag{5.27}$$

is the constraint that the observation is explained by e'^o and for all the other observable event $e''^o \in E_{\widehat{C}_\ell}$, $e''^o \neq e'^o$ we have the constraints that $e''^o \in E_{\widehat{C}_\ell} \setminus \{e'^o\}$ happens after e'^o:

$$\mathcal{K}_{e''^o \neq e'^o} := \bigwedge_{e''^o \in E_{\widehat{C}_\ell} \setminus \{e'^o\}} \{\mathtt{max}(L_\ell(e''^o), \theta_{obs_1}) \leq \theta_{e''^o} \leq U_\ell(e''^o)\} \tag{5.28}$$

If e^o is the event that explains the observed label obs_1 and e^o corresponds with $\widehat{\theta}_\ell$ then the time-interval configurations that result after imposing set of constraints $\mathcal{K}(\mathcal{O}_1^\theta)$ given by Eq.5.27 and Eq.5.28 are extended up to a new discarding time that corresponds with the second observation.

By running Algorithm 11 for each constraint κ_e of $\mathcal{K}_{C_\ell}$ we obtain the set of $\mid E_{C_\ell} \mid$-hyperboxes

$$\Gamma(\mathcal{K}_\ell) = \left\{ \mathbf{I}_{\nu_\ell} \mid \nu_\ell \in \mathcal{V}_\ell^{obs_1} \right\}$$

For $\widehat{C}_\ell(\mathbf{I}_\ell)$ constrained by received observation $\mathcal{O}_1^\theta$ we obtain the set of time-interval configurations:

$$\widehat{\mathcal{TC}}_\ell(\mathcal{O}_1^\theta) = \left\{ \widehat{C}_\ell(I_{\nu_\ell}) \mid \nu_\ell \in \mathcal{V}_{\ell_{obs_1}} \right\} \tag{5.29}$$

In general given a set of time interval configurations $\widehat{\mathcal{TC}}(\mathtt{obs}_k \mid \mathcal{O}_{k-1}^\theta)$ denote by $\Xi(\widehat{\mathcal{TC}})$ the set of discarding times $\widehat{\theta}_\ell$ of the time interval configurations $\widehat{C}_\ell(\mathbf{I}_\ell) \in \widehat{\mathcal{TC}}(\mathtt{obs}_k \mid \mathcal{O}_{k-1}^\theta)$:

$$\Xi(\widehat{\mathcal{TC}}) = \left\{\widehat{\theta}_\ell \mid \widehat{C}_\ell(\mathbf{I}_\ell) \in \widehat{\mathcal{TC}}(\mathtt{obs}_k \mid \mathcal{O}^\theta_{k-1})\right\}$$

Assume that $\Xi(\widehat{\mathcal{TC}}(\mathtt{obs}_k \mid \mathcal{O}^\theta_k))$ is sorted by the time progress, that is $HEAD(\Xi(\widehat{\mathcal{TC}}))$ is the smallest discarding time and $\widehat{\theta}_\ell$ is closer to $HEAD(\Xi)$ than $\widehat{\theta}_\upsilon$ iff $\widehat{\theta}_\ell \leq \widehat{\theta}_\upsilon$. Let $HEAD^{-1}(\Xi(\widehat{\mathcal{TC}}))$ be the time interval configuration that has the discarding time $\widehat{\theta} = HEAD(\Xi(\widehat{\mathcal{TC}}))$.

The monitoring system that we design works as follows:

Algorithm 16 Monitor

```
Require: $\langle \mathcal{N}^\theta, M_0^\theta \rangle$
Ensure: $\widehat{\mathcal{TC}}(\mathcal{O}^\theta_\xi)$
the plant analysis starts at a given time $\xi = 0$
compute $\widehat{\mathcal{TC}}(\mathtt{obs} \mid \mathcal{O}^\theta_\xi))$ and $\Xi(\widehat{\mathcal{TC}}(\mathcal{O}^\theta_\xi))$
repeat
  $\xi = \xi + \delta\xi$ (time progress)
  if $\xi > HEAD(\Xi(\widehat{\mathcal{TC}}))$ and $\mathcal{O}^\theta_\xi = \mathcal{O}^\theta_{\xi-\delta\xi}$ then
    remove $HEAD^{-1}(\Xi)$ from $\widehat{\mathcal{TC}}$
  end if
  if $\mathcal{O}^\theta_{\xi+\delta\xi} = \mathcal{O}^\theta_\xi \cup \langle obs, \theta_{obs} \rangle$ then
    for all $\widehat{C}_\ell(\mathbf{I}_\ell) \in \widehat{\mathcal{TC}}(\mathtt{obs} \mid \mathcal{O}^\theta_\xi)$ do
      if case 1 or case 2 then
        remove $\widehat{C}_\ell(\mathbf{I}_\ell)$ from $\widehat{\mathcal{TC}}(\mathtt{obs} \mid \mathcal{O}^\theta_\xi)$
      end if
      if case 3 then
        for all $e^o_\lambda \in E_{\widehat{C}_\ell}$ s.t. $l_o(e^o_\lambda) = obs$ and $\theta_{obs} \in [L_\ell(e^o_\lambda), U_\ell(e^o_\lambda)]$ do
          $\mathcal{K}_\ell$ for $e^o_\lambda$ (Eq.5.27, Eq.5.28)
          $\widehat{\mathcal{TC}}_\ell(\mathcal{O}^\theta_\xi) = \left\{\widehat{C}_\ell(\mathbf{I}_{\nu_\ell}) \mid \nu_\ell \in \mathcal{V}_\ell\right\}$
          for all $C_{\nu_\ell}(\mathbf{I}_{\nu_\ell}) \in \widehat{\mathcal{TC}}_\ell(\mathcal{O}^\theta_\xi)$ do
            extend $C_{\nu_\ell}(\mathbf{I}_{\nu_\ell})$ up the next discarding time $(\widehat{\mathcal{TC}}_{\nu_\ell}(\mathcal{O}^\theta_\xi))$
            $\widehat{\mathcal{TC}}(\mathtt{obs} \mid \mathcal{O}^\theta_\xi) = \widehat{\mathcal{TC}}(\mathtt{obs} \mid \mathcal{O}^\theta_\xi) \cup \widehat{\mathcal{TC}}_{\nu_\ell}(\mathtt{obs} \mid \mathcal{O}^\theta_\xi)$
          end for
        end for
        update the set of discarding timed $\Xi(\widehat{\mathcal{TC}}(\mathtt{obs} \mid \mathcal{O}^\theta_\xi))$
      end if
    end for
  end if
until stop monitoring
```

Let $\widehat{\mathcal{TC}}(\mathcal{O}^\theta_\xi)$ be the set of time interval configurations that are derived by running on-line the Algorithm 16 at the time ξ after the last observed event has occurred.

Denote in the following by $\mathcal{TC}(\mathcal{O}_\xi^\theta)$ the set of time interval configurations that are derived by Algorithm 16 at the time ξ without calculating the further extensions that predict the next observation. Then $C_\ell(\mathbf{I}_\ell) \in \mathcal{TC}(\mathcal{O}_\xi^\theta)$ iff:

1. $\exists \widehat{C}_v(\mathbf{I}_v) \in \widehat{\mathcal{TC}}(\mathcal{O}_\xi^\theta)$ and $C_\ell(\mathbf{I}_\ell) \sqsubseteq \widehat{C}_v(\mathbf{I}_v)$
2. $\forall e \in ENABLED(C_\ell)$
   (a) if $\phi(e) \in \mathcal{T}_{uo}$ then $U_\ell(e) > \theta_{obs_n}$
   (b) if $\phi(e^o) \in \mathcal{T}_o$ then $L_\ell(e) > \theta_{obs_n}$
3. $\forall e \in E_C$, $L_\ell(e) \leq \theta_{obs_n}$

Notice that for $C_\ell(\mathbf{I}_\ell) \in \mathcal{TC}(\mathcal{O}_\xi^\theta)$ we have that the last unobservable events in $C_\ell(\mathbf{I}_\ell)$ can be executed before or after ξ.

Definition 60. *Given a time-interval configuration $C_\ell(\mathbf{I}_\ell)$, denote by $\langle E_{C_\ell(\mathbf{I}_\ell)}\rangle$ the set of time linearizations of $(E_{C_\ell(\mathbf{I}_\ell)}, \preceq)$ where:*

$$\langle E_{C_\ell(\mathbf{I}_\ell)}\rangle = \Big\{\sigma = e_1 \dots e_\iota \dots e_\lambda \dots e_{|E_{C_\ell}|} \mid \iota < \lambda \Rightarrow \\ \Rightarrow (e_\iota \preceq e_\lambda) \vee ((e_\iota \| e_\lambda) \wedge (L_\ell(e_\iota) \leq U_\ell(e_\lambda)))\}$$

The set of untimed traces represented by the set of time interval configurations $\mathcal{TC}(\mathcal{O}_\xi^\theta)$ (calculated at the time ξ by running Algorithm 16) is:

$$\mathcal{E}(\mathcal{O}_\xi^\theta) = \Big\{\sigma \in \langle E_{C_\ell(\mathbf{I}_\ell)}\rangle \mid C_\ell(\mathbf{I}_\ell) \in \mathcal{TC}(\mathcal{O}_\xi^\theta)\Big\}$$

Denote $\mathcal{L}^{fc}(\mathcal{O}_\xi^\theta)$ the set of untimed support traces that are represented by $\mathcal{TC}(\mathcal{O}_\xi^\theta)$:

$$\mathcal{L}^{fc}(\mathcal{O}_\xi^\theta) = \{\tau = \phi(\sigma) \mid \sigma \in \mathcal{E}(\mathcal{O}_\xi^\theta)\}$$

Proposition 24. *Given $\mathcal{L}_{\mathcal{N}^\theta}(\mathcal{O}_\xi^\theta)$ (the untimed support language of $\mathcal{L}_{\mathcal{N}^\theta}^\theta(\mathcal{O}_\xi^\theta)$) and $\mathcal{L}^{fc}(\mathcal{O}_\xi^\theta)$ we have that:*

$$\mathcal{L}_{\mathcal{N}^\theta}(\mathcal{O}_\xi^\theta) \equiv_{\Sigma_\mu} \mathcal{L}^{fc}(\mathcal{O}_\xi^\theta)$$

that is:

i) $\forall \tau \in \mathcal{L}_{\mathcal{N}^\theta}(\mathcal{O}_\xi^\theta) \Rightarrow \exists \tau' \in \mathcal{L}^{fc}(\mathcal{O}_\xi^\theta)$ *s.t.* $\Sigma_\mu(\tau) = \Sigma_\mu(\tau')$

ii) and $\forall \tau' \in \mathcal{L}^{fc}(\mathcal{O}_\xi^\theta) \Rightarrow \exists \tau \in \mathcal{L}_{\mathcal{N}^\theta}(\mathcal{O}_\xi^\theta)$ *s.t.* $\Sigma_\mu(\tau) = \Sigma_\mu(\tau')$

Proof. ($\Rightarrow$) Consider $\tau \in \mathcal{L}_{\mathcal{N}^\theta}(\mathcal{O}_\xi^\theta)$. Given the untimed unfolding $\mathcal{U}_\mathcal{N}(M_0)$, let C_ℓ be an untimed configuration s.t. $\exists \sigma \in \langle E_{C_\ell}\rangle$ s.t. $\tau = \phi(\sigma)$. Denote C_ℓ^θ the TPN obtained from C_ℓ by attaching to each event e the static interval of its image transition t ($I_e^s = I_t^s$ and $\phi(e) = \phi(t)$). Denote $K_{C_\ell^\theta}$ the characteristic system of C_ℓ^θ and then denote $K_{C_\ell^\theta} \bigwedge_{k=1}^{k=n} \kappa_{e_k^o}$ the system of inequalities obtained by adding to the characteristic system $K_{C_\ell^\theta}$ equalities regarding the observed events, i.e. for $k = 1, \ldots, n$ $\kappa_{e_k^o} := \left\{\theta_{e_k^o} = \theta_{t_k^o} : \phi(e_k^o) = t_k^o\right\}$.

We have that $\Theta(\tau) \in Sol(K_{C_\ell^\theta} \bigwedge_{k=1}^{k=n} \kappa_{e_k^o})$ where for $\tau = t_1 \ldots t_\gamma$, $\Theta(\tau) = (\theta_{\phi(t_1)}, \ldots, \theta_{\phi(t_\gamma)})$.

Hence there exists a time interval configuration $C_\ell(\mathbf{I}_{\nu_\ell})$ such that $\tau = \phi(\sigma)$ with $\sigma \in \langle E_{C_\ell} \rangle$.

Let $\Gamma(\bigwedge_{k=1}^{k=n} \kappa_{e_k^o}) = \{\mathbf{I}_{\nu_\ell} \mid \nu_\ell \in \mathcal{V}_\ell\}$ be the set of $\mid E_{C_\ell} \mid$ rectangles that is obtained running Algorithm 11 for C^θ and the set of constrains $\left\{\kappa_{e_1^o}, \ldots, \kappa_{e_n^o}\right\}$. Since $Sol(K_{C_\ell^\theta} \bigwedge_{k=1}^{k=n} \kappa_{e_k^o}) \subseteq \bigcup_{\nu_\ell \in \mathcal{V}_\ell} \mathbf{I}_{\nu_\ell}$ it means that $\exists \nu_\ell \in \mathcal{V}_\ell$ s.t. $\Theta(\tau) \in \mathbf{I}_{\nu_\ell}$.

Hence $\exists C_\ell(\mathbf{I}_{\nu_\ell}) \in \mathcal{TC}(\mathcal{O}_\xi^\theta)$ s.t. $\tau = \phi(\sigma)$ with $\sigma \in \langle E_{C_\ell} \rangle$.

($\Leftarrow$) The proof is straightforward. □

Denote by $\mathcal{DR}^{fc}(\mathcal{O}_\xi^\theta)$ the diagnosis result for a free choice TPN derived using $\mathcal{L}^{fc}(\mathcal{O}_\xi^\theta)$. Then we have the following result:

Theorem 7. *Given TPN model $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ that is 1-safe and free-choice and the observation generated by the plant $\mathcal{O}_\xi^\theta$, we have that:*

$$\mathcal{DR}(\mathcal{O}_\xi^\theta) = \mathcal{DR}^{fc}(\mathcal{O}_\xi^\theta) \tag{5.30}$$

Proof. The proof is straightforward since by Proposition 24 we have that $\mathcal{L}_{\mathcal{N}^\theta}(\mathcal{O}_\xi^\theta) \equiv_{\Sigma_\mu} \mathcal{L}^{fc}(\mathcal{O}_\xi^\theta)$. □

Example 31. *Consider the TPN in Fig. 5.10 where next to each transition its static interval is displayed. The observable transitions are $\mathcal{T}_o = \{t_3, t_4, t_{13}, t_{14}\}$ while all the other transitions are unobservable (silent). The observable transitions have the same observation label, i.e. $l(t_3) = l(t_4) = l(t_{13}) = l(t_{14})$. The fault transitions are t_9 and t_{12}.*

The centralized monitoring Algorithm 16, works as follows. Assume that the plant analysis starts at the global time 0*.*

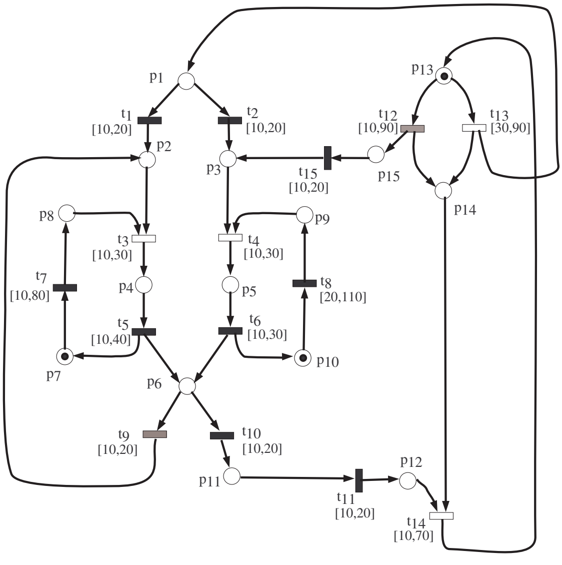


Figure 5.10:

The set of configurations that estimate the first observed event is:

$$\widehat{\mathcal{C}}(\texttt{obs}_\texttt{1} \mid \mathcal{O}_0^\theta) = \left\{\widehat{C}_1(\texttt{obs}_\texttt{1} \mid \mathbf{I}_1), \widehat{C}_2(\texttt{obs}_\texttt{1} \mid \mathbf{I}_2)\right\}$$

$\widehat{C}_1(\texttt{obs}_\texttt{1} \mid \mathbf{I}_1)$ *(Fig. 5.11) is obtained in the following way:*

1. *first e_7 is appended since it is the enabled event with the lower upper limit of its executions interval.*

2. *the observable event e_{13} is appended and the discarding time for the time configuration $\widehat{C}_1(\texttt{obs}_\texttt{1} \mid \mathbf{I}_1)$ is is $\widehat{\theta}_1 = U(e_{13}) = 90$*

3. *the unobservable event e_8 is appended*

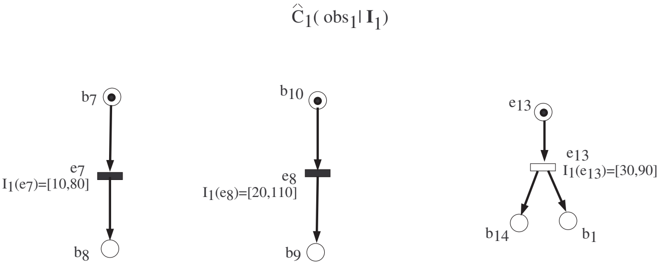


Figure 5.11:

$\widehat{C}_2(\mathtt{obs_1} \mid \mathbf{I}_2)$ is obtained similarly having its discarding time $\widehat{\theta}_2 = U(e_4) = 140$ (Fig. 5.12).

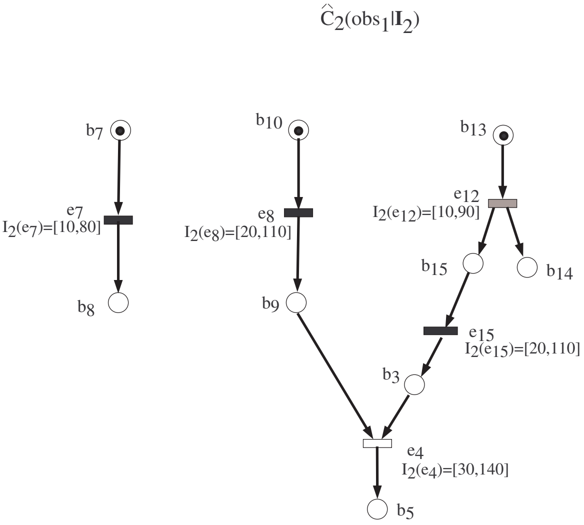


Figure 5.12:

Consider that at the time $\xi = 80$ the first observation is received. The constraint $\theta_{e_{13}} = 80$ ($I(e_{13}) = 80$) is imposed to $\widehat{C}_1(\mathtt{obs_1} \mid \mathbf{I}_1)$ and we obtain $\widehat{C}_1(\mathbf{I}_1)$ (Fig. 5.13).

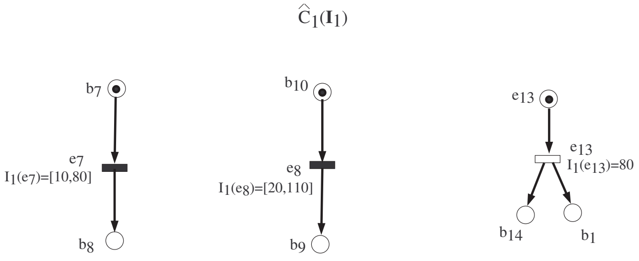


Figure 5.13:

The constraint $\theta_{e_{13}} = 80$ *(*$I(e_4) = 80$*) is imposed to* $\widehat{C}_2(\mathtt{obs_1} \mid \mathbf{I}_2)$ *as follows:*

- $\theta_{e_{13}} = 80$ *implies that both* e_{15} *and* e_7 *were executed before* 70

- *and at least one of them was executed after the time* 50.

- *we obtain* $\widehat{C}_{1_2}(\mathbf{I}_{1_2})$ *(Fig. 5.14) and* $\widehat{C}_{2_2}(\mathbf{I}_{2_2})$ *(Fig. 5.15).*

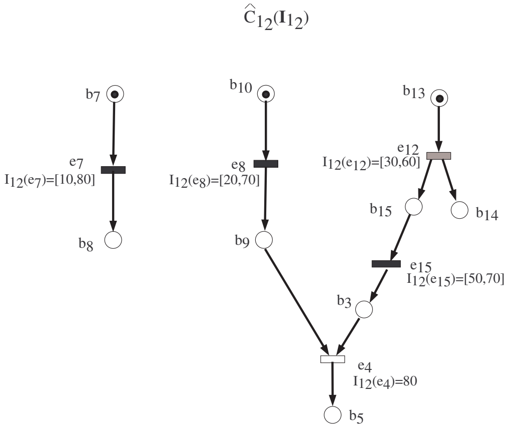


Figure 5.14:

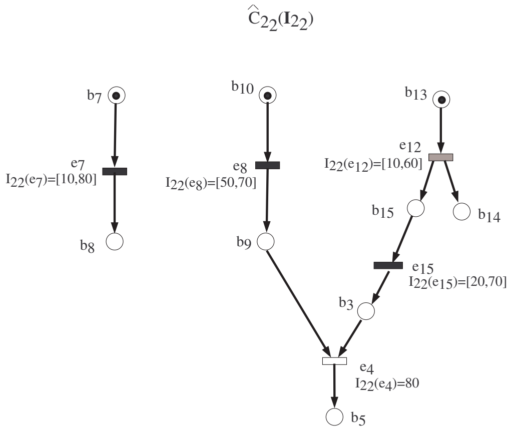


Figure 5.15:

After the first observation at the time 80 *we obtain:*

$$\mathcal{C}(\mathcal{O}_1^\theta) = \left\{ \widehat{C}_1(\mathbf{I}_1), \widehat{C}_{1_2}(\mathbf{I}_{1_2}), \widehat{C}_{2_2}(\mathbf{I}_{2_2}) \right\}$$

Then each of the three time-interval configurations is extended up to the next discarding time for estimating the next observation.

For $\widehat{C}_1(\mathbf{I}_1)$ *we obtain* $\widehat{\mathcal{C}}_1(\mathtt{obs}_2 \mid \mathcal{O}_1^\theta) = \left\{ \widehat{C}_{1_1}(\mathtt{obs}_2 \mid \mathbf{I}_{1_1}), \widehat{C}_{2_1}(\mathtt{obs}_2 \mid \mathbf{I}_{2_1}) \right\}$.

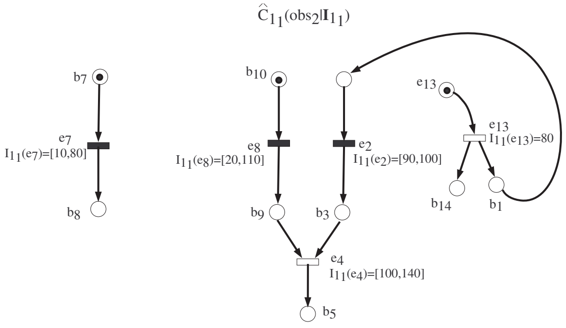


Figure 5.16:

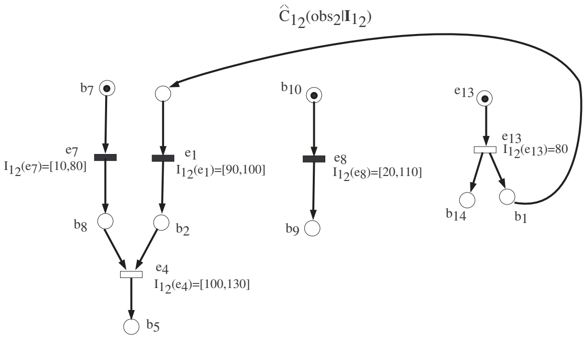


Figure 5.17:

For $\widehat{C}_{1_2}(\mathbf{I}_{1_2})$ *we obtain* $\widehat{\mathcal{C}}_{1_2}(\mathtt{obs}_2 \mid \mathcal{O}_1^{\theta}) = \Big\{\widehat{C}_{1_{1_2}}(\mathtt{obs}_2 \mid \mathbf{I}_{1_{1_2}}), \widehat{C}_{2_{1_2}}(\mathtt{obs}_2 \mid \mathbf{I}_{2_{1_2}})\Big\}.$

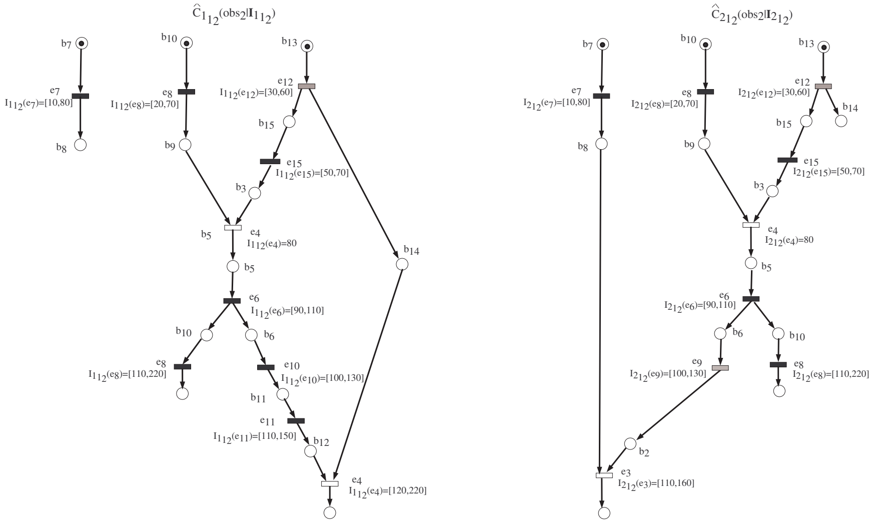


Figure 5.18:

For $\widehat{C}_{2_2}(\mathbf{I}_{2_2})$ *we obtain* $\widehat{\mathcal{C}}_{2_2}(\mathtt{obs}_2 \mid \mathcal{O}_1^{\theta}) = \Big\{\widehat{C}_{1_{2_2}}(\mathtt{obs}_2 \mid \mathbf{I}_{1_{2_2}}), \widehat{C}_{2_{2_2}}(\mathtt{obs}_2 \mid \mathbf{I}_{2_{2_2}})\Big\}.$

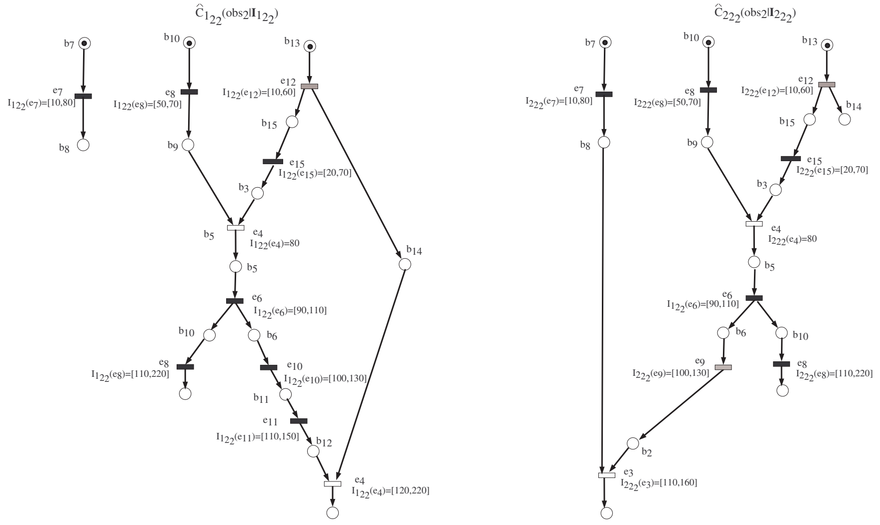


Figure 5.19:

Consider that the second observation is received at the time 135 *(i.e. the second observable event occurs at the time* 135*). At the time* 130 *the time configuration* $\widehat{C}_{2_1}(\texttt{obs}_2 \mid \mathbf{I}_{2_1})$ *is discarded because its discarding time* $\widehat{\theta}_{2_1}$ *is* 130*. The remaining time-configurations are refined imposing the constraint that the observation occurs at time* 135 *as follows.*

For $\widehat{C}_{1_1}(\mathbf{I}_{1_1})$ *we obtain* $\widehat{C}_{1_1}(\mathcal{O}_2^{\theta})$ *(Fig. 5.20). Notice that in* $\widehat{C}_{1_1}(\mathcal{O}_2^{\theta})$*, the event* e_8 *must be executed in the interval* $I(e_8) = [105, 110]$*.*

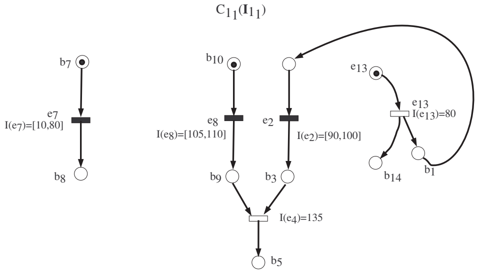


Figure 5.20:

For $\widehat{C}_{1_{1_2}}(\mathbf{I}_{1_{1_2}})$ *we obtain* $\widehat{C}_{1_{1_2}}(\mathcal{O}_2^{\theta})$ *(Fig. 5.21-left) where* e_4 *observed at the time* 135 *implies that the execution time-intervals of* e_{11}*,* e_{10}*,* e_6 *and* e_8 *are* $I_{1_{1_2}}(e_{11}) =$

$[110, 125]$*,* $I_{1_{1_2}}(e_{10}) = [100, 115]$*,* $I_{1_{1_2}}(e_6) = [90, 105]$ *and* $I_{1_{1_2}}(e_8) = [110, 210]$ *respectively.*

For $\widehat{C}_{2_{1_2}}(\mathbf{I}_{2_{1_2}})$ *we obtain* $\widehat{C}_{2_{1_2}}(\mathcal{O}_2^{\theta})$ *(Fig. 5.21-right) where* e_3 *observed at the time* 135 *implies that the execution time-interval of* e_9 *is* $I_{2_{1_2}}(e_3) = [100, 125]$.

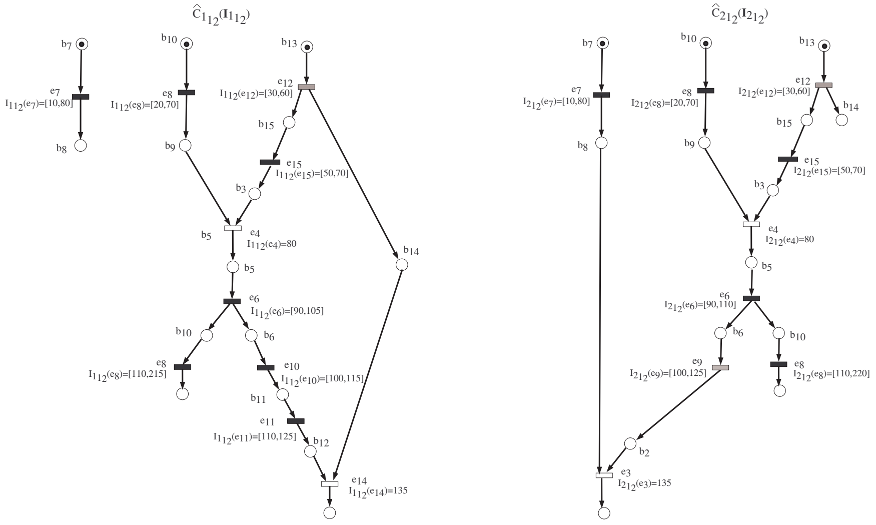


Figure 5.21:

For $\widehat{C}_{1_{2_2}}(\mathbf{I}_{1_{2_2}})$ *we obtain* $\widehat{C}_{1_{2_2}}(\mathcal{O}_2^{\theta})$ *(Fig. 5.22-left) where the fact that* e_4 *is observed at the time* 135 *implies that the execution time-intervals of* e_{11}*,* e_{10}*,* e_6 *and* e_8 *are* $I_{1_{2_2}}(e_{11}) = [110, 125]$*,* $I_{1_{2_2}}(e_{10}) = [100, 115]$*,* $I_{1_{2_2}}(e_6) = [90, 105]$ *and* $I_{1_{2_2}}(e_8) = [110, 210]$ *respectively.*

For $\widehat{C}_{2_{2_2}}(\mathbf{I}_{2_{2_2}})$ *we obtain* $\widehat{C}_{2_{2_2}}(\mathbf{I}_{2_{2_2}})$ *(Fig. 5.22-right) where the fact that* e_3 *is observed at the time* 135 *implies that the execution time-interval of* e_9 *is* $I_{2_{2_2}}(e_3) = [100, 125]$.

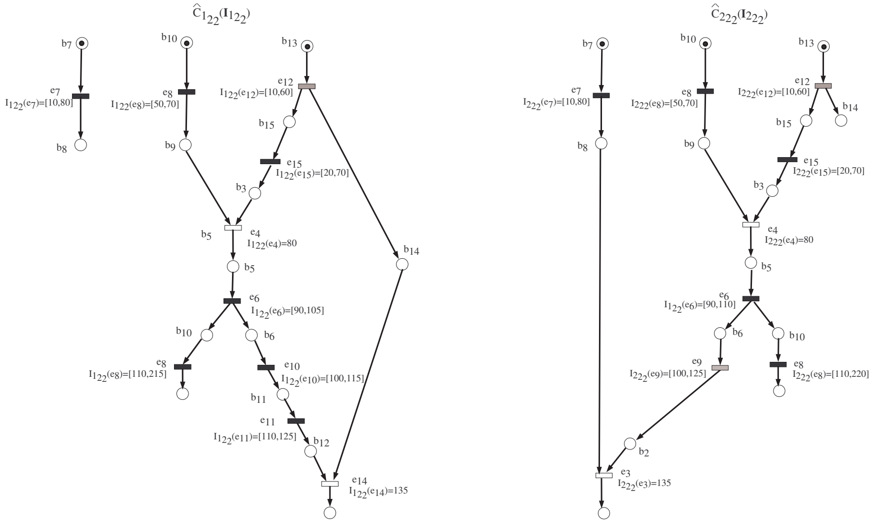


Figure 5.22:

At the time $\xi = 135$ we have obtained:

$$\mathcal{C}(\mathcal{O}_2^{\theta}) = \left\{\widehat{C}_{1_1}(\mathbf{I}_{1_1}), \widehat{C}_{1_{1_2}}(\mathbf{I}_{1_{1_2}}), \widehat{C}_{2_{1_2}}(\mathbf{I}_{2_{1_2}})\widehat{C}_{1_{2_2}}(\mathbf{I}_{1_{2_2}}), \widehat{C}_{2_{2_2}}(\mathbf{I}_{2_{2_2}})\right\}$$

The diagnosis results after observing $\mathcal{O}_2^{\theta} = \langle obs, 80\rangle\langle obs, 135\rangle$ is uncertain that a fault happened in the plant, i.e. $\mathcal{DR}^{fc}(\mathcal{O}_2^{\theta}) = \{\mathtt{UF}_{t_9}, \mathtt{UF}_{t_{12}}\}$ because $\widehat{C}_{1_1}$ contains no fault events whereas $\widehat{C}_{1_{1_2}}$ and $\widehat{C}_{1_{2_2}}$ contain the fault event e_{12} while $\widehat{C}_{2_{1_2}}$ and $\widehat{C}_{2_{2_2}}$ contain both the two fault events e_9 and e_{12} respectively.

In the following we derive the plant behavior up to the time ξ in order to compare the calculations based on time-processes with the calculations based on atomic state classes.

Given $\mathcal{TC}(\mathcal{O}_{\xi}^{\theta})$ denote by $\mathcal{TC}^{\xi}(\mathcal{O}_{\xi}^{\theta})$ the set of time-interval configurations that are prefixes of $C_{\upsilon}(\mathbf{I}_{\upsilon}) \in \mathcal{TC}(\mathcal{O}_{\xi}^{\theta})$ and have the maximum upper limit for the execution of the last events greater than ξ.

We have that $C_{\ell}^{\xi}(\mathbf{I}_{\ell}) \in \mathcal{TC}^{\xi}(\mathcal{O}_{\xi}^{\theta})$ if:

1. $C_{\ell}^{\xi}(\mathbf{I}_{\ell}) \sqsubseteq C_{\ell}(\mathbf{I}_{\ell})$ for $C_{\ell}(\mathbf{I}_{\ell}) \in \mathcal{TC}(\mathcal{O}_{\xi}^{\theta})$ and

2. $\forall e \in ENABLED(C_{\ell}^{\xi}) \Rightarrow (U_{\ell}(e)) > \xi$

Then for each time-interval configuration $C_{\ell}^{\xi}(\mathbf{I}_{\ell}) \in \mathcal{TC}^{\xi}(\mathcal{O}_{\xi}^{\theta})$ impose the following constraints:

i) $\forall e_{\iota} \in {}^{\bullet}CUT(C_{\ell}^{\xi})$, $\kappa_{e_{\iota}} := \{\theta_{e_{\iota}} \leq \xi\}$ (the events in $E_{C_{\ell}^{\xi}}$ are executed before ξ)

ii) $\forall e_{\lambda} \in ENABLED(C_{\ell}^{\xi})$, $\kappa_{e_{\lambda}} := \{\vartheta_{e_{\lambda}} \geq \xi\}$ (the enabled events are executed after the global time ξ)

Denote by $\mathcal{TC}^{\xi}(\mathcal{O}_{\xi}^{\theta})$ the set of time interval configurations obtained imposing the constraints mentioned at $i)$ and $ii)$ above by running the Algorithm 11.

Given a time-interval configuration $C_{\ell}^{\xi}(\mathbf{I}) \in \mathcal{TC}^{\xi}(\mathcal{O}_{\xi}^{\theta})$ denote $CSC_{\iota}(\mathcal{O}_{\xi}^{\theta})$ the set of states (the state class) that correspond with the time interval configuration $C_{\ell}^{\xi}(\mathcal{O}_{\xi}^{\theta})$ where:

$$CSC_{\ell}^{\xi}(\mathcal{O}_{\xi}^{\theta}) = (M_{\ell}, \mathcal{FI}_{\ell}) \tag{5.31}$$

with:

i) $M_{\ell} = mark(C_{\ell}^{\xi})$

ii) and $\mathcal{FI}_{\ell} = \left\{ I_{\ell}(t) \mid t \in ENABLED(C_{\ell}^{\xi}) \right\}$

where $I_{\ell}(t) = [L_{\ell}(t), U_{\ell}(t)]$ with $L_{\ell}(t) = \mathtt{max}(\xi, \mathtt{max}_{e \in {}^{\bullet\bullet}t}(L_{\ell}(e)) + L_t^s$ and $U_{\ell}(t) = \mathtt{max}(\xi, \mathtt{max}_{e \in {}^{\bullet\bullet}t}(U_{\ell}(e)) + U_t^s$.

Denote by $\mathcal{CSC}^{\xi}(\mathcal{O}_{\xi}^{\theta})$ the set of all configuration state classes at the global time ξ:

$$\mathcal{CSC}^{\xi}(\mathcal{O}_{\xi}^{\theta}) = \left\{ CSC_{\ell}^{\xi}(\mathcal{O}_{\xi}^{\theta}) \mid C_{\ell}^{\xi} \in \mathcal{TC}^{\xi}(\mathcal{O}_{\xi}^{\theta}) \right\}$$

and by $\overline{\mathcal{S}}_{\xi}(\mathcal{O}_{\xi}^{\theta})$ the set of states that is obtained by the union of all states that are contained in the configuration state classes $CSC_{\ell}^{\xi}(\mathcal{O}_{\xi}^{\theta}) \in \mathcal{CSC}^{\xi}(\mathcal{O}_{\xi}^{\theta})$:

$$\overline{S}_\xi(\mathcal{O}^\theta_\xi) = \bigcup_{C^\xi_\ell \in \mathcal{TC}^\xi(\mathcal{O}^\theta_\xi)} CSC^\xi((\mathcal{O}^\theta_\xi))$$

Then we have:

Theorem 8. *Given a TPN $\langle \mathcal{N}^\theta, M^\theta_0 \rangle$ and an arbitrary observation $\mathcal{O}^\theta_\xi$ we have that $\overline{\mathcal{S}}_\xi(\mathcal{O}^\theta_\xi) = \mathcal{S}_\xi(\mathcal{O}^\theta_\xi)$.*

To prove this we need the following result:

Proposition 25. *Consider the atomic state class graph $ASCG(\mathcal{O}^\theta_\xi)$ derived given the received observation $\mathcal{O}^\theta_\xi$. Then we have that $\forall C^\xi_\ell(\mathbf{I}) \in \mathcal{TC}^\xi(\mathcal{O}^\theta_\xi)$:*

$$CSC^\xi_\ell(\mathcal{O}^\theta_\xi) = \bigcup_{\sigma \in \mathcal{E}(C^\xi_\ell(\mathbf{I}_\ell))} ASC_{\xi,\tau}(\mathcal{O}^\theta_\xi) \tag{5.32}$$

where $\tau = \phi(\sigma)$ and $ASC_0 \xrightarrow{\tau(\rho)} ASC_{\xi,\tau}(\mathcal{O}^\theta_\xi)$ for $\rho \in ASCG(\mathcal{O}^\theta_\xi)$.

Proof. The proof is straightforward. □

Based on Proposition 25 we have that $\overline{\mathcal{S}}_\xi \subseteq \mathcal{S}_\xi$. The proof that $\overline{\mathcal{S}}_\xi \supseteq \mathcal{S}_\xi$ is immediate based on Proposition 24. This completes the proof of Theorem 8.

Example 32. *Consider again the TPN in Fig. 5.4 where $\mathcal{T}_o = \{t_4, t_5, t_9\}$ and $l_o(t_4) = l_o(t_5) = l_o(t_9)$. Before receiving the first observation we derive (see Fig. 5.23):*

$$\widehat{\mathcal{TC}}(\texttt{obs}_1 \mid \mathcal{O}^\theta_\xi) = \left\{ \widehat{C}_\ell(\mathbf{I}) \mid \ell = 1, \ldots, 4 \right\}$$

Then consider that the first observed event occurs at the time $\theta_{obs} = 10$. $\widehat{C}_1(\mathbf{I}_1)$ and $\widehat{C}_3(\mathbf{I}_3)$ are discarded because $I_1(e_9)$ in $\widehat{C}_1(\mathbf{I}_1)$ contradicts $\theta_{obs} = 10 \notin [12, 24]$; similarly $I_1(e_5)$ in $\widehat{C}_3(\mathbf{I}_3)$ contradicts $\theta_{obs} = 10 \notin [15, 28]$.

For $\widehat{C}_2(\mathbf{I}_2)$ we impose the temporal constraint that $\theta_{e_4} = 10$ and obtain $\widehat{C}_2(\mathbf{I}_2)$ $I_2(e_1) = [3, 4]$, $I_2(e_2) = [6, 7]$, $I_2(e_4) = 10$, $I_2(e_6) = [4, 8]$, and $I_2(e_7) = [8, 16]$.

Similarly for $\widehat{C}_4(\mathbf{I}_4)$ we impose the temporal constraint that $\theta_{e_4} = 10$ and obtain $I_4(e_1) = [3, 4]$, $I_4(e_2) = [6, 7]$, $I_4(e_4) = 10$, $I_4(e_6) = [4, 8]$, $I_4(e_8) = [8, 16]$, and $I_4(e_9) = [12, 24]$. No constraint is imposed to e_9 since $L_4(e_9) > \theta_{obs_1} = 10$.

We now want to derive the plant behavior up to the time $\xi = 10$. From $\widehat{C}_2(\mathbf{I}_2)$,$\widehat{C}_4(\mathbf{I}_4)$ we derive the set of prefix time interval configurations $\mathcal{TC}^\xi_2(\mathbf{I}_2) = \left\{ C^\xi_{2_1}(\mathbf{I}_{2_1}), C^\xi_{2_2}(\mathbf{I}_{2_2}) \right\}$, respectively $\mathcal{TC}^\xi_4(\mathbf{I}_4) = \left\{ C^\xi_{4_1}(\mathbf{I}_{4_1}), C^\xi_{4_2}(\mathbf{I}_{4_2}) \right\}$ displayed in Fig. 5.24.

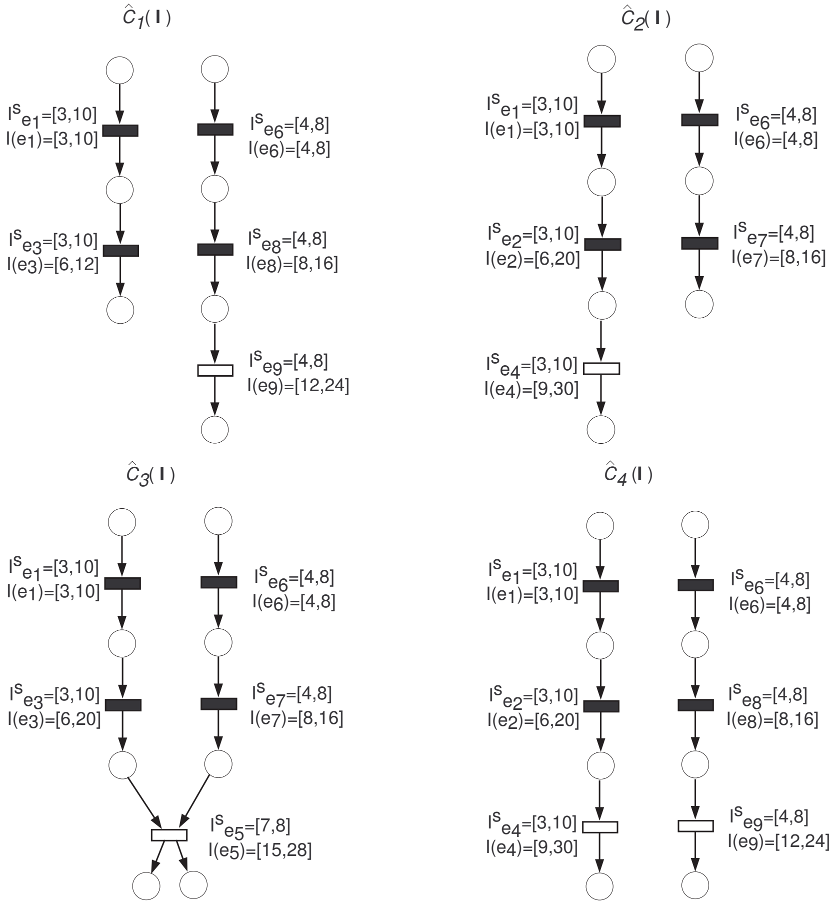


Figure 5.23:

- *for* $C^{\xi}_{2_1}(\mathbf{I}_{2_1})$ ($C^{\xi}_{4_2}$) *we impose the constraint that* $\theta_{e_7} \leq 10$ *and obtain:* $C^{\xi}_{2_1}(\mathbf{I}_{2_1}) : I_{2_1}(e_1) = [3,4]$, $I_{2_1}(e_2) = [6,7]$, $I_{2_1}(e_4) = 10$, $I_{2_1}(e_6) = [4,6]$, *and* $I_{2_1}(e_7) = [8,10]$.

  The firing domain of $CSC(C^{\xi}_{2_1}(\mathbf{I}))$ *is* $FI_{2_1}(e_1) =]10,13]$.

- *for* $C^{\xi}_{2_2}(\mathbf{I}_{2_2})$ *we impose the inequalities that* $\vartheta_{e_7} \geq 10$ *and* $\vartheta_{e_8} \geq 10$ *and obtain:*

  $C^{\xi}_{2_2}(\mathbf{I}_{2_2}) : I_{2_2}(e_1) = [3,4]$, $I_{2_2}(e_2) = [6,7]$, $I_{2_2}(e_4) = 10$, $I_{2_2}(e_6) = [4,8]$

  The firing domain of $CSC(C^{\xi}_{2_2}(\mathbf{I})_{2_2})$ *is:*

  $\mathcal{FI}(C^{\xi}_{2_2}(\mathbf{I})) = \{FI_{2_2}(e_1) =]10,13], FI_{2_2}(e_7) =]10,16], FI_{2_2}(e_8) =]10,16]\}$

- *for* $C^{\xi}_{4_1}(\mathbf{I}_{4_1})$ ($C^{\xi}_{2_2}(\mathbf{I}_{2_2})$) *we impose the constraint that* $\theta_{e_8} \leq 10$ *and obtain:*

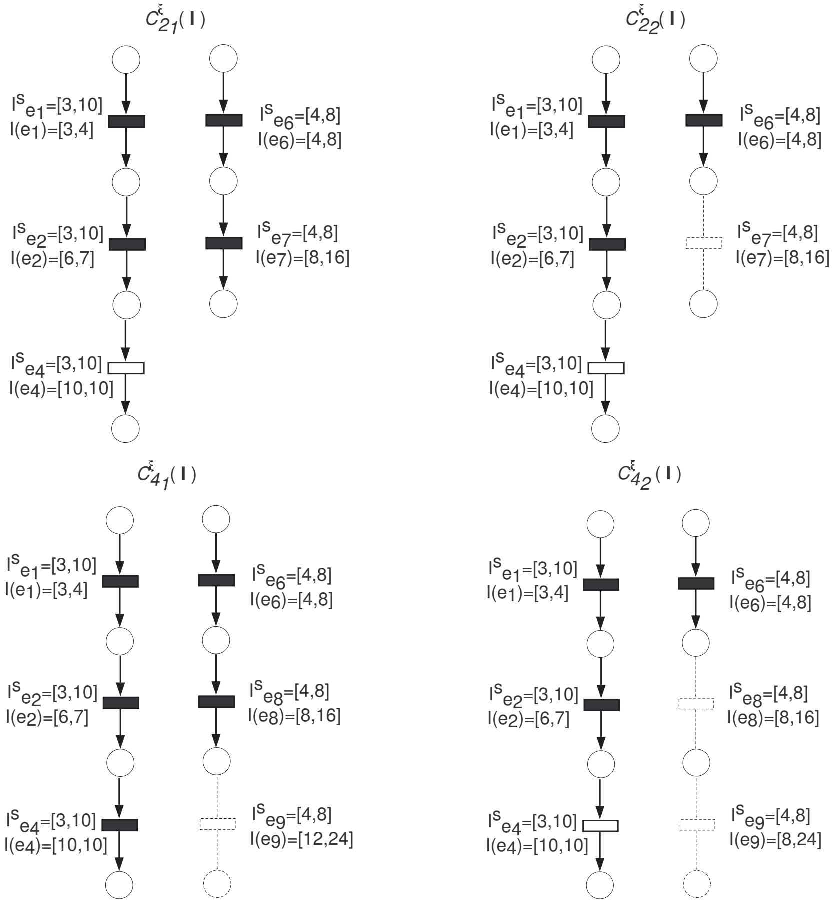


Figure 5.24:

$C^{\xi}_{4_1}(\mathbf{I}_{4_1}) : I_{4_1}(e_1) = [3,4]$, $I_{4_1}(e_2) = [6,7]$, $I_{4_1}(e_4) = 10$, $I_{4_1}(e_6) = [4,8]$, $I_{4_1}(e_8) = [8,10]$ *The firing domain of* $CSC(C^{\xi}_{4_1}(\mathbf{I}))$ *is:*

$$\mathcal{FI}(C^{\xi}_{4_1}(\mathbf{I})) = \{FI_{4_1}(e_1) =]10,13], FI_{4_1}(e_9) = [12,18]\}$$

Then consider the atomic state class $ASC_{\tau,\xi}$ *derived in Ex.28 for* $\tau = t_1t_2t_6t_4$. $ASC_{\tau,\xi} = (M, \mathcal{FI})$ *where:*

$$\mathcal{FI}(ASC_{\tau,\xi}) = \{FI(t_1) =]10,23], FI(t_7) =]10,16], FI(t_7) =]10,16]\}$$

We have that $\tau \in \langle E_{C^{\xi}_{2_2}} \rangle$ *and consequently* $\mathcal{FI}(ASC_{\tau,\xi}) = \mathcal{FI}(C^{\xi}_{4_1}(\mathbf{I}))$.

5.5.4 Checking for redundancy

Consider a time interval configuration $C_\ell(\mathbf{I}_\ell) \in \mathcal{TC}(\mathcal{O}_\xi^\theta)$. The problem in this section is to derive which inequalities can be deleted since they are redundant (and hence irrelevant) for the future calculations. To illustrate this consider a part of a fictitious time interval configuration $C_\ell(\mathbf{I}_\ell)$ in Fig. 5.25 where the dotted lines emerging places emphasize that e_3 and e_4 are the last events in C_ℓ $(e_3, e_4 \in {}^\bullet CUT(C_\ell))$.

To each event e_v, $v = 1, \ldots, 4$ the execution time interval $I_\ell(e_v)$ is attached; for e_3 and e_4 we have indicated also their static firing intervals.

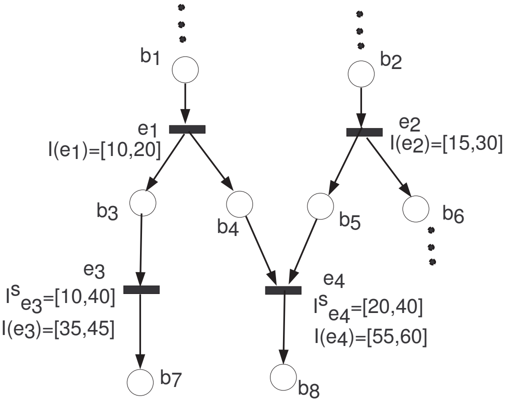


Figure 5.25:

Since $L_\ell(e_1) + U^s_{e_3} > U_\ell(e_3)$ and $U_\ell(e_1) + L^s_{e_3} < L_\ell(e_3)$ it is obvious that any temporal constraint applied to e_3 cannot modify $I_\ell(e_1)$. We say that e_1 is passive w.r.t. e_3. We must also check if e_1 is passive w.r.t. e_4; if this is also true we can remove θ_{e_1} from the set of path variables since the occurrence of e_1 can not be modified by a future observation.

For e_4 we have that:

$$\begin{aligned} \mathtt{max}(L_\ell(e_1), L_\ell(e_2)) + U^s_{e_4} \geq U_\ell(e_4) \\ \mathtt{max}(U_\ell(e_1), U_\ell(e_2)) + L^s_{e_4} \leq L_\ell(e_4) \end{aligned} \tag{5.33}$$

We claim that e_1 is passive w.r.t. e_4. This is true because further constraints that may be applied to e_2 may only decrease $U_\ell(e_2)$ or increase $L_\ell(e_2)$. In both cases the inequalities in 5.33 remain true. If an event-node e_v is passive w.r.t. all its predecessor event-nodes then the path variable θ_{e_v} can be removed from K_{C_ℓ}.

Since our calculations operate on the configuration structure, we remove the event-node e_v from C_ℓ and then remove all the event-nodes that have all their predecessor nodes being removed from C_ℓ.

Definition 61. *Consider a time-interval configuration $C_\ell(\mathbf{I}_\ell) \in \mathcal{TC}(\mathcal{O}_\xi^\theta)$. For $e_\upsilon \in {}^\bullet CUT(C_\ell)$ consider a temporal constraint $\kappa_{e_\upsilon} := \{\theta_{e_\upsilon} \in I'_\ell\}$ s.t. $I'_\ell(e_\lambda) \subset I_\ell(e_\lambda)$. $\mathcal{TC}_\ell(\mathbf{I}_\ell \mid \kappa_{e_\upsilon})$ is the set of all the time-interval configurations that are obtained by imposing κ_{e_υ} on $C_\ell(\mathbf{I}_\ell) \in \mathcal{TC}_\ell(\mathcal{O}_\xi^\theta)$:*

$$\mathcal{TC}_\ell(\mathbf{I}_\ell \mid \kappa_{e_\upsilon}) = \{C_{\nu_\ell}(\mathbf{I}_{\nu_\ell}) \mid \nu_\ell \in \mathcal{V}_\ell\}$$

where $\Gamma_\ell(\kappa_{e_\upsilon}) = \{\mathbf{I}_{\nu_\ell} \mid \nu_\ell \in \mathcal{V}_\ell\}$ is the set of $\mid E_{C_\ell} \mid$-hyperboxes that are obtained running the Algorithm 11 for $C_\ell(\mathbf{I}_\ell)$ and κ_{e_υ}.

Then we say that:

- e_γ is passive w.r.t. e_υ *if for any κ_{e_υ} we have that:*

$$\forall \mathbf{I}_{\nu_\ell} \in \Gamma_\ell(\kappa_{e_\upsilon}) \Rightarrow \mathbf{I}_{\nu_\ell}(e_\gamma) = \mathbf{I}_\ell(\mathbf{e}_\gamma)$$

e_γ is passive in $C_\ell(\mathbf{I}_\ell)$ *iff* $\forall e_\upsilon \in {}^\bullet CUT(C_\ell)$, e_γ is passive w.r.t. e_υ

In words e_γ is passive w.r.t. e_υ *means that constraining the occurrence of e_υ has no effect on the occurrence of e_γ. Then e_γ* is passive in $C_\ell(\mathbf{I}_\ell)$ *means that for any extension of $C_\ell(\mathbf{I}_\ell)$ and for any further temporal constraint applied to the events that are appended to $C_\ell(\mathbf{I}_\ell)$ we have that the time interval of the occurrence of the event e_γ remains unchanged.*

We have the following results that are needed for formally presenting the algorithm that eliminates the passive events within a configuration $C_\ell(\mathbf{I}_\ell)$.

Proposition 26. *Given a time interval configuration $C_\ell(\mathbf{I}_\ell) \in \mathcal{TC}(\mathcal{O}_\xi^\theta)$ and two events $\forall e_\gamma, e_\upsilon \in E_{C_\ell}$ s.t. $e_\gamma \prec e_\upsilon$ we have that: e_γ* is passive w.r.t. e_υ *if for any event $\forall e_\lambda \in E_{C_\ell}$ s.t. $e_\gamma \prec e_\lambda \prec e_\upsilon$ we have that e_λ* is passive w.r.t. e_υ.

Proof. Trivial. □

Proposition 27. *Given a time interval configuration $C_\ell(\mathbf{I}_\ell) \in \mathcal{TC}(\mathcal{O}_\xi^\theta)$ and two events $e_\gamma, e_\upsilon \in E_{C_\ell}$ s.t. $e_\gamma \in {}^{\bullet\bullet}e_\upsilon$ we have that e_γ is passive w.r.t. e_υ if the two inequalities in 5.34 are satisfied:*

$$\begin{aligned} \mathtt{max}_{e_\lambda \in {}^{\bullet\bullet}e_\upsilon}(L_\ell(e_\lambda)) + U^s_{e_\upsilon} &\geq U_\ell(e_\upsilon) \\ \mathtt{max}_{e_\lambda \in {}^{\bullet\bullet}e_\upsilon}(U_\ell(e_\lambda)) + L^s_{e_\upsilon} &\leq L_\ell(e_\upsilon) \end{aligned} \tag{5.34}$$

Proof. Trivial. □

Let in the following the event-set of E_{C_ℓ} be partitioned in disjunct subsets (layers) as follows:

- $Layer^\uparrow_\ell[0] = {}^\bullet CUT(C_\ell)$

- $Layer^\uparrow_\ell[y] = \left\{e_\upsilon \in E_{C_\ell} \mid e_\upsilon^{\bullet\bullet} \subseteq \bigcup_{z=0}^{y-1} Layer^\uparrow_\ell[z] \wedge e_\upsilon^{\bullet\bullet} \cap Layer^\uparrow_\ell[y-1] \neq \emptyset\right\}$

The number n_{max} of layers is finite since the number of events in $C_\ell(\mathbf{I}_\ell)$ is finite. The following Algorithm 17 eliminates the events from a time-interval configuration.

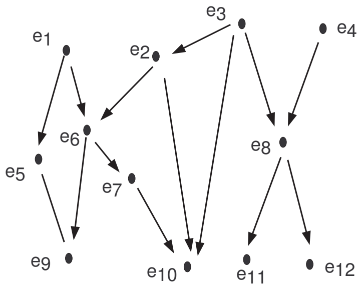
Algorithm 17 Eliminate passive event-nodes

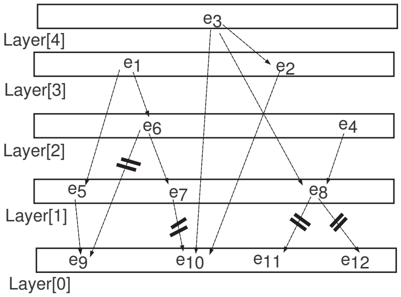
```
Require: $C_\ell(\mathbf{I}_\ell) \in \mathcal{TC}(\mathcal{O}^\theta_\xi)$
Ensure: $ELIM(C_\ell(\mathbf{I}_\ell))$
$KEEP(C_\ell(\mathbf{I}_\ell)) = Layer_\ell[0]$
$y = 1$
while $y \leq y_{max}$ do
   for all $e_v \in Layer^{\uparrow}_{\ell}[y]$ do
      for all $e_\gamma \in KEEP(C_\ell(\mathbf{I}_\ell)) \cap e_v^{\bullet\bullet}$ do
         if $\neg(e_v$ is passive w.r.t. $e_\gamma)$ then
            $KEEP(C_\ell(\mathbf{I}_\ell)) = KEEP(C_\ell(\mathbf{I}_\ell)) \cup \{e_v\}$
            exit loop for
         end if
      end for
   end for
   $y = y + 1$
end while
$ELIM(C_\ell(\mathbf{I}_\ell)) = E_{C_\ell} \setminus KEEP(C_\ell(\mathbf{I}_\ell))$
```

Example 33. *Consider the partial order between the events in a time-configuration $C_\ell(\mathbf{I}_\ell)$ as displayed in Fig. 5.5.4-left. The layer decomposition of E_{C_ℓ} is displayed in Fig. 5.5.4-right. Algorithm 17 works as follows. The set of event nodes $KEEP(C_\ell(\mathbf{I}_\ell))$ is initialized with the set of events contained in $Layer^{\uparrow}_{\ell}[0]$ since the last events in $C_\ell(\mathbf{I}_\ell)$ can not be found as passive. Then the counter y is set 1 and for all the events in $Layer^{\uparrow}_{\ell}[1]$ we check if the time-dependency between $e_v \in Layer^{\uparrow}_{\ell}[1]$ and the events in $KEEP(C_\ell(\mathbf{I}_\ell))$ that are the first successors of e_v is passive. If for e_v there exists a time-dependency that is not passive, the checking is stop since the node e_v cannot be declared passive. If all the nodes in $Layer[1]$ have been checked then the set of nodes in the next layer are checked. For instance if we have for e_5 in Fig. 5.5.4-right that e_5 is not passive w.r.t. e_9 then e_9 is added to $KEEP(C_\ell(\mathbf{I}_\ell))$, while e_7 and e_8 are not added to $KEEP(C_\ell(\mathbf{I}_\ell))$ if we have that their time-dependencies are passive as is graphically indicated by the two thick lines that cross the arrow between two events.*

For $Layer^{\uparrow}_{\ell}[2]$ we do not have to check e_4 since its only first successor e_8 does not belong to $KEEP(C_\ell(\mathbf{I}_\ell))$. Consequently e_4 is not added to $KEEP(C_\ell(\mathbf{I}_\ell))$. e_6 must only be checked against e_9. If e_6 is passive w.r.t. e_9 as is indicated then e_6 is not added to $KEEP(C_\ell(\mathbf{I}_\ell))$. Finally we obtain $KEEP(C_\ell(\mathbf{I}_\ell)) = \{e_1, e_2, e_3, e_5, e_9, e_{10}, e_{11}, e_{12}\}$. The set of events that can be removed is:

$$ELIM(C_\ell(\mathbf{I}_\ell)) = \{e_4, e_6, e_7, e_8\}$$





5.5.5 Time Processes for general Petri Nets

In this section we treat the general case of TPN models dropping the simplifying assumption that $\mathcal{N}$ should be free-choice net. By removing this assumption the analysis becomes more complicated because the firing of a transition t does not only depend on the time the tokens have arrived in its input places but the firing can also be pre-emptied by some other transitions that have different enabling conditions and that are simultaneously enabled.

The fundamental difference is that in a free-choice TPN if each of the input places of a transition t can become marked then no matter what time the tokens arrive in its input places, it eventually becomes enabled and moreover it can be fired while for a general TPN this is not true.

To illustrate this consider the TPN displayed in Fig. 5.26.

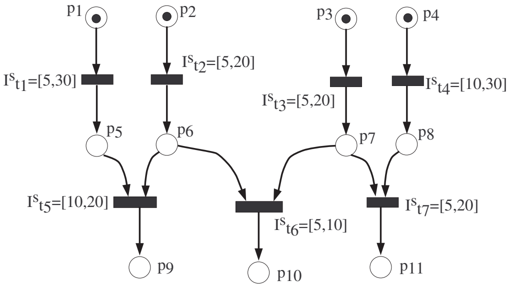


Figure 5.26:

We have that t_1, t_2, t_3, t_4 can be executed concurrently from the initial marking. Thus each of the places p_5, p_6, p_7, p_8 will eventually become marked (but notice that this does not imply that a marking M that considers tokens in p_5, p_6, p_7, p_8 is necessarily reachable in TPN). Let us calculate when transition t_5 can fire, assuming that t_1 fires at the time $\theta_{t_1} = 30$ and t_2 fires at $\theta_{t_2} = 5$. Thus the time t_5 becomes enabled would be in this case $\theta^{en}_{t_5} = 30$. However if the token from p_6 would have been removed by firing t_6, t_5 becomes disabled. The firing of t_6 requires beside the token in p_6 also a token in p_7 whose arrival time is given by the time t_3 is executed. Hence depending on the time t_3 and t_4 are executed (e.g. $\theta_{t_3} = \theta_{t_4} = 10$), t_7 may fire (e.g. at the time $\theta_{t_7} = 15$) and then t_6 becomes disabled; this means that t_5 becomes enabled at the time $\theta^{en}_{t_5} = 30$ and fires at some time $\theta_{t_5} \in [40, 50]$.

Whether t_5 becomes enabled or not (and consequently whether t_5 can fire or not) depends on the time when t_1, t_2, t_3, t_4 are executed, the possible firing time $\vartheta_{t_6} \in [\theta^{en}_{t_6} + L^s_{t_6}, \theta^{en}_{t_6} + U^s_{t_6}]$ for executing t_6, and the firing time that is chosen for executing t_7 ($\vartheta_{t_7} \in [\theta^{en}_{t_7} + L^s_{t_7}, \theta^{en}_{t_7} + U^s_{t_7}]$).

Hence in a general TPN the execution of a transition t depends on the arrival time of the tokens in the input places of $cluster(t)$ and on the firing times that are chosen within the firing domains of the transitions that are enabled in $cluster(t)$.

Given a configuration C_ℓ in a net unfolding $\mathcal{U}_\mathcal{N}(M_0)$ denote by $\breve{E}_{C_\ell}$ the set of all the events that could have been executed but were not executed in C_ℓ that is:

$$\breve{E}_{C_\ell} = \{\breve{e} \in E \setminus E_{C_\ell} \mid \phi(\breve{e}) \subseteq B_{C_\ell}\}$$

A conflicting event $\breve{e}$ is connected with its pre-set by dotted lines to emphasize that it is a pseudo-event. In other words e is an event whose occurrence would have disabled events in E_{C_ℓ} with which it shares some input places.

Definition 62. *Consider a general TPN model $\langle \mathcal{N}^\theta, M^\theta_0 \rangle$ and then let $\mathcal{U}_\mathcal{N}(M_0)$ be the unfolding of its untimed PN support $\langle \mathcal{N}, M_0 \rangle$. Denote by $\mathcal{C}$ the set of all (untimed) configurations in $\mathcal{U}_\mathcal{N}(M_0)$.*

Then denote by $C^\theta_\ell = (B_{C_\ell}, E_{C_\ell}, \preceq, I^s, \breve{E}_{C_\ell})$ the TPN endowed with a set of conflicting events $\breve{E}_{C_\ell}$ where:

1. *$C_\ell = (B_{C_\ell}, E_{C_\ell}, \preceq)$ is the untimed support*

2. *$\breve{E}_{C_\ell}$ the set of conflicting events*

3. *$I^s : E_{C_\ell} \cup \breve{E}_{C_\ell} \to \mathcal{I}(\mathbb{Q}_+)$ is the function that gives the static intervals where:*

   (a) for $e \in E_{C_\ell}$, $I^s(e) = I^s(t)$ where $t = \phi(e)$

   (b) for $e \in \breve{E}_{C_\ell}$, $I^s(\breve{e}) = U^s_t$ where $t = \phi(\breve{e})$

Denote by $K_{C_\ell^\theta}$ the characteristic system of C_ℓ^θ:

$$K_{C_\ell^\theta} = \begin{cases} \mathtt{max}_{e_\gamma \in {\bullet\bullet} e_\upsilon}(\theta_{e_\gamma}) + L^s_{e_\upsilon} \;\leq\; \theta_{e_\upsilon} \;\leq\; \mathtt{max}_{e_\gamma \in {\bullet\bullet} e_\upsilon}(\theta_{e_\gamma}) + U^s_{e_\upsilon} \\ \hfill \textit{for } e_\upsilon \in E_C \\ \bigvee\limits_{e_\gamma \sharp_1 \breve{e}_\lambda} \{\theta_{e_\gamma} \leq \vartheta_{\breve{e}_\lambda}\} \qquad\qquad \textit{for } \breve{e}_\lambda \in \breve{E}_{C_\ell} \end{cases} \tag{5.35}$$

$Sol(K_{C_\ell^\theta})$ is the set of solutions of C_ℓ^θ.

Consider $Sol_\nu(K_{C_\ell^\theta})$ a subset of $Sol(K_{C_\ell^\theta})$ and then denote $\mathbf{Y}_{\nu_\ell} = (\mathbf{I}_{\nu_\ell}, \breve{\mathbf{I}}_{\nu_\ell})$ the smallest $\mid E_{C_\ell} \mid + \mid \breve{E}_{C_\ell} \mid$-hyperbox that includes $Sol_{\nu_\ell}(K_{C^\theta})$.

We have that $C(\mathbf{Y}_{\nu_\ell})$ is a time interval configuration of $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ if $Sol_\nu(K_{C_\ell^\theta})$ has the time independence property for the concurrent events in E_{C_ℓ}.

Consider a general TPN $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ and let $C_\ell \in \mathcal{C}$ an untimed configuration in the net unfolding $\mathcal{U}_\mathcal{N}(M_0)$ of the untimed PN $\langle \mathcal{N}, M_0 \rangle$. For the TPN $C_\ell^\theta = (B_{C_\ell}, E_{C_\ell}, \preceq, I^s, \breve{E}_{C_\ell})$ we have that in general $Sol(K_{C_\ell^\theta})$ does not have the time independence property.

In the following we present an algorithm to derive a set of subsets $\left\{ Sol_{\nu_\ell}(K_{C_\ell^\theta}) \mid \nu_\ell \in \mathcal{V}_\ell \right\}$ of $Sol(K_{C_\ell^\theta})$ that is a cover of the entire solution set where each sub-set $Sol_{\nu_\ell}(K_{C_\ell^\theta})$ has the time independence property.

Basically we want to derive a set of $\mid E_{C_\ell} \mid + \mid \breve{E}_{C_\ell} \mid$-hyperboxes
$\Gamma = \{\mathbf{Y}_{\nu_\ell} \mid \nu_\ell \in \mathcal{V}_\ell\}$ s.t. $Sol_\nu(K_{C_\ell^\theta}) = Sol(K_{C_\ell^\theta}) \cap \mathbf{Y}_{\nu_\ell}$ for all $\nu_\ell \in \mathcal{V}_\ell$.

Consider that $Sol(K_{C_\ell^\theta})$ does not have the time independence property.

Denote $\mathcal{I}(\mathbb{Q}^+)$ the set of all intervals with non-negative rational limits. Then consider two intervals $I_1, I_2 \in \mathcal{I}_\mathbb{Q}$ where $I_i = [L_i, U_i]$ for $i = 1, 2$. We say as in Allen's algebra [All83] that:

I_1 `before` I_2 ($\Leftrightarrow$ I_2 `after` I_1) if $U_1 < L_2$

I_1 `meets` I_2 ($\Leftrightarrow$ I_2 `is met by` I_1) if $U_1 = L_2$

I_1 `overlaps` I_2 ($\Leftrightarrow$ I_2 `overlapped by` I_1) if $L_1 \leq L_2 \leq U_1 \leq U_2$

I_1 `during` I_2 ($\Leftrightarrow$ I_2 `includes` I_1) if $L_2 \leq L_1 \leq U_1 \leq U_2$

Given $C_\ell(\mathbf{Y}_\ell)$ we say that the pair $(e, \breve{e}) \in E_{C_\ell} \times \breve{E}_{C_\ell}$ is:

- *active* if $\{I_\ell(\breve{e})$ `overlaps` $I_\ell(e)\}$ or $\{I_\ell(\breve{e})$ `during` $I_\ell(e)\}$
- *not-active* if $\{I_\ell(e)$ `overlaps` $I_\ell(\breve{e})\}$
- *passive* if $\{I_\ell(e)$ `before` $I_\ell(\breve{e})\}$
- *canceling* if $\{I_\ell(\breve{e})$ `before` $I_\ell(e)\}$

We use the following notation:

- $ACTIVE(C_\ell(\mathbf{Y}_\ell))$ is the set of pairs $(e, \breve{e})$ that are *active*
- $NOTACT(C_\ell(\mathbf{Y}_\ell))$ is the set of pairs $(e, \breve{e})$ that are $not - active$
- $PASSIVE(C_\ell(\mathbf{Y}_\ell))$ is the set of pairs $(e, \breve{e})$ that are $passive$
- $CANCEL(C_\ell(\mathbf{Y}_\ell))$ is the set of pairs $(e, \breve{e})$ that are $canceling$

We have that:

- if $\exists \breve{e} \in \breve{E}_C$ s.t. $(e \in E_C$ and $e \sharp_1 \breve{e}) \Rightarrow ((e, \breve{e}) \in CANCEL(C_\ell(\mathbf{Y}_\ell)))$ then $C_\ell(\mathbf{Y}_\ell)$ is impossible
- if a conflicting event $\breve{e}$ is such that $\exists (e, \breve{e}) \in PASSIVE(C_\ell(\mathbf{Y}_\ell))$ then $\breve{e}$ can be removed from $\breve{E}_\ell$

Notice that a conflicting event $\breve{e}$ s.t. $(e, \breve{e}) \in NOTACT(C_\ell(\mathbf{Y}_\ell))$ is *not-active w.r.t.* e for $\mathbf{Y}_\ell$ but $(e, \breve{e})$ may become active for $\mathbf{Y}'_\ell \subset \mathbf{Y}_\ell$.

If the set of conflicting pairs in $ACTIVE(C_\ell(\mathbf{Y}_\ell))$ is not empty then we need to check whether it is possible to impose temporal-constraints allowing derivation for $C_\ell(\mathbf{Y}_\ell)$ of a finite set of time-configurations $\{C_\ell(\mathbf{Y}_{\nu_\ell}) \mid \nu_\ell \in \mathcal{V}_\ell\}$ such that:

1. $\forall \breve{e} \in \breve{E}_{C_\ell} \Rightarrow \exists\ e \in E_{C_\ell}$ such that $(e, \breve{e}) \in PASSIVE(C_\ell(\mathbf{Y}_{\nu_\ell})) \cup NOTACT(C_\ell(\mathbf{Y}_{\nu_\ell}))$
2. $Sol_{\nu_\ell}(K_{C_\ell^\theta}) = Sol(K_{C_\ell^\theta}) \cap \mathbf{Y}_{\nu_\ell}$ has the time independence property
3. and $Sol(K_{C_\ell^\theta}) = \bigcup_{\nu_\ell \in \mathcal{V}_{\nu_\ell}} Sol_{\nu_\ell}(K_{C_\ell^\theta})$

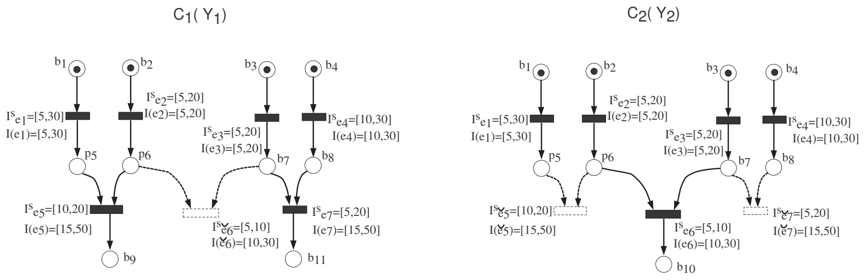


Figure 5.27:

Example 34. *Consider for TPN in Fig. 5.26 the two time-processes $C_1(\mathbf{Y}_1)$ and $C_2(\mathbf{Y}_2)$ displayed in Fig. 5.27. For $C(\mathbf{Y}_1)$ we have that $ACTIVE(C(\mathbf{Y}_1)) = \{(e_5, \breve{e}_6), (e_7, \breve{e}_6)\}$.*

Proposition 28. *$C_\ell(\mathbf{Y}_{\nu_\ell})$ is a time-interval configuration if $\forall \breve{e} \in \breve{E}_{C_\ell} \Rightarrow \exists\ e \in E_{C_\ell}$ s.t. either $(e, \breve{e}) \in PASSIVE(C_\ell(\mathbf{Y}_{\nu_\ell}))$ or $(e, \breve{e}) \in NOTACT(C_\ell(\mathbf{Y}_{\nu_\ell}))$.*

Proof. The proof is straightforward. □

Thus given $C_\ell(\mathbf{Y}_\ell)$ that is not a time interval configuration we should impose additional constraints such that for each conflicting event $\breve{e} \in E_{C_\ell}$ there is an event e s.t. $(e, \breve{e})$ is either $passive$ or $not - active$. In other words we should impose constraints such that all the conflicting events are deactivated.

Consider in the following a conflicting event $\breve{e}$ s.t. $(\forall e \in E_C)$, $e \sharp \breve{e} \Rightarrow (e, \breve{e}) \in ACTIVE(C_\ell(\mathbf{Y}_\ell))$ or $(e, \breve{e}) \in CANCEL(C_\ell(\mathbf{Y}_\ell))$. To deactivate $\breve{e}$ we have different policies where a policy corresponds to the selection of an event e s.t. $(e, \breve{e}) \in ACTIVE(C_\ell(\mathbf{Y}_\ell))$. Denote $POL(\breve{e})$ the set of all policies $\mathit{pol}(\breve{e}) = (e, \breve{e})$ to deactivate the conflicting event $\breve{e}$:

$$POL(\breve{e}) = \{\mathit{pol}(\breve{e}) = (e, \breve{e}) \mid (e, \breve{e}) \in ACTIVE(C_\ell(\mathbf{Y}_\ell))\}$$

Consider $\mathbf{Y}_{\nu_\ell} \subseteq \mathbf{Y}_\ell$. We have that $(e, \breve{e}) \in NOTACTIVE(C_\ell(\mathbf{Y}_{\nu_\ell})) \cup PASSIVE(C_\ell(\mathbf{Y}_{\nu_\ell}))$ if:

$$\{I_{\nu_\ell}(e)\ \mathtt{overlaps}\ I_{\nu_\ell}(\breve{e})\} \text{ or } \{I_{\nu_\ell}(e)\ \mathtt{before}\ I_{\nu_\ell}(\breve{e})\}$$

For $I_{\nu_\ell}(\breve{e}) = [L_{\nu_\ell}(\breve{e}), U_{\nu_\ell}(\breve{e})]$ and $I_{\nu_\ell} = [L_{\nu_\ell}(e), U_{\nu_\ell}(e)]$ we have that:

$$(I_{\nu_\ell}(e)\ \mathtt{overlaps}\ I_{\nu_\ell}(\breve{e})) \vee (I_{\nu_\ell}(e)\ \mathtt{before}\ I_{\nu_\ell}(\breve{e})) \Leftrightarrow$$
$$\Leftrightarrow (L_{\nu_\ell}(e) \leq L_{\nu_\ell}(\breve{e})) \wedge (U_{\nu_\ell}(e) \leq U_{\nu_\ell}(\breve{e}))$$

Given a conflicting event $\breve{e}$ that is active in $C_\ell(\mathbf{Y}_\ell)$ and a policy $\mathit{pol}(\breve{e}) = (e, \breve{e})$ to deactivate $\breve{e}$ ($\mathit{pol}(\breve{e}) \in POL(\breve{e})$), denote by $ACT(\mathit{pol}(\breve{e})$ the set of actions to deactivate $\breve{e}$ where:

$$ACT(\mathit{pol}(\breve{e})) = \{\kappa_{L_\ell}(\breve{e}), \kappa_{U_\ell}(e)\}$$

with:

1. $\kappa_{L_\ell(\breve{e})} := \{L'_\ell(\breve{e}) = \mathtt{max}(L_\ell(e), L_\ell(\breve{e}))\}$
2. $\kappa_{U_\ell(e)} := \{U'_\ell(e) = \mathtt{min}(U_\ell(e), U_\ell(\breve{e}))\}$

We have that $\kappa_{U_\ell(e)}$ is a constraint that applies only to e whereas $\kappa_{L_\ell(\breve{e})}$ applies to one of the predecessor events of $\breve{e}$, that is:

$$\kappa_{L_\ell(\breve{e})} := \bigvee_{e_\iota \in {}^{\bullet\bullet}\breve{e}} \{L'_\ell(e_\iota) := \mathtt{max}(L_\ell(e_\iota), L_\ell(\breve{e}) - U^s_{\breve{e}}))\}$$

Given a time-interval configuration $C_\ell(\mathbf{Y}_\ell)$ that is not valid denote:

1. $\mathcal{POL}(C_\ell(\mathbf{Y}_\ell)) = \Big\{pol(\breve{e}) \mid \breve{e} \in \breve{E}_{C_\ell} \wedge pol(\breve{e}) \in POL(\breve{e})\Big\}$ the set of all possible policies in $C_\ell(\mathbf{Y}_\ell)$
2. $\mathcal{ACT}(C_\ell(\mathbf{Y}_\ell)) = \{ACT(pol(\breve{e})) \mid pol(\breve{e}) \in \mathcal{POL}(C_\ell(\mathbf{Y}_\ell))\}$ and the set of all constraint actions in $C_\ell(\mathbf{Y}_\ell)$.

Consider $\kappa \in \mathcal{ACT}(C_\ell(\mathbf{Y}_\ell))$ and then denote by $\mathcal{TC}(\mathbf{Y}_\ell \mid \kappa)$ the set of time-interval configurations that are derived by imposing κ to $C_\ell(\mathbf{Y}_\ell)$, i.e. by running Algorithm 11 for $C_\ell(\mathbf{I}_\ell)$ and κ where $C_\ell(\mathbf{I}_\ell)$ is obtained from $C_\ell(\mathbf{I}_\ell, \breve{\mathbf{I}}_\ell)$ omitting the conflicting events $\breve{E}_{C_\ell}(\breve{\mathbf{I}}_\ell)$:

$$\mathcal{TC}(\mathbf{Y}_\ell \mid \kappa) = \{\mathcal{TC}(\mathbf{Y}_{\nu_\ell}) \mid \nu_\ell \in \mathcal{V}_\ell\}$$

For $C_\ell(\mathbf{Y}_{\nu_\ell})$ we have that:

- $ACTIVE(C_\ell(\mathbf{Y}_\ell)) \nsubseteq ACTIVE(C_\ell(\mathbf{Y}_{\nu_\ell}))$ and

- $(\mathbf{I}_{\nu_\ell}, \breve{\mathbf{I}}_{\nu_\ell}) \subset (\mathbf{I}_\ell, \breve{\mathbf{I}}_\ell)$ that is $\mathbf{I}_{\nu_\ell} \subseteq I_\ell$ and $\breve{\mathbf{I}}_{\nu_\ell} \subseteq \breve{I}_\ell$ and either $\mathbf{I}_{\nu_\ell} \subset \mathbf{I}_\ell$ or $\breve{\mathbf{I}}_{\nu_\ell} \subset \breve{\mathbf{I}}_\ell$.

Given a time-interval configuration $C_\ell(\mathbf{Y}_\ell)$ we present in the following a recursive algorithm to extract the set of time-interval configurations (valid time processes). The algorithm works as follows:

1. if there does not exist a conflicting event $\breve{e} \in \breve{E}_{C_\ell}$ that cancels $C_\ell(\mathbf{Y}_\ell)$ then:
   (a) choose an arbitrary event $\breve{e}$ such that $\forall e \in E_C$, $e \sharp \breve{e} \Rightarrow (e, \breve{e}) \in ACTIVE(C_\ell(\mathbf{Y}_\ell))$ or $(e, \breve{e}) \in CANCEL(C_\ell(\mathbf{Y}_\ell))$ and then for all policies $pol(\breve{e}) \in POL(\breve{e})$ to deactivate $\breve{e}$:
   - for all constraint-actions $\kappa \in ACT(pol)$
   - run Algorithm 11 for $C_\ell(\mathbf{Y}_\ell)$ and κ deriving $\mathcal{TC}(\mathbf{Y}_\ell \mid \kappa)$
2. then iterate the steps above for every $C_\ell(\mathbf{Y}_{\nu_\ell}) \in \mathcal{TC}(\mathbf{Y}_\ell \mid \kappa)$ until
   - either $\forall \breve{e} \in \breve{E}_C$, $\exists e \in E_C$, $e \sharp \breve{e}$ and $(e, \breve{e}) \in NOTACTIVE(C_\ell(\mathbf{Y}_{\nu_\ell}))$ or $(e, \breve{e}) \in PASSIVE(C_\ell(\mathbf{Y}_{\nu_\ell}))$
   - or $\breve{e}$ cancels $C_\ell(\mathbf{Y}_{\nu_\ell})$
   - or κ can not be imposed

Algorithm 18 Time-interval configurations for general TPN

```
Require: $C_\ell(\mathbf{Y}_\ell)$
Ensure: $\Gamma(\mathbf{Y}_\ell) = \{\mathbf{Y}_{\nu_\ell} \mid \nu_\ell \in \mathcal{V}_\ell\}$
$\mathcal{Y}_\ell = \{\mathbf{Y}_\ell\}$; $\Gamma(\mathbf{Y}_\ell) = \emptyset$
while $\mathcal{Y}_\ell \neq \emptyset$ do
    pick up $\mathbf{Y}_v \in \mathcal{Y}_\ell$ and then delete $\mathbf{Y}_v$ from $\mathcal{Y}_\ell$
    $ACTIVE(C_\ell(\mathbf{Y}_v))$, $NOTACTIVE(C_\ell(\mathbf{Y}_v))$, $CANCEL(C_\ell(\mathbf{Y}_v))$
    if $\breve{e} \in \breve{E}_C$ cancels $C_\ell(\mathbf{Y}_v)$ then
        remove $C_\ell(\mathbf{Y}_v)$ from $\Gamma(\mathbf{Y}_\ell)$
    else
        if $\forall \breve{e} \in \breve{E}_C \Rightarrow \exists e \in E_C$,$e\sharp\breve{e}$ and $(e,\breve{e}) \in NOTACTIVE(C_\ell(\mathbf{Y}_v))$ or
        $(e,\breve{e}) \in PASSIVE(C_\ell(\mathbf{Y}_v))$ then
            $\Gamma(\mathbf{Y}_\ell) = \Gamma(\mathbf{Y}_\ell) \cup \{\mathbf{Y}_v\}$
        else
            choose $\breve{e} \in \breve{E}_C$ s.t. $\forall e \in E_C$, $e\sharp\breve{e} \Rightarrow (e,\breve{e}) \in ACTIVE(C_\ell(\mathbf{Y}_v))$ or
            $(e,\breve{e}) \in CANCEL(C_\ell(\mathbf{Y}_v))$
            for all $pol(\breve{e}) \in POL(\breve{e})$ do
                for all $\kappa \in ACT(pol(\breve{e}))$ do
                    run Algorithm 11 for $C(\mathbf{Y}_v \mid \kappa)$
                    $\mathcal{TC}(\mathbf{Y}_v \mid \kappa) = \{C_\ell(\mathbf{Y}_{\nu_v}) \mid \nu_v \in \mathcal{V}_v\}$
                    add $\{\mathbf{Y}_{\nu_v} \mid \nu_v \in \mathcal{V}_v\}$ to $\mathcal{Y}_\ell$
                end for
            end for
        end if
    end if
end while
```

Consider a run of Algorithm 18 for the time interval configuration $C_\ell(\mathbf{Y}_\ell)$ and denote $\Gamma(\mathbf{Y}_\ell) = \{\mathbf{Y}_{\nu_\ell} \mid \nu_\ell \in \mathcal{V}_\ell\}$ the set of $\mid E_{C_\ell} \mid + \mid \breve{E}_{C_\ell} \mid$-hyperboxes.

Denote ϖ_{ν_ℓ} the set of choices that are made running Algorithm 18 for deriving $\mathbf{Y}_{\nu_\ell} \in \Gamma(\mathcal{Y}_\ell)$. Let $\mathcal{CHOICE}^*(C_\ell(\mathbf{Y}_\ell))$ be the abstract space of all possible sequences of choices and let ϖ be a generic sequence of choices. Denote by $\Gamma(\mathbf{Y}_\ell, \varpi)$ the set of $\mid E_{C_\ell} \mid + \mid \breve{E}_{C_\ell} \mid$-hyperboxes derived considering the choices given by ϖ.

Theorem 9. *Given a time-process $C_\ell(\mathbf{Y}_\ell)$ and an arbitrary sequence of choices $\varpi \in \mathcal{CHOICE}^*(C_\ell(\mathbf{Y}_\ell))$ made running Algorithm 18 we have that:*

(1) *ϖ is of finite length*

(2) *$\forall \mathbf{Y}_{\nu_\ell} \in \Gamma(\mathbf{Y}_\ell, \varpi)$ then $C(\mathbf{Y}_{\nu_\ell})$ is a time-interval configuration*

(3) $Sol(K_{C^\theta_\ell}) = \bigcup_{\mathbf{Y}_{\nu_\ell} \in \Gamma(\mathbf{Y}_\ell, \varpi)} Sol(K_{C^\theta_\ell}) \cap \mathbf{Y}_{\nu_\ell}$

Proof. The proof of (1) is simple. Any constraint κ that is imposed to $\mathbf{Y}_\upsilon$ results in a set of $\mid E_{C_\ell} \mid$-hyperboxes $\{\mathbf{Y}_{\nu_\upsilon} \mid \nu_\upsilon \in \mathcal{V}_\upsilon\}$ where $\forall \nu_\upsilon \in \mathcal{V}_\upsilon$, $\mathbf{Y}_{\nu_\upsilon} \subset \mathbf{Y}_\upsilon$. If the constrain κ has positive integer limits the proof is straighforward that is also the case when κ has rational limits (by multiplying all the coefficients in the system). Thus ϖ is of finite length.

The proof of (2) is based on Proposition 28.

The proof of (3) is as follows. Running Algorithm 18 a conflicting event is chosen and then all the policies to deactivate $\breve{e}$ are considered one at the time, namely for all $pol(\breve{e}) \in POL(\breve{e})$ we consider a constraint $\kappa \in ACT(pol(\breve{e}))$ that is applied to $\mathbf{Y}_\upsilon$. The statement is proven straightforwardly by induction. □

5.5.6 Centralized diagnosis for general Time Petri Nets based on time processes

For a general TPN model the monitoring algorithm that we propose is a bit different from the one presented for the case of free-choice nets. The difference is that the diagnoser does not make calculations to anticipate the observation but waits first to have an observed event and then it makes the calculations to explain the received observation.

Consider the first event that was observed in the plant $\mathcal{O}_1^\theta = \langle obs_1, \theta_{obs_1} \rangle$. We can calculate as for free-choice nets the set of time-interval configurations $\mathcal{TC}(\mathcal{O}_1^\theta)$ where $C_\ell(\mathbf{I}_\ell) \in \mathcal{TC}(\mathcal{O}_1^\theta)$ s.t.:

- $\exists e^o \in E_{C_\ell}$ s.t. $\ell(\phi(e^o)) = obs_1$ and $\theta_{obs_1} \in I(e^o)$

- $e \in E_{C_\ell} \setminus \{e^o\}$, $\phi(e) \in \mathcal{T}_{uo}$

- $\forall e \in ENABLED(C_\ell(\mathbf{I}_\ell))$

  1. if $\phi(e) \in \mathcal{T}_{uo}$ then $U_\ell(e) > \theta_{obs_1}$
  2. if $\phi(e^o) \in \mathcal{T}_o$ then $L_\ell(e) > \theta_{obs_1}$

For each $C_\ell(\mathbf{I}_\ell) \in \mathcal{TC}(\mathcal{O}_1^\theta)$ consider the set of conflicting events $\breve{E}_{C_\ell}$ and then let $C_\ell(\mathbf{Y}_\ell)$ be obtained in this way. Taking into account the received observation $\mathcal{O}_1^\theta$ the characteristic system K_{C_ℓ} has the form:

$$K_{C_\ell^\theta} = \begin{cases} \max\limits_{e' \in {}^{\bullet\bullet}e}(\theta_{e'}) + L_e^s \leq \ \theta_e \leq \max\limits_{e' \in {}^{\bullet\bullet}e}(\theta_{e'}) + U_e^s & e \in E_{C_\ell} \\ \mathtt{min}_{e' \sharp_1 \breve{e}}(\theta_{e'}) \leq \max\limits_{e'' \in {}^{\bullet\bullet}\breve{e}}(\theta_{e''}) + U_{\breve{e}}^s & \breve{e} \in \breve{E}_{C_\ell} \\ \theta_{e^o} = \theta_{obs_1} \text{ for } \phi(e^o) = t^o \text{ and } \ell(t^o) = obs_1 & \\ \theta_{e'^o} \geq \theta_{obs_1} \text{ for all } e'^o \in ENABLED(C_\ell) & \end{cases} \tag{5.36}$$

Proposition 29. *Given the observation generated by the plant $\mathcal{O}_1^\theta$ we have that $\tau^\theta \in \mathcal{L}^\theta_{\mathcal{N}^\theta}(\mathcal{O}_1^\theta)$ iff:*
$i)$ $\tau = \phi(\sigma)$, $\sigma \in \langle E_{C_\ell} \rangle_\preceq$ and $C_\ell \in \mathcal{C}(\mathcal{O}_1)$
$ii)$ Θ is a solution of $K_{C_\ell^\theta}$
$iii)$ $\forall e \in E_{C_\ell} \setminus \{e_1^o\} \Rightarrow \theta_e \leq \theta_{obs_1}$

Proof. $\Rightarrow$ Consider a time trace $\tau^\theta \in \mathcal{L}^\theta_{\mathcal{N}^\theta}(\mathcal{O}_1^\theta)$. $\tau^\theta = \langle t_1, \theta_{t_1} \rangle, \ldots, \langle t_\lambda, \theta_{t_\lambda} \rangle$ Clearly there is an untimed configuration $C_\ell \in \mathcal{C}(\mathcal{O}_1)$ s.t. $\tau = \phi(\sigma)$, $\sigma \in \langle E_{C_\ell} \rangle_\preceq$. Since the TPN is untimed 1 safe we have that E_{C_ℓ} is unique up to isomorphism and moreover we do not have transitions that are multiply enabled.

We show that $\Theta = (\theta_{t_1}, \ldots, \theta_{t_\lambda}) = (\theta_{e_1}, \ldots, \theta_{e_\lambda})$ ($t_\iota = \phi(e_\iota)$ for $\iota = 1, \ldots, \lambda$) is a solution of the characteristic system $K_{C_\ell^\theta}$. Consider the first transition t_1. Since $\langle t_1, \theta_{t_1} \rangle$ is legal we have that $L^s_{t_1} \leq \theta_{t_1} \leq U^s_{t_1}$ and $\forall t' \in ENABLED(M_0)$, $\theta_{t_1} \leq U^s_{t'}$. We have that there is an unique event $e_1 \in E_C$ s.t. $\phi(e_1) = t_1$ and $^\bullet e = \emptyset$. Thus we have that $L^s_{e_1} \leq \theta_{e_1} \leq U^s_{e_1}$. Then consider an event $\breve{t}$ that is enabled from the initial marking M_0^θ and $^\bullet\breve{t} \cap {}^\bullet t_1 \neq \emptyset$. Similarly there is an unique conflicting event $\breve{e} \in \breve{E}_{C_\ell}$ s.t. $\phi(\breve{e})$ and $^\bullet\breve{e} = \emptyset$. Obviously $\theta_{t_1} \leq U^s_{\breve{t}}$ that means that $\theta_{e_1} \leq U^s_{\breve{e}}$. Then inductively in the length of the trace τ it is easy to prove that the inequalities in $K_{C_\ell^\theta}$ are satisfied that implies that Θ is a solution of $K_{C_\ell^\theta}$. $iii)$ is trivial.

$\Leftarrow$ Consider $C_\ell(\Theta)$ s.t. $i)$, $ii)$ and $iii)$ are satisfied. The proof that τ^θ is a legal trace is straightforward by induction, proving that the enabling and firing conditions are satisfied. □

Remark 20. *Condition $iii)$ eliminates solutions $\Theta \in Sol(K_{C_\ell^\theta})$ such that Θ considers unobservable events $e \in E_{C_\ell}$, $\phi(e) \in \mathcal{T}_{uo}$ (that are concurrent with the last observed event e^o) with execution times θ_e that are bigger that θ_{obs_1}. However for practical calculations condition $iii)$ can be dropped since the consideration of events that will be executed after θ_{obs_n} can be seen as a prognosis. Notice that at item 9) in the setting the faults are assumed unpredictable, thus at the time θ_{obs_1} only fault events that are executed before θ_{obs_1} can be diagnosed that for sure happened.*

For each $C_\ell(\mathbf{Y}_\ell)$ we obtain running Algorithm 18 the set of time interval configurations $\{C_\ell(\mathbf{Y}_{\nu_\ell}) \mid \nu \in \mathcal{V}^{obs_1}_{\nu_\ell}\}$.

By a recursive calculation we obtain the set of time-interval configurations:

$$\mathcal{TC}(\mathcal{O}_n^\theta) = \{C_\ell(\mathbf{Y}_{\nu_\ell}) \mid \nu_\ell \in \mathcal{V}^{obs_n}_{\nu_\ell}\}$$

derived after receiving the observation $\mathcal{O}_n^\theta$.

Denote by $\mathcal{E}(\mathcal{O}_n^\theta)$ the set of untimed traces represented by the set of time interval configurations $\mathcal{TC}(\mathcal{O}_n^\theta)$ calculated at the time θ_{obs_n}:

$$\mathcal{E}(\mathcal{O}_n^\theta) = \{\mathcal{E}(C(\mathbf{Y})) \mid C(\mathbf{Y}) \in \mathcal{TC}(\mathcal{O}_n^\theta)\}$$

and $\mathcal{L}^{gen}(\mathcal{O}_n^\theta) = \{\tau = \phi(\sigma) \mid \sigma \in \mathcal{E}(\mathcal{O}_n^\theta)\}$

Proposition 30. *Given $\mathcal{L}_{\mathcal{N}^\theta}(\mathcal{O}_n^\theta)$ (the untimed support language of $\mathcal{L}^\theta_{\mathcal{N}^\theta}(\mathcal{O}_n^\theta)$), and $\mathcal{L}^{gen}(\mathcal{O}_b^\theta)$ we have that:*

$$\mathcal{L}_{\mathcal{N}^\theta}(\mathcal{O}_n^\theta) \equiv_{\Sigma_\mu} \mathcal{L}^{gen}(\mathcal{O}_n^\theta)$$

that is:

i) $\forall \tau \in \mathcal{L}_{\mathcal{N}^\theta}(\mathcal{O}_n^\theta) \Rightarrow \exists \tau' \in \mathcal{L}^{gen}(\mathcal{O}_n^\theta)$ *s.t.* $\Sigma_\mu(\tau) = \Sigma_\mu(\tau')$ *and*

ii) $\forall \tau' \in \mathcal{L}^{gen}(\mathcal{O}_n^\theta) \Rightarrow \exists \tau \in \mathcal{L}_{\mathcal{N}^\theta}(\mathcal{O}_n^\theta)$ *s.t.* $\Sigma_\mu(\tau) = \Sigma_\mu(\tau')$

Proof. Based on Theorem 9 we have that running Algorithm 18 $\forall \mathbf{Y}_{\nu_\ell} \in \Gamma(\mathbf{Y}_\ell, \varpi)$, $C(\mathbf{Y}_{\nu_\ell})$ is a time-interval configuration and

$$Sol(K_{C_\ell^\theta}) = \bigcup_{\mathbf{Y}_{\nu_\ell} \in \Gamma(\mathbf{Y}_\ell, \varpi)} Sol(K_{C_\ell^\theta}) \cap \mathbf{Y}_{\nu_\ell}$$

Since the conflicting events are deactivated the proof is straightforward based on Proposition 24. □

Denote by $\mathcal{DR}^{gen}(\mathcal{O}_n^\theta)$ the diagnosis result derived using $\mathcal{L}^{gen}(\mathcal{O}_n^\theta)$. Then we have the following result:

Theorem 10. *Given a general TPN model $\langle \mathcal{N}^\theta, M_0^\theta \rangle$ and the observation generated by the plant $\mathcal{O}_n^\theta$, we have that:*

$$\mathcal{DR}(\mathcal{O}_n^\theta) = \mathcal{DR}^{gen}(\mathcal{O}_n^\theta) \tag{5.37}$$

Proof. The proof is straightforward based on Proposition 30 □

5.6 Backward Time Processes for free choice TPNs

In this chapter we present for the class of free-choice Time Petri Nets an alternative method to derive the set of time-interval configurations based on the backward calculation of the set of minimal time-interval configurations.

As for the untimed case the backward calculation is less sensitive to uncertainty in the knowledge of the initial marking. Thus the main purpose for developing a method based on a backward search is to use it in a distributed setting where the marking of a local site is uncertain because of the tokens that can unobservably enter/exit the local site TPN model via the common border places.

Let the first received observation be $\mathcal{O}_1^\theta = \langle obs_1, \theta_{obs_1} \rangle$. For each observable event t^o s.t. $l(t^o) = obs_1$ consider the set of backward untimed unfolding $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(t^o)$ calculated as presented in Section 3.2.2 with the only difference that during computation the configurations that violate the 1-safe property of the net are discarded.

The set $\overleftarrow{\mathcal{C}}(t^o)$ contains all the minimal untimed configurations with the first observed event as $t^o = l^{-1}(obs_1)$.

Denote $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(obs_1)$ the set of untimed backward net unfoldings:

$$\overleftarrow{\mathcal{U}}_{\mathcal{N}}(obs_1) = \left\{ \overleftarrow{\mathcal{U}}_{\mathcal{N}}(t^o) \mid t^o \in \mathcal{T}_o \wedge l(t^o) = obs_1 \right\}$$

and denote by $\overleftarrow{\mathcal{C}}(obs_1)$ the union of all the configurations in $\overleftarrow{\mathcal{U}}_{\mathcal{N}}(obs_1)$:

$$\overleftarrow{\mathcal{C}}(obs_1) = \left\{ \overleftarrow{\mathcal{C}}(t^o) \mid t^o \in \mathcal{T}_o \wedge l(t^o) = obs_1 \right\}$$

Consider a minimal untimed configuration $\overleftarrow{C_\ell} \in \overleftarrow{\mathcal{C}}(obs_1)$. The following algorithm endows the events that are executed within $\overleftarrow{C_\ell}$ with timing information.

Partition the event-set $\overleftarrow{E}_{\overleftarrow{C_\ell}}$ in disjunct subsets (layers) as follows:

- $Layer^{\downarrow}[0] = \left\{ e \in \overleftarrow{E}_{\overleftarrow{C_\ell}} \mid {}^\bullet e = \emptyset \right\}$
- $Layer^{\downarrow}[y] = \left\{ e \in \overleftarrow{E}_{\overleftarrow{C_\ell}} \mid {}^{\bullet\bullet} e \subseteq \bigcup_{z=0}^{y-1} Layer^{\downarrow}[z] \ \wedge \ {}^{\bullet\bullet} e \cap Layer^{\downarrow}[y-1] \neq \emptyset \right\}$

The number of layers is finite (denoted y_{max}) since the number of events in $\overleftarrow{E}_{\overleftarrow{C}_\ell}$ is finite. Obviously we have that:

$$Layer^{\downarrow}[y_{max}] = \left\{ e^o \in \overleftarrow{E}_{\overleftarrow{C}_\ell} \mid \phi(e^o) = t^o \right\}$$

Algorithm 19 below verifies a backward minimal configuration $\overleftarrow{C}_\ell$ starting from the top events (the events that have no predecessors $e \in Layer[0]^{\downarrow}$) deriving the time-intervals for the execution of the events in $\overleftarrow{C}_\ell$.

If the execution time-interval of the observed event does not include the occurrence time that was reported then $\overleftarrow{C}_\ell$ is discarded. Otherwise the temporal constraint that the observed event t^o was executed at the time θ_{obs_1} is imposed and the set of time-interval configurations $\overleftarrow{C}(\mathbf{I}) \in \overleftarrow{\mathcal{TC}}(\mathcal{O}_1^\theta)$ is derived.

Algorithm 19 Backward Time Process

```
Require: $\overleftarrow{\mathcal{C}}(\mathcal{O}_1^\theta)$
Ensure: $\overleftarrow{\mathcal{TC}}(\mathcal{O}_1^\theta)$
for all $\overleftarrow{C}_\ell \in \overleftarrow{\mathcal{C}}(\mathcal{O}_1^\theta)$ do
  for $y = 0$ to $y_{max}$ do
    for all $e \in Layer_\ell^{\downarrow}[y]$ do
      $L_\ell(e) = \mathtt{max}_{e' \in \bullet\bullet e}(L_\ell(e')) + L_e^s$
      if $L_\ell(e) > \theta_{obs_1}$ then
        abort $\overleftarrow{C}_\ell$
      end if
      $U_\ell(e) = \mathtt{max}_{e' \in \bullet\bullet e}(U_\ell(e')) + U_e^s$
    end for
  end for
  $\kappa_{obs_1} := L_\ell(e^o) = U_\ell(e^o) = \theta_{obs_1}$
  run Algorithm 11 for $\overleftarrow{C}_\ell(\mathbf{I}_\ell)$ and $\kappa_{obs_1}$
  $\overleftarrow{\mathcal{TC}}_\ell(\mathcal{O}_1^\theta) = \left\{ \overleftarrow{C}_\ell(\mathbf{I}_{\nu_\ell}) \mid \nu_\ell \in \mathcal{V}_\ell^{obs_1} \right\}$
end for
$\overleftarrow{\mathcal{TC}}(\mathcal{O}_1^\theta) = \bigcup_{\ell \in \mathcal{V}} \overleftarrow{\mathcal{TC}}_\ell(\mathcal{O}_1^\theta)$
```

Consider in the following a minimal time-interval configuration $\overleftarrow{C}_v(\mathbf{I}_v) \in \overleftarrow{\mathcal{TC}}(\mathcal{O}_1^\theta)$. We calculate for $\overleftarrow{C}_v(\mathbf{I}_v)$ all its unobservable time-interval configuration extensions $C_\lambda(\mathbf{I}_\lambda)$ ($\overleftarrow{C}_v(\mathbf{I}_v) \sqsubset C_\lambda(\mathbf{I}_\lambda)$) s.t. all the enabled events in $C_\lambda(\mathbf{I}_\lambda)$ have their upper limits greater than θ_{obs_1}.

We have that $C_\lambda(\mathbf{I}_\lambda) \in \mathcal{TC}(\mathcal{O}_1^\theta)$ iff:

1. $\exists \overleftarrow{C}_v(\mathbf{I}_v) \in \overleftarrow{\mathcal{TC}}(\mathcal{O}_1^\theta)$ and $\overleftarrow{C}_v(\mathbf{I}_v) \sqsubset C_\lambda(\mathbf{I}_\lambda)$
2. $\forall e \in E_{C_\lambda} \setminus E_{\overleftarrow{C}_v} \Rightarrow \phi(e) \in \mathcal{T}_{uo}$
3. $\forall e \in ENABLED(C_\lambda)$
   (a) if $\phi(e) \in \mathcal{T}_{uo}$ then $U_\lambda(e) > \theta_{obs_1}$
   (b) if $\phi(e^o) \in \mathcal{T}_o$ then $L_\lambda(e) > \theta_{obs_1}$

To extend this result for a sequence of observed events $\mathcal{O}_n^\theta = \langle obs_1, \theta_{obs_1}\rangle, \ldots, \langle obs_n, \theta_{obs_n}\rangle$ let $\overleftarrow{\mathcal{TC}}(\mathcal{O}_n^\theta)$ and $\mathcal{TC}(\mathcal{O}_n^\theta)$ be derived recursively in the straightforward manner. Obviously $\mathcal{TC}(\mathcal{O}_n^\theta)$ contains the same set of time-interval configurations that would have been derived forward by running Algorithm 16 without calculating the extensions of the time-interval configurations up to the next discarding time that corresponds with the $n+1^{th}$ observed event.

5.7 Distributed diagnosis for Time Petri Nets

In this section we extend the distributed diagnosis algorithm presented in Section 4.3 for the case when each component is modeled by a TPN. The distributed setting and the requirements for the distributed algorithm are similar as for the untimed models. The differences are that the overall plant TPN model is untimed 1-safe and free-choice net, and that within each component, every oriented path that starts in a input border place and ends in an output border place includes at least one observable event.

5.7.1 The distributed setting

The distributed plant description is as follows:

1. $\langle \mathcal{N}, M_0\rangle$ is 1-safe
2. $\mathcal{N}$ is free-choice
3. $\mathcal{N}^\theta = \bigcup_{i\in J} \mathcal{N}_i^\theta$ where $\mathcal{N}^\theta = (\mathcal{P}, \mathcal{T}, F, I^s)$ and for $i \in J$
   $\mathcal{N}_i^\theta = (\mathcal{P}_i, \mathcal{T}_i, F_i, I_s^s)$
4. $\mathcal{P} = \bigcup_{i\in J} \mathcal{P}_i$, $\forall i \in J, \exists j \in J, i \neq j$ s.t. $\mathcal{P}_i \cap \mathcal{P}_j \triangleq \mathcal{P}_{ij} \neq \emptyset$
5. $\mathcal{T} = \bigcup_{i\in J} \mathcal{T}_i$, $\forall i,j \in I, i \neq j \Rightarrow \mathcal{T}_i \cap \mathcal{T}_j = \emptyset$
6. $F_i = F\mid_{\mathcal{N}_i}$
7. $\mathcal{P}_{ij} = \mathcal{P}_{IN_{ij}} \cup \mathcal{P}_{OUT_{ij}}$, $\mathcal{P}_{IN_{ij}} \cap \mathcal{P}_{OUT_{ij}} = \emptyset$

8. $\mathcal{P}_{IN_{ij}} = \mathcal{P}_{OUT_{ji}} = \{p \in \mathcal{P}_{ij} \mid p^\bullet \subseteq \mathcal{T}_i \wedge {}^\bullet p \subseteq \mathcal{T}_j\}$

9. $\mathcal{P}_{IN_{ji}} = \mathcal{P}_{OUT_{ij}} = \{p \in \mathcal{P}_{ji} \mid {}^\bullet p \subseteq \mathcal{T}_i \wedge p^\bullet \subseteq \mathcal{T}_j\}$

10. in each component $i \in J$ every oriented path that starts in an input border place $p_{IN_i} \in \mathcal{P}_{IN_i}$ and ends in an output border place $p_{OUT_i} \in \mathcal{P}_{OUT_i}$ includes at least one observable event

For simplicity we assume that $\mathcal{P}_{IN_{ij}}$ and $\mathcal{P}_{OUT_{ij}}$ are disjunct and we consider $M_{0_{ij}} = 0$ ($M_{0_{ij}} = M_0(\mathcal{P}_{12})$, $\forall i, j \in J$). Moreover to avoid unnecessary complications we consider that $\forall p \in \mathcal{P}$, ${}^\bullet p \subseteq \mathcal{T}_i$ for some $i \in J$ and similarly $\forall p \in \mathcal{P}$, $p^\bullet \subseteq \mathcal{T}_j$ for some $j \in J$.

Given the set of agents $\mathcal{AG} = \{Ag_i \mid i \in J\}$, the knowledge an agent Ag_i has $KNW_i = \langle \mathcal{N}_i^\theta, M_0^\theta, \mathcal{T}_{o_i}, \mathcal{T}_{F_i}, \mathcal{P}_{IN_i}, \mathcal{P}_{OUT_i} \rangle$ considers that:

i) the plant observation is distributed $\mathcal{O}_\xi^\theta = \otimes_{i \in J}^{gc} \mathcal{O}_\xi^{\theta_i}$ where

$$\mathcal{O}_\xi^{\theta_i} = \langle obs_{1_i}, \theta_{obs_{1_i}} \rangle, \ldots, \langle obs_{n_i}, \theta_{obs_{n_i}} \rangle$$

is the local observation recorded at site $i \in J$ up to the global time ξ and $\langle obs_{k_i}, \theta_{obs_{k_i}} \rangle$ indicates that an observable transition $t_i^o \in \mathcal{T}_{o_i}$ s.t. $l(t_i^o) = obs_{k_i}$ happened in component i at the time $\theta_{obs_{k_i}}$ measured with perfect accurracy according to a global clock (denoted *gc*).

ii) the communication between agents is not event-driven that is the agents are allowed to communicate at time e.g. $\theta_{com_1}, \theta_{com_2} \ldots$ that do not necessarily depend on the plant observation.

Problem formulation: Similarly as for the distributed diagnosis of the untimed models we have that given the setting described above, design a distributed algorithm such that:

R1) before communicating with its neighboring agents, each agent Ag_i ($i \in J$) derives a local preliminary diagnosis of the local site i

R2) at the global time θ_{com} (that does not necessarily depend on the time the observable events are reported):

2.1) each local agent derives the (limited) information that should be sent to the neighboring agents for achieving the consistency of the local calculations

2.2) the local calculation of site i is updated when new information is received

R3) then each local agent iterates the step 2.1) and 2.2) until a stopping criterion is achieved (the communication protocol terminates)

R4) the completion of the communication protocol at the communication time θ_{com} guarantees that the agents recover the diagnosis result of a centralized agent by consistent pairs of local diagnostics

The assumption made above is that the communication exchange between two neighboring agents is simultaneous (synchronous) and takes place in different communication rounds and that the local calculations at each site do not include new observations (events observed happening after θ_{com}).

In the following we present a distributed algorithm that comprises:

i) a procedure for performing the local preliminary calculations in absence of of any external information (Section 5.7.2)

ii) a procedure for information exchange (Section 5.7.3)

iii) a procedure for updating a local calculation to incorporate the received information (Section 5.7.4)

Then in Section 5.7.5 we prove the main result of this chapter that is the distributed algorithm we propose terminates after finitely many communication rounds and by the completion of the information exchange (communication protocol) the local projection of the centralized diagnosis result is recovered by each agent.

5.7.2 Procedure for performing the preliminary calculations

In this section we present the preliminary calculations that are performed by a local agent Ag_i before communicating with its neighbors.

Consider in the following that Ag_i receives the first local observation $\mathcal{O}^{\theta}_{1_i} = \langle obs_{1_i}, \theta_{obs_{1_i}} \rangle$. Since the communication with its neighbors is not possible prior to θ_{com}, Ag_i should derive an analysis using the local model the local observation and the known initial marking of internal places but not knowing the marking of the border places of its component.

Similarly as presented in Section 5.6 the local agent Ag_i computes the set of minimal time-interval configurations and derives the set of minimal assumptions about the tokens that entered the local site TPN model as follows:

1. first Ag_i derives the set of untimed minimal configurations that explain the first local observation $\overleftarrow{\mathcal{C}}_i(\mathcal{O}^{\theta}_{1_i})$ where for minimal untimed configuration $\overleftarrow{C}_{\ell_i}(\mathcal{O}^{\theta}_{1_i}) \in \overleftarrow{\mathcal{C}}_i(\mathcal{O}^{\theta}_{1_i})$ we have $\overleftarrow{B}_{\overleftarrow{C}_{\ell_i}}(IN_i)$ representing the set of

minimal assumptions about the tokens that are required to have entered the local site i via the input border places $\mathcal{P}_{IN_i}$

2. for $\overleftarrow{C}_{\ell_i} \in \overleftarrow{\mathcal{C}}_i(\mathcal{O}^\theta_{1_i})$ consider for each token that entered $b_{IN_i} \in \overleftarrow{B}_{\overleftarrow{C}_{\ell_i}}(IN_i)$ the over-approximant time-interval of the date of birth of the token in b_{IN_i} as $\overline{dob}(b_{IN_i}) = [0, \theta_{obs_{1_i}})$ and derive the set of minimal time-interval configurations $\overleftarrow{\mathcal{TC}}_{\ell_i}(\mathcal{O}^\theta_{1_i})$ by running Algorithm 19 for $\overleftarrow{C}_{\ell_i} \in \overleftarrow{\mathcal{C}}_i(\mathcal{O}^\theta_{1_i})$

3. for $\overleftarrow{C}_{\ell_i}(\mathbf{I}_{\nu_{\ell_i}}) \in \overleftarrow{\mathcal{TC}}_{\ell_i}(\mathcal{O}^\theta_{1_i})$ we have for each border place $b_{IN_{\ell_i}}$ the time-interval $\underline{dob}(b_{IN_i}) \subseteq \overline{dob}(b_{IN_i}))$ that represents the time-requirement for arriving of the token in b_{IN_i}

4. then for each $\overleftarrow{C}_{\ell_i}(\mathbf{I}_{\nu_{\ell_i}}) \in \overleftarrow{\mathcal{TC}}_{\ell_i}(\mathcal{O}^\theta_{1_i})$, Ag_i computes forward the set of unobservable time-interval configurations extension that:
   (a) do not violate the 1-safe property of the overall model
   (b) do not violate the sensor failure-free assumption

5. denote $\underline{\mathcal{TC}}_i(\mathcal{O}^\theta_{1_i})$ the set of time-interval configurations obtained in this way where we use the "underline" to express that a time-interval configuration $\underline{C}_{\ell_i}(\mathbf{I}_{\ell_i}) \in \underline{\mathcal{TC}}_i(\mathcal{O}^\theta_{1_i})$ is based on the minimal assumption that tokens corresponding with $b_{IN_{\ell_i}} \in \overleftarrow{B}_{\overleftarrow{C}_{\ell_i}}(IN_i)$ have entered the local site i in the time-interval given by $\underline{dob}(b_{IN_{\ell_i}})$.

We have that $\underline{C}_{\upsilon_i}(\mathbf{I}_{\upsilon_i}) \in \underline{\mathcal{TC}}_i(\mathcal{O}^\theta_{1_i})$ iff:

1. $\exists \overleftarrow{C}_\ell(\mathbf{I}_\ell) \in \overleftarrow{\mathcal{TC}}_i(\mathcal{O}^\theta_1)$ s.t. $\overleftarrow{C}_{\ell_i}(\mathbf{I}_{\ell_i}) \sqsubseteq \underline{C}_{\upsilon_i}(\mathbf{I}_{\upsilon_i})$
2. $\forall e \in E_{\underline{C}_{\upsilon_i}} \setminus \overleftarrow{E}_{\overleftarrow{C}_{\ell_i}} \Rightarrow \phi(e) \in \mathcal{T}_{uo}$
3. $\forall b, b' \in B_{\underline{C}_{\upsilon_i}}$, $\phi(b) = \phi(b') \Rightarrow (b \preceq b' \vee b' \preceq b)$
4. $\forall e \in ENABLED(\underline{C}_{\upsilon_i}) \Rightarrow L_{\upsilon_i}(e) > \theta_{obs_1}$

Given the already received observation $\mathcal{O}^\theta_{n-1_i}$ and the last observation $\langle obs_{n_i}, \theta_{obs_{n_i}} \rangle$ we have that:

1. Ag_i derives the set of untimed minimal configuration that explain the n_i^{th} local observation $\overleftarrow{\mathcal{C}}_i(\mathcal{O}^\theta_{n_i})$ where for minimal untimed configuration $\overleftarrow{C}_{\ell_i}(\mathbf{I}_{\ell_i}) \in \overleftarrow{\mathcal{C}}_i(\mathcal{O}^\theta_{n_i})$ we have $\overleftarrow{B}_{\overleftarrow{C}_{\ell_i}}(IN_i)$ representing the set of minimal assumptions about the tokens that are required that have entered the local site i via the input border places $\mathcal{P}_{IN_i}$.

2. then for $\overleftarrow{C}_{\ell_i}(\mathbf{I}_{\ell_i}) \in \overleftarrow{\mathcal{C}}_{\ell_i}(\mathcal{O}^{\theta}_{n_i})$ consider $\overline{dob}(b_{IN_{\ell_i}}) = [0, \theta_{obs_{n_i}})$ an over-approximant time-interval of the date of birth of the token in $b_{IN_{\ell_i}}$ and then derive the set of minimal time-interval configurations $\overleftarrow{\mathcal{TC}}_{\ell_i}(\mathcal{O}^{\theta}_{n_i})$ by running Algorithm 19

3. $\overleftarrow{\mathcal{TC}}_i(\mathcal{O}^{\theta}_{n_i}) = \bigcup_{\ell_i \in \mathcal{V}_i^{obs_{n_i}}} \overleftarrow{\mathcal{TC}}_{\ell_i}(\mathcal{O}^{\theta}_{n_i})$

4. for each $\overleftarrow{C}_{v_i}(\mathbf{I}_{v_i}) \in \overleftarrow{\mathcal{TC}}_i(\mathcal{O}^{\theta}_{n_i})$ compute forward the set of unobservable time-interval configurations extension that: do not violate the sensor failure-free assumption

5. denote $\underline{\mathcal{TC}}_i(\mathcal{O}^{\theta}_{n_i})$ the set of time-interval configurations obtained in this way

For the overall PN model the 1-safe property can be alternatively expressed as follows. Consider a PN $\langle \mathcal{N}, M_0 \rangle$ that is untime 1-safe. Given the unfolding of $\langle \mathcal{N}, M_0 \rangle$, $\mathcal{U}_{\mathcal{N}}(M_0)$ and $\mathcal{C}$ the set of all the configurations in $\mathcal{U}_{\mathcal{N}}(M_0)$ we have that:

$$\forall C \in \mathcal{C},\ \forall b, b' \in E_C,\ \phi(b) = \phi(b') \ \Rightarrow \ (b \preceq b') \vee (b' \preceq b)$$

In a distributed setting this property cannot be checked since a local agent has no knowledge of the plant model out of its local site. However we present bellow a set of constraints that a local agent can derive based on its knowledge that the overall PN model has the 1-safe property.

Consider $\underline{C}_{v_i}(\mathbf{I}_{v_i})$ and consider moreover that $\exists b_i, b_i' \in B_{\underline{C}_{v_i}}$ such that $b_i \neq b_i'$ and $\phi(b_i) = \phi(b_i')$ (otherwise it is not required to check the violation of 1-safeness). Then consider that b_i is concurrent with b_i' in $\underline{C}_{v_i}(\mathbf{I}_{v_i})$ (denoted as $b_i \parallel_i b_i'$). Since $b_i \parallel_i b_i'$ in $\underline{C}_{v_i}(\mathbf{I}_{v_i})$ we have that in any global configuration there will be either $b_i \prec b_i'$ or $b_i' \prec b_i$. The idea is to impose "artificially" the causality that $(b_i \prec_i^{art} b_i') \vee (b_i' \prec_i^{art} b_i)$ (the upper index 'art' emphasizes that $b_i' \prec_i^{art} b_i$ is an artificial relation).

In the distributed setting for TPN we have assumed that in each component every oriented path that starts in an input border place and ends in an output border places contains at least one observed event. This implies that either $b_i \prec b_i'$ or $b_i' \prec b_i$ hold true but not both relations can be true in global configurations that includes $\underline{C}_{v_i}(\mathbf{I}_{v_i})$ as a sub-net.

Consider the case that the order relation that is possible is $b_i \prec_i^{art} b_i'$. Denote by $[b_i]^{\uparrow}_{OUT_i}$ the set of successor condition-nodes of the node b that are output border conditions in $\underline{C}_{v_i}(\mathbf{I}_{v_i})$:

$$[b_i]^{\uparrow}_{OUT_{v_i}} = \left\{ b_{OUT_{v_i}} \in B_{\underline{C}_{v_i}} \mid b_i \preceq b_{OUT_{v_i}} \right\}$$

Denote by $[b'_i]^{\downarrow}(IN_i)$ the set of predecessor condition-nodes of the the node b'_i that correspond with input border conditions in $\underline{C}_{\upsilon_i}(\mathbf{I}_{\upsilon_i})$:

$$[b'_i]^{\downarrow}_{IN_{\upsilon_i}} = \left\{ b'_{IN_{\upsilon_i}} \in B_{\underline{C}_{\upsilon_i}} \mid b'_{IN_{\upsilon_i}} \preceq b'_i \right\}$$

We have that:

$$b_i \prec b'_i \Leftrightarrow \exists (b_{OUT_{\upsilon_i}}, b'_{IN_{\upsilon_i}}) \in [b_i]^{\uparrow}_{OUT_i} \times [b'_i]^{\downarrow}_{IN_i} \text{ s.t. } b_{OUT_{\upsilon_i}} \prec b'_{IN_{\upsilon_i}}$$

Denote in the following by $\underline{eout}_i(b_i)$ and $\underline{lout}_i(b_i)$ the smallest earliest time, respectively the smallest latest time when a token that corresponds with a successor output-border condition of b_i can leave component i:

$$\underline{eout}_i(b_i) = \mathtt{min}_{b_{OUT_{\upsilon_i}} \in [b_i]^{\uparrow}_{OUT_i}} (edob(b_{OUT_{\upsilon_i}}))$$

$$\underline{lout}_i(b_i) = \mathtt{min}_{b_{OUT_{\upsilon_i}} \in [b_i]^{\uparrow}_{OUT_i}} (ldob(b_{OUT_{\upsilon_i}}))$$

We have for $\underline{C}_{\upsilon_i}(\mathbf{I}_{\upsilon_i})$ that $\exists b'_{IN_{\upsilon_i}} \in [b'_i]^{\downarrow}(IN_i)$ s.t.:

$$edob(b'_{IN_{\upsilon_i}}) \geq \underline{eout}_i(b_i) \tag{5.38}$$

and

$$ldob(b'_{IN_{\upsilon_i}}) \geq \underline{lout}_i(b_i) \tag{5.39}$$

Hence for all the condition-nodes in $\underline{C}_{\upsilon_i}(\mathbf{I}_{\upsilon_i})$ that are concurrent and that have the same image via ϕ we can derive constrains of the form (5.38) and/or (5.39) that can be imposed for refining the local preliminary calculation. Notice that these constrains may even force a local agent to discard unfeasible local configurations before communicating with its neighbour.

We have that $\underline{C}_{\upsilon_i}(\mathbf{I}_{\upsilon_i}) \in \underline{\mathcal{TC}}_i(\mathcal{O}^{\theta}_{n_i})$ iff:

(1) $\exists \overleftarrow{C}_{\lambda_i}(\mathbf{I}_{\lambda_i}) \in \overleftarrow{\mathcal{TC}}_i(\mathcal{O}^{\theta}_{n_i})$ s.t. $\overleftarrow{C}_{\lambda_i}(\mathbf{I}_{\lambda_i}) \sqsubset \underline{C}_{\upsilon_i}$

(2) $\forall e \in E_{\underline{C}_{\upsilon_i}} \setminus E_{\overleftarrow{C}_{\lambda_i}} \Rightarrow \phi(e) \in \mathcal{T}_{uo}$

(3) $\forall e \in ENABLED(\underline{C}_{\upsilon_i}(\mathbf{I}_{\upsilon_i}))$, $\phi(e) \in \mathcal{T}_{uo} \Rightarrow L_{\upsilon_i}(e) > \theta_{obs_{n_i}}$

In words $\underline{C}_{\upsilon_i}(\mathbf{I}_{\upsilon_i})$ is a preliminary time-interval configuration of component i if it is a an unobservable extension of a minimal time-interval configuration (1), (2) and all the enabled events can be executed after the time when the observation is recorded (3).

5.7.3 Procedure for information exchange

Consider in the following that at the time θ_{com} the communication between the agents is allowed. θ_{com} may be initiated by the supervisor (possibly at the request of local agents) or may be periodic.

Let $\underline{\mathcal{TC}}_i(\mathcal{O}^{\theta}_{n_i})$ be the set of local minimal time-interval configurations derived by a local agent Ag_i after receiving the last observation up to the time θ_{com}. Denote by $\mathcal{TC}_i(\mathcal{O}^{\theta}_{\theta_{com}})$ the set of all local time-interval configurations calculated by agent Ag_i that unobservably extend the configurations of $\underline{\mathcal{TC}}_i(\mathcal{O}^{\theta}_{n_i})$ until all the enabled events have their upper limit greater that θ_{com}.

$C_{\ell_i}(\mathbf{I}_\ell) \in \mathcal{TC}_i(\mathcal{O}^{\theta}_{\theta_{com}})$ iff:

1. $\exists \underline{C}_{v_i}(\mathbf{I}_{v_i}) \in \underline{\mathcal{TC}}_i(\mathcal{O}^{\theta}_{n_i})$ and $\underline{C}_{v_i}(\mathbf{I}_{v_i}) \sqsubseteq C_{\ell_i}(\mathbf{I}_{\ell_i})$
2. $\forall e \in E_{C_{\ell_i}} \setminus E_{\underline{C}_{v_i}} \Rightarrow \phi(e) \in \mathcal{T}_{uo}$
3. $\forall e \in ENABLED(C_{\ell_i})$,
   (a) if $\phi(e) \in \mathcal{T}_{uo}$ then $U_{\ell_i}(e) \geq \theta_{com}$
   (b) if $\phi(e) \in \mathcal{T}_o$ then $L_{\ell_i}(e) \geq \theta_{com}$

For a local preliminary time-interval configuration $C_{\ell_i}(\mathbf{I}_{\ell_i}) \in \mathcal{TC}_i(\mathcal{O}^{\theta}_{\theta_{com}})$ denote by $B^{\theta}_{C_{\ell_i}}(IN_i)$ the set of minimal temporal assumptions about the tokens that have entered to component i:

$$B^{\theta}_{C_{\ell_i}}(IN_i) = \Big\{(b_{IN_i}, dob(b_{IN_i})) \mid b_{IN_i} \in B_{C_{\ell_i}}(IN_i)\Big\}$$

Denote by $B^{\theta}_{C_{\ell_i}}(OUT_i)$ the set of tokens that could have exited component i and entered to component j where for each token that leaves i we have the departure time-interval given by the date of birth interval $dob(b_{OUT_i})$ of the token in the output border place b_{OUT_i}:

$$B^{\theta}_{C_{\ell_i}}(OUT_i) = \Big\{(b_{OUT_i}, dob(b_{OUT_i})) \mid b_{OUT_i} \in \underline{B}_{C_{\ell_i}}(OUT_i)\Big\}$$

Then at time θ_{com} Ag_i sends to Ag_j for each local preliminary time-interval configuration $C_{\ell_i}(\mathbf{I}_{\ell_i}) \in \mathcal{TC}_i(\mathcal{O}^{\theta}_{\theta_{com}})$ the message:

$$MSG_{i\to j}(C_{\ell_i}(\mathbf{I}_{\ell_i})) = (B^{\theta}_{C_{\ell_i}}(IN_i), B^{\theta}_{C_{\ell_i}}(OUT_i))$$

In other words the message that is sent by Ag_i to Ag_j at the time θ_{com} contains these messages for all $C_{\ell_i}(\mathbf{I}_{\ell_i}) \in \mathcal{TC}_i(\mathcal{O}^{\theta}_{\theta_{com}})$:

$$\mathcal{MSG}_{i\to j}(\theta_{com}) = \{MSG_{i\to j}(\underline{C}_{\ell_i}(\mathbf{I}_{\ell_i})) \mid C_{\ell_i}(\mathbf{I}_{\ell_i}) \in \mathcal{TC}_i(\mathcal{O}^{\theta}_{\theta_{com}})\}$$

5.7.4 Procedure for updating the local calculations

In this section we present how the local agent Ag_i includes the received information for updating the local calculations and for checking the consistency of its results with the results of the agent Ag_j.

Agents i and j check the consistency of their preliminary results for each pair of preliminary local time-interval configurations:

$$(C_{\ell_i}(\mathbf{I}_{\ell_i}), C_{\ell j}(\mathbf{I}_{\ell j})) \in \mathcal{TC}_i(\mathcal{O}^{\theta}_{\theta_{com}}) \times \mathcal{TC}_j(\mathcal{O}^{\theta}_{\theta_{com}})$$

Both agents must check whether a global time-interval configuration exists that explains both local configurations or not.

In a distributed way the checking is done recursively by exchanging information about the temporal border-conditions until either an agreement is achieved and thus a global time-interval configuration is recovered or it is found that the two local configurations are not consistent.

Fix attention for the time being to a pair $(C_{\ell_i}(\mathbf{I}_{\ell_i}), C_{\ell j}(\mathbf{I}_{\ell j}))$ of local configurations in component i, resp. component j, and consider that Ag_i has received the message:

$$MSG^1_{j\to i}(C_{\ell_j}(\mathbf{I}_{\ell_j})) = (B^{\theta}_{\ell_j}(IN_j), B^{\theta}_{\ell_j}(OUT_j))$$

Agent Ag_i combines this message with its local information on $C_{\ell_i}(\mathbf{I}_{\ell_i})$ analyzing the pair $(C_{\ell_i}(\mathbf{I}_{\ell_i}), C_{\ell j}(\mathbf{I}_{\ell j}))$. This pair is possibly consistent if:

1. $\forall b_{IN_i} \in B_{C_{\ell_i}}(IN_i)$, $\exists b_{OUT_j} \in B_{C_{\ell_j}}(OUT_j)$ s.t. $\phi(b_{IN_i}) = \phi(b_{OUT_j})$ and $dob(b_{IN_i}) \cap dob(b_{OUT_j}) \neq \emptyset$.
2. $\forall b_{IN_j} \in B_{C_{\ell_j}}(IN_j)$, $\exists b_{OUT_i} \in B_{C_{\ell_i}}(OUT_i)$ s.t. $\phi(b_{IN_j}) = \phi(b_{OUT_i})$ and $dob(b_{IN_j}) \cap dob(b_{OUT_i}) \neq \emptyset$.

For a pair of local time-interval configurations $(C_{\ell_i}(\mathbf{I}_{\ell_i}), C_{\ell j}(\mathbf{I}_{\ell j}))$ the interpretation function of the common border conditions $\psi_{\ell_i\ell_j}$ is uniquely defined since the net is 1-safe and each oriented path that starts in an input place and ends in an output place of the component i includes at least one observable event.

The pair of preliminary time-interval configurations $(C_{\ell_i}(\mathbf{I}_{\ell_i}), C_{\ell j}(\mathbf{I}_{\ell j}))$ is moved to the list of consistent pairs if:

1. $\mid B_{C_{\ell_i}}(IN_i) \mid = \mid B_{C_{\ell_j}}(OUT_j) \mid$
2. $\mid B_{C_{\ell_i}}(OUT_i) \mid = \mid B_{C_{\ell_j}}(IN_j) \mid$
3. $\forall b_{IN_i} \in B_{C_{\ell_i}}(IN_i)$, $\exists b_{OUT_j} \in B_{C_{\ell_j}}(OUT_j)$ s.t. $\phi(b_{IN_i}) = \phi(b_{OUT_j})$ and $dob(b_{IN_i}) = dob(b_{OUT_j})$.

4. $\forall b_{IN_j} \in B_{C_{\ell_j}}(IN_j)$, $\exists b_{OUT_i} \in B_{C_{\ell_i}}(OUT_i)$ s.t. $\phi(b_{IN_j}) = \phi(b_{OUT_i})$ and $dob(b_{IN_j}) = dob(b_{OUT_i})$.

Notice that for a possibly consistent pair $(C_{\ell_i}(\mathbf{I}_{\ell_i}), C_{\ell j}(\mathbf{I}_{\ell j}))$ if either:

$$| B_{C_{\ell_i}}(IN_i) | < | B_{C_{\ell_j}}(OUT_j) | \text{ or } | B_{C_{\ell_i}}(OUT_i) | > | B_{C_{\ell_j}}(IN_j) |$$

then it is not certain that $(C_{\ell_i}(\mathbf{I}_{\ell_i}), C_{\ell j}(\mathbf{I}_{\ell j}))$ can be expanded to a consistent pair of local time-interval configurations, because the extra-input border conditions may be forced by upper time bounds to fire an observable transition that was not locally observed. Since the observation is correct it means that such a pair must be discarded from the list of possible consistent pairs in Ag_i.

Consider the case of a possibly consistent pair $(C_{\ell_i}(\mathbf{I}_{\ell_i}), C_{\ell j}(\mathbf{I}_{\ell j}))$.

Denote by $\mathcal{TC}^i_{\ell_i\ell_j} = \left\{C_{\ell_i}(\mathbf{I}_{\nu^i_{\ell_i\ell_j}}) \mid \nu^i_{\ell_i\ell_j} \in \mathcal{V}^i_{\ell_i\ell_j}\right\}$ the set of local time-interval configurations obtained from $\underline{C}_{\ell_i}(\mathbf{I}_{\ell_i})$ imposing all the constraints $\mathcal{K}^i_{\ell_i\ell_j}$ regarding the common border places:

$$\mathcal{K}^{1_i}_{\ell_i\ell_j} = \mathcal{K}^i_{\ell_i\ell_j}(IN_i) \wedge \mathcal{K}^i_{\ell_i\ell_j}(OUT_i)$$

where:

$$\mathcal{K}^{1_i}_{\ell_i\ell_j}(IN_i) = \Big\{\kappa_{b_{IN_i}} := \Big\{dob'_{\ell_i\ell_j}(b_{IN_i}) = dob_{\ell_i}(b_{IN_i}) \cap dob_{\ell_j}(b_{OUT_j})\Big\} \\ \text{for } b_{IN_i} \in B_{C_{\ell_i}}(IN_i)\Big\}$$

and:

$$\mathcal{K}^{1_i}_{\ell_i\ell_j}(OUT_i) = \Big\{\kappa_{b_{OUT_i}} := \Big\{dob'_{\ell_i\ell_j}(b_{OUT_i}) = dob_{\ell_i}(b_{OUT_i}) \cap dob_{\ell_j}(b_{IN_j})\Big\} \\ \text{for } b_{OUT_i} \in B_{C_{\ell_i}}(OUT_i)\Big\}$$

Denote by $\Delta B_{\ell_i\ell_j}(IN_i)$ and $\Delta B_{\ell_i\ell_j}(IN_j)$ the set of extra border-conditions that are provided to component i and j.

Then denote by $\mathcal{TC}^{1_i}_{\ell_i\ell_j}$ the set of all unobservable extensions of the time-interval configuration $C_{\ell_i}(\mathbf{I}_{\ell_i\ell_j}) \in \mathcal{TC}^i_{\ell_i\ell_j}$. The extensions are derived using the extra border-conditions $\Delta B_{\ell_i\ell_j}(IN_i)$ appending unobservable events until the time θ_{com} imposing also the constrains that no other observable events are executed before θ_{com}.

We have that $C_{\nu^i_{\ell_i\ell_j}}(\mathbf{I}_{\nu_{\ell_i\ell_j}}) \in \mathcal{TC}^i_{\ell_i\ell_j}$ if:

1. $C_{\ell_i} \sqsubseteq C_{\nu^i_{\ell_i\ell_j}}$

2. $\forall e \in E_{C_{\nu^i_{\ell_i\ell_j}}} \setminus E_{C_{\ell_i}} \Rightarrow \phi(e) \in \mathcal{T}_{uo}$

3. $\forall e \in ENABLED(E_{C_{\nu_{\ell_i \ell_j}}})$
   (a) if $\phi(e) \in \mathcal{T}_{uo}$ then $U_{\nu^i_{\ell_i \ell_j}}(e) \geq \theta_{com}$
   (b) if $\phi(e) \in \mathcal{T}_{o}$ then $L_{\nu^i_{\ell_i \ell_j}}(e) \geq \theta_{com}$
4. $\forall e \in E_C$, $L_{\nu^i_{\ell_i \ell_j}}(e) \leq \theta_{com}$

Let $\mathcal{TC}^{1_i}_{\ell_i \ell_j}$ and $\mathcal{TC}^{1_j}_{\ell_i \ell_j}$ be the set of all the unobservable extensions in component i and component j respectively derived after the first communication round:

$$\mathcal{TC}^{1_i}_{\ell_i \ell_j} = \left\{ \mathcal{TC}^{i}_{\nu^i_{\ell_i \ell_j}} \mid \nu^i_{\ell_i \ell_j} \in \mathcal{V}^i_{\ell_i \ell_j} \right\}$$

$$\mathcal{TC}^{1_j}_{\ell_i \ell_j} = \left\{ \mathcal{TC}^{j}_{\nu^j_{\ell_j \ell_i}} \mid \nu^j_{\ell_j \ell_i} \in \mathcal{V}^j_{\ell_i \ell_j} \right\}$$

If no agreement has been reached yet (the set of possibly consistent pairs is non-empty) in a new communication round Ag_i and Ag_j exchange the time intervals for common border conditions for the remaining pairs of possibly consistent time-interval configurations $\mathcal{TC}^{1_i}_{\ell_i \ell_j}$ and $\mathcal{TC}^{1_j}_{\ell_i \ell_j}$.

Consider in the following that:

$$\mathcal{TC}^{1_i}_{\ell_i \ell_j} = \left\{ C_{\gamma_{1_i}}(\mathbf{I}_{\gamma_{1_i}}) \mid \gamma_{1_i} \in \mathcal{V}^{1_i}_{\ell_i \ell_j} \right\}$$

For a local configuration $C_{\gamma_{1_i}}(\mathbf{I}_{\gamma_{1_i}}) \in \mathcal{TC}^{1_i}_{\ell_i \ell_j}$ obtained by updating a preliminary configuration after the first communication round, Ag_i sends to Ag_j the message:

$$MSG^2_{i \to j}(C_{\gamma_{1_i}}(\mathbf{I}_{\gamma_{1_i}})) = (B^{\theta}_{\gamma_{1_i}}(IN_i), B^{\theta}_{\gamma_{1_i}}(OUT_i))$$

The message that is sent by Ag_i to Ag_j regarding the time-configurations that are derived from the pair of preliminary time-interval configurations $(\underline{C}_{\ell_i}(\mathbf{I}_{\ell_i}), \underline{C}_{\ell j}(\mathbf{I}_{\ell j}))$ after the first update is:

$$MSG^2_{i \to j}(\ell_i \ell_j) = \left\{ MSG^2_{i \to j}(C_{\gamma_{1_i}}(\mathbf{I}_{\gamma_{1_i}})) \mid \gamma_{1_i} \in \mathcal{V}^{1_i}_{\ell_i \ell_j} \right\}$$

Then Ag_i receives the message sent by Ag_j in the second communication round. Let $(C_{\gamma_{1_i}}(\mathbf{I}_{\gamma_{1_i}}), C_{\gamma_{1_j}}(\mathbf{I}_{\gamma_j})) \in \mathcal{TC}^{1_i}_{\ell_i \ell_j} \times \mathcal{TC}^{1_j}_{\ell_i \ell_j}$ be a pair of local configurations obtained after the first update of the preliminary results.

The set of constrains that must be imposed by Ag_i to $C_{\gamma_{1_i}}(\mathbf{I}_{\gamma_{1_i}})$ is

$$\mathcal{K}^{2_i}_{\gamma_{1_i}\gamma_{1_j}} = \mathcal{K}^{2_i}_{\gamma_{1_i}}(IN_i) \wedge \mathcal{K}^{2_i}_{\gamma_{1_i}\gamma_{1_j}}(OUT_i)$$

where:

$$\mathcal{K}^{2_i}_{\gamma_{1_i}\gamma_{1_j}}(IN_i) = \Big\{\kappa_{b_{IN_i}} := \Big\{dob'_{\gamma_{1_i}}(b_{IN_i}) = dob_{\gamma_{1_i}}(b_{IN_i}) \cap dob_{\gamma_{1_j}}(b_{OUT_j})\Big\} \\ \text{for } \; b_{IN_i} \in B_{C_{\gamma_{1_i}}}(IN_i)\Big\}$$

and:

$$\mathcal{K}^{2_i}_{\gamma_{1_i}\gamma_{1_j}}(OUT_i) = \Big\{\kappa_{b_{OUT_i}} = \Big\{dob'_{\gamma_i}(b_{OUT_i}) = dob_{\gamma_{1_i}}(b_{OUT_i}) \cap dob_{\gamma_{1_j}}(b_{IN_j})\Big\} \\ \text{for } b_{OUT_i} \in B_{C_{\gamma_{1_i}}}(OUT_i)\Big\}$$

Denote by $\mathcal{TC}^{2_i}_{\ell_i\ell_j}$ the set of possibly consistent time-interval configurations derived by Ag_i after the second communication round:

$$\mathcal{TC}^{2_i}_{\ell_i\ell_j} = \Big\{\mathcal{TC}^{2_i}_{\gamma_{1_i}\gamma_{1_j}} \mid \gamma_{1_i} \in \mathcal{V}^{1_i}_{\ell_i\ell_j} \;\; ; \;\; \gamma_{1_j} \in \mathcal{V}^{1_j}_{\ell_i\ell_j}\Big\}$$

where:

$$\mathcal{TC}^{2_i}_{\gamma_{1_i}\gamma_{1_j}} = \Big\{C_{\gamma_{1_i}\gamma_{1_j}}(\mathbf{I}_{\gamma_{2_i}}) \mid \gamma_{2_i} \in \mathcal{V}^{2_i}_{\gamma_{1_i}\gamma_{1_j}}\Big\}$$

The message that is sent by Ag_i to Ag_j at the third communication regarding the time-interval configurations $C_{\gamma_{1_i}\gamma_{1_j}}(\mathbf{I}_{\gamma_{2_i}}) \in \mathcal{TC}^{2_i}_{\ell_i\ell_j}$ comprises only information about the output border conditions since the time-intervals of the input border conditions remained unchanged:

$$MSG^3_{i\to j}(C_{\gamma_{1_i}\gamma_{1_j}}(\mathbf{I}_{\gamma_{2_i}})) = \Big\{B^{\theta}_{C_{\gamma_{1_i}\gamma_{1_j}}}(OUT_i)\Big\}$$

The message that is sent by Ag_i to Ag_j at the third communication round regarding the time-interval configurations that are derived from the pair of preliminary time-interal configurations $(\underline{C}_{\ell_i}(\mathbf{I}_{\ell_i}), \underline{C}_{\ell j}(\mathbf{I}_{\ell j}))$ after the second update is:

$$MSG^3_{i\to j}(\ell_i\ell_j)) = \Big\{MSG^3_{i\to j}(C_{\gamma_{1_i}\gamma_{1_j}}(\mathbf{I}_{\gamma_{2_i}})) \mid \gamma_{2_i} \in \mathcal{V}^{2_i}_{\gamma_{1_i}\gamma_{1_j}} \wedge \gamma_{1_i} \in \mathcal{V}^{1_i}_{\ell_i\ell_j} \\ \wedge \; \gamma_{1_j} \in \mathcal{V}^{1_j}_{\ell_i\ell_j}\Big\}$$

Let $(C_{\gamma_{1_i}\gamma_{1_j}}(\mathbf{I}_{\gamma_{2_i}}), C_{\gamma_{1_i}\gamma_{1_j}}(\mathbf{I}_{\gamma_{2_j}})) \in \mathcal{TC}^{2_i}_{\ell_i\ell_j} \times \mathcal{TC}^{2_j}_{\ell_i\ell_j}$ be a pair of local time-interval configurations obtained after the first update of the preliminary results.

The set of constraints that must be imposed by Ag_i to $C_{\gamma_1\gamma_2}((\mathbf{I}_{\gamma_{2_i}}))$ is

$$\mathcal{K}^{3_i}_{\gamma_{2_i}\gamma_{2_j}} = \mathcal{K}^{3_i}_{\gamma_{2_i}\gamma_{2_j}}(IN_i)$$

where:

$$\mathcal{K}^{3_i}_{\gamma_{2_i}\gamma_{2_j}}(IN_i) = \Big\{\kappa_{b_{IN_i}} := dob'_{\gamma_{2_i}}(b_{IN_i}) = dob_{\gamma_{2_j}}(b_{OUT_j}) \mid b_{IN_i} \in B_{C_{\gamma_{2_i}}}(IN_i)\Big\}$$

Denote by $C_{\gamma_1\gamma_2}(\mathbf{I}_{\gamma_{2_i}\gamma_{2_j}})$ the time-interval configuration that is obtained imposing $\mathcal{K}^{3_i}_{\gamma_{2_i}\gamma_{2_j}}$ to $C_{\gamma_1\gamma_2}(\mathbf{I}_{\gamma_{2_i}})$. $C_{\gamma_1\gamma_2}(\mathbf{I}_{\gamma_{2_i}\gamma_{2_j}})$ is unique since constraints imposed to the input border conditions do not require to split up a time-interval configuration.

The two agents stop the checking procedure after (maximum) three communication rounds since the time-intervals for all the border-conditions can not be further modified.

Summarizing the checking procedure comprises three stages, each stage including a communication round followed by a local update of the local results:

Procedure for checking consistency

- **stage 1**

  1.1 the two agents have the first communication round exchanging information about the border-conditions of the preliminary local time-interval configurations derived prior to communication.

  1.2 for each pair of local preliminary time-interval configurations $(C_{\ell_i}(\mathbf{I}_{\ell_i}), C_{\ell_j}(\mathbf{I}_{\ell_j}))$ the two agents check first if they are possibly consistent. Possibly consistent means first that the local configurations C_{ℓ_i}, C_{ℓ_j} are untimed consistent and notice that since the overall PN model is one safe there is a unique interpretation function for each pair of local configurations. Moreover for each common border condition there is a non-empty intersection of the time-intervals with which it is considered in the local time-interval configurations. If there are extra input border-conditions, the pairs that are possibly consistent are then extended up to the time θ_{com} by appending unobservable events and also refined in order to include the temporal constraints due to the common border conditions. Consider that for the pair of local preliminary configurations C_{ℓ_i}, C_{ℓ_j} each agent derives a set of local time-interval configurations $\mathcal{TC}^{1_i}_{\ell_i\ell_j}$ respectively $\mathcal{TC}^{1_j}_{\ell_i\ell_j}$.

- **stage 2**

  2.1 the two agents have the second communication round exchanging information about the border-conditions of the local time-interval configurations derived at the first update.

  2.2 then for each pair $(C_{\gamma_{1_i}}(\mathbf{I}_{\gamma_{1_i}}), C_{\gamma_{1_j}}(\mathbf{I}_{\gamma_{1_j}})) \in \mathcal{TC}^{1_i}_{\ell_i\ell_j} \times \mathcal{TC}^{1_j}_{\ell_i\ell_j}$ constraints regarding the input and output border conditions are imposed by Ag_i and Ag_j to $C_{\gamma_{1_i}}(\mathbf{I}_{\gamma_{1_i}})$ and $C_{\gamma_{1_j}}(\mathbf{I}_{\gamma_{1_j}})$ respectively. Consider that for $C_{\gamma_{1_i}}(\mathbf{I}_{\gamma_{1_i}})$ Ag_i derives a set of local time-interval configurations:

$$\mathcal{TC}^{2_i}_{\gamma_i\gamma_j} = \left\{ C_{\gamma_{1_i}}(\mathbf{I}_{\gamma_{2_i}}) \mid \gamma_{2_i} \in \mathcal{V}_{\gamma_{1_i}\gamma_{1_j}} \right\}$$

Notice that $\mathbf{I}_{\gamma_{1_i}}$ is split-up because of the constraints applied to the output border conditions $B_{C_{\gamma_{1_i}}}$ (i.e. constraints that are propagated "backwards").

- **stage 3**

  3.1 the two agents have the third communication round exchanging information. A local agent sends information only about its output border-conditions of the local time-interval configurations derive at the first update.

  3.2 then for each pair $(C_{\gamma_{1_i}}(\mathbf{I}_{\gamma_{2_i}}), C_{\gamma_{1_j}}(\mathbf{I}_{\gamma_{2_i}})) \in \mathcal{TC}^{2_i}_{\gamma_{1_i}\gamma_{1_j}} \times \mathcal{TC}^{2_j}_{\gamma_{1_i}\gamma_{1_j}}$ constrains regarding the input conditions are imposed by Ag_i and Ag_j to $C_{\gamma_{1_i}}(\mathbf{I}_{\gamma_{2_i}})$ and $C_{\gamma_{1_j}}(\mathbf{I}_{\gamma_{2_j}})$ respectively. The procedure stops since no further refinements of the local time configurations are possible.

Notice that the checking procedure is completed in the worst case after three communications rounds. It may be that for some pairs of local preliminary time-interval configurations the procedure finishes after one or two communication rounds.

The distributed algorithm for TPN models can be summarized as follows:

1. Ag_i monitors component i for $\mathcal{O}^\theta_{n_i} = \langle obs_{1_i}, \theta_{obs_{1_i}}\rangle, \ldots, \langle obs_{n_i}, \theta_{obs_{n_i}}\rangle$
2. at the time θ_{com}, Ag_i derives $\underline{\mathcal{TC}}_i(\mathcal{O}^\theta_{\theta_{com}})$
3. then *Procedure for checking consistency*
4. let $\mathcal{TC}^{gcon}_i(\mathcal{O}_{\theta_{com}})$ and respectively $\mathcal{TC}^{gcon}_j(\mathcal{O}_{\theta_{com}})$ be the set of local time configurations in component i and component j respectively, found consistent after running the distributed algorithm, i.e.:

   (a) $\forall C_{v_i}(\mathbf{I}_{v_i}) \in \mathcal{TC}^{con}_i(\mathcal{O}_{\theta_{com}}) \Rightarrow \exists C_{v_j}(\mathbf{I}_{v_j}) \in \mathcal{TC}^{con}_j(\mathcal{O}_{\theta_{com}})$ such that $(C_{v_i}(\mathbf{I}_{v_i}), C_{v_j}(\mathbf{I}_{v_j}))$ is consistent

   (b) $\forall C_{v_j}(\mathbf{I}_{v_j}) \in \mathcal{TC}^{con}_j(\mathcal{O}_{\theta_{com}}) \Rightarrow \exists C_{v_i}(\mathbf{I}_{v_i}) \in \mathcal{TC}^{con}_i(\mathcal{O}_{\theta_{com}})$ such that $(C_{v_i}(\mathbf{I}_{v_i}), C_{v_j}(\mathbf{I}_{v_j}))$ is consistent

5.7.5 The main result

Consider $\mathcal{TC}^{gcon}_i(\mathcal{O}_{\theta_{com}})$ and $\mathcal{TC}^{gcon}_j(\mathcal{O}_{\theta_{com}})$ derived as presented above and considered $\mathcal{TC}(\mathcal{O}_{\theta_{com}})$ the set of time-interval configurations derived at the time θ_{com} by a centralized agent for the overall plant model having the overall plant observation. Then we have the following results:

Proposition 31. *Given $\mathcal{TC}_i^{gcon}(\mathcal{O}_{\theta_{com}})$, $\mathcal{TC}_j^{gcon}(\mathcal{O}_{\theta_{com}})$, and $\mathcal{TC}(\mathcal{O}_{\theta_{com}})$ for a plant consisting of two components i and j we have that:*

i) $\forall (C_{\nu_i}(\mathbf{I}_{\nu_i}), C_{\nu_j}(\mathbf{I}_{\nu_j})) \in \mathcal{TC}_i^{gcon}(\mathcal{O}_{\theta_{com}}) \times \mathcal{TC}_j^{gcon}(\mathcal{O}_{\theta_{com}})$
if $(C_{\nu_i}(\mathbf{I}_{\nu_i}), C_{\nu_j}(\mathbf{I}_{\nu_j}))$ *is consistent then*
$\exists C_\nu(\mathbf{I}_\nu) \in \mathcal{TC}(\mathcal{O}_{\theta_{com}})$ *s.t.* $C_\nu(\mathbf{I}_\nu) = (C_{\nu_i}(\mathbf{I}_{\nu_i}), C_{\nu_j}(\mathbf{I}_{\nu_j}))$

ii) $\forall C_\nu(\mathbf{I}_\nu) \in \mathcal{TC}(\mathcal{O}_{\theta_{com}}) \Rightarrow \exists (C_{\nu_i}(\mathbf{I}_{\nu_i}), C_{\nu_j}(\mathbf{I}_{\nu_j})) \in \mathcal{TC}_i^{gcon}(\mathcal{O}_{\theta_{com}}) \times \mathcal{TC}_j^{gcon}(\mathcal{O}_{\theta_{com}})$ *s.t.* $C_\nu(\mathbf{I}_\nu) = (C_{\nu_i}(\mathbf{I}_{\nu_i}), C_{\nu_j}(\mathbf{I}_{\nu_j}))$

Proof. The proof of $i)$ is straightforward since $C_{\nu_i}(\mathbf{I}_{\nu_i})$ and $C_{\nu_j}(\mathbf{I}_{\nu_j})$ have the time independence property and they consider the same time-intervals for the common border conditions.

The proof of $ii)$ is as follows. For each input border condition of a untimed configuration $\overleftarrow{\mathcal{C}}_{\ell_i}(\mathcal{O}^\theta_{n_i})$ we have considered $\overline{dob}(b_{IN_i}) = [0, \theta_{obs_{n_i}})$ and the extra input border conditions were taken into account deriving unobservable extensions for the preliminary time-interval configurations. □

For component i denote by $\mathcal{E}_i^{gcon}(\mathcal{O}^\theta_{\theta_{com}})$ the set of local untimed traces that are derived as linearizations of the partial order relation of the events of time-interval configurations:

$$\mathcal{E}_i^{gcon}(\mathcal{O}^\theta_{\theta_{com}}) = \{\sigma_i \in \langle E_{C_{\upsilon_i}}(\mathbf{I}_{\upsilon_i})\rangle \mid C_{\upsilon_i}(\mathbf{I}_{\upsilon_i}) \in \mathcal{TC}_i^{con}(\mathcal{O}^\theta_{\theta_{com}})\}$$

Denote $\mathcal{L}_i^{gcon}(\mathcal{O}^\theta_{\theta_{com}}) = \{\tau_i = \phi(\sigma_i) \mid \sigma_i \in \mathcal{E}_i(\mathcal{O}^\theta_{\theta_{com}})\}$ and then denote by $\mathcal{LDR}_i(\mathcal{O}^\theta_{\theta_{com}})$ the local diagnosis result derived by Ag_i based on $\mathcal{L}_i^{gcon}(\mathcal{O}^\theta_{\theta_{com}})$. We have:

Theorem 11. *Consider the distributed plant description as in Section 5.7.1 and an arbitrary time θ_{com} when a communication is executed. The algorithm of Section 5.7.2, Section 5.7.3, and Section 5.7.4 guarantees that the local diagnosis result that would have been derived by a centralized agent is recovered by the two agents running the distributed diagnosis algorithm for TPN:*

$$\mathcal{DR}(\mathcal{O}^\theta_{\theta_{com}}) = (\mathcal{LDR}_i(\mathcal{O}^\theta_{\theta_{com}}), \mathcal{LDR}_i(\mathcal{O}^\theta_{\theta_{com}}))$$

Proof. The proof is straightforward using Proposition 31 and the Theorem 7 for $\xi = \theta_{com}$. □

In the following we examine what the preliminary diagnosis includes. For a local preliminary time-interval configuration $C_{\ell_i}(\mathbf{I}_{\ell_i}) \in \mathcal{TC}_i(\mathcal{O}^\theta_{n_i})$ denote $\langle E_{\ell_i}\rangle$ the set of linearizations of the partial order of the events in $C_{\ell_i}(\mathbf{I}_{\ell_i})$.

Denote by $\mathcal{L}_i^{prel}(\mathcal{O}^\theta_{n_i})$ the set of all untimed traces obtained for $C_{\ell_i}(\mathbf{I}_{\ell_i}) \in \mathcal{TC}_i(\mathcal{O}^\theta_{n_i})$:

$$\mathcal{L}_i^{prel}(\mathcal{O}^\theta_{n_i}) = \{\tau_i = \phi(\sigma_i) \mid \sigma_i \in \langle E_{\ell_i}\rangle \wedge C_{\ell_i}(\mathbf{I}_{\ell_i}) \in \mathcal{TC}_i(\mathcal{O}^\theta_{n_i})\} \tag{5.40}$$

Let $\mathcal{LPD}_i(\mathcal{O}^{\theta}_{n_i})$ be the preliminary local diagnosis derived at the time θ_{com} based on $\mathcal{L}_i^{prel}(\mathcal{O}^{\theta}_{n_i})$ and let $\mathcal{D}_i(\mathcal{O}^{\theta}_{n})$ be the diagnosis of component i derived by a centralized agent considering the global plant observation $\mathcal{O}^{\theta}_{n}$. We say that:

1. the plant observation is uncertain if $\forall t \in \mathcal{T}_o$, $\exists t' \in \mathcal{T}_{uo}$ s.t. $^{\bullet}t' = {}^{\bullet}t$ and $t'^{\bullet} = t^{\bullet}$
2. the interactions between components are uncertain if $\forall p \in \mathcal{P}_{IN}$, $\exists t \in \mathcal{T}_{uo}$ s.t. $^{\bullet}t = p$ and $t^{\bullet} = \emptyset$

The local preliminary diagnosis is an over-diagnosis of component i w.r.t. the faults that would have been detected by the centralized diagnosis if for any observation generated by the plant $\mathcal{O}^{\theta}_{n}$ we have that:

> if the centralized diagnoser detects that a fault of kind $\mathcal{T}_{\mathtt{F_i}}$ happened for sure ($\mathcal{DR}(\mathcal{O}^{\theta}_{n}) = \mathtt{F}_{\mathtt{F_i}}$)
>
> then the local preliminary diagnosis result of site i is either sure that a fault of kind $\mathtt{F_i}$ happened or is uncertain:
>
> $$\left\{\mathcal{DR}_i(\mathcal{O}^{\theta}_{n}) = \mathtt{F}_{\mathtt{F_i}}\right\} \Rightarrow \left\{\mathcal{LPDR}_i(\mathcal{O}^{\theta}_{n_i}) = \mathtt{F}_{\mathtt{F_i}}\right\} \vee \left\{\mathcal{LPDR}_i(\mathcal{O}^{\theta}_{n_i}) = \mathtt{UF}_{\mathtt{F_i}}\right\}$$

Theorem 12. *If either the plant observation or the interactions between the components are uncertain then for any observation generated by the plant the local preliminary diagnosis of component i, is an over-diagnosis of component i w.r.t. the faults that would have been detected by the centralized agent that for sure happened in component i.*

Proof. The proof is straightforward. □

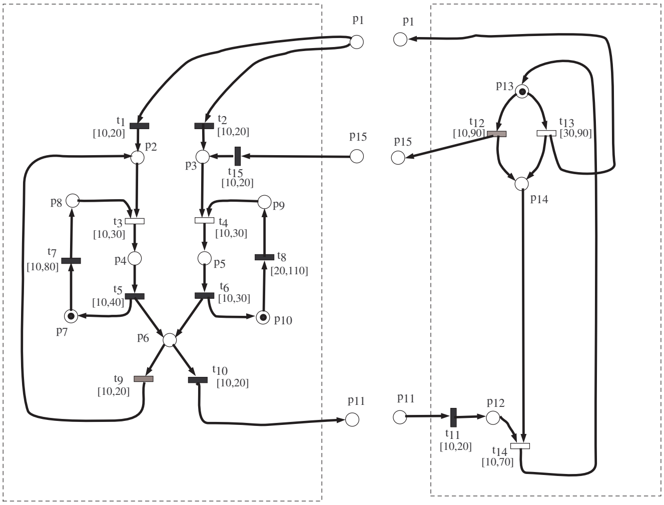


Figure 5.28:

Example 35. *Consider TPN in Fig. 5.10 decomposed in two place-bordered TPNs as displayed in Fig. 5.28. We have that $\mathcal{N}^\theta = \mathcal{N}_i^\theta \cup \mathcal{N}_j^\theta$ with the common border places p_1, p_{11} and p_{15}.*

The observable transitions in component i are $\mathcal{T}_{o_i} = \{t_3, t_4\}$ while all the other transitions are unobservable (silent). The observable transitions have the same observation label, i.e. $l(t_3) = l(t_4)$. The only fault transition in component i is t_9.

In component j the observable transitions are $\mathcal{T}_{o_i} = \{t_{13}, t_{14}\}$ while all the other transitions are unobservable (silent). The observable transitions have the same observation label, i.e. $l(t_{13}) = l(t_{14})$. The only fault transition in component j is t_{12}.

We consider for the distributed setting the following scenario:

- *Ag_i receives the first observation at the time 135*
- *Ag_j receives the first observation at the time 80 and the second observation at the time 200*
- *after the second observation in component j the communication is allowed (i.e. $\theta_{com} = \theta_{obs_{2_j}} = 200$).*

The preliminary analysis of Ag_i.

At the time 135 the first observation is generated by component i and received by Ag_i. At the same time Ag_i computes backwards the minimal time-interval configurations that explain the first observation using only its model knowledge

Considering the case that the observation was generated by the execution of t_3, Ag_1 derives $\overleftarrow{C}_{1_i}(\mathbf{I}_{1_i})$ (Fig. 5.29) in the following way:

- first the untimed backward configuration $\overleftarrow{C}_{1_i}$ is derived as for untimed models.

- then for the input border condition b_1, Ag_i considers $\underline{dob}_{1_i}(b_1) = [0, 135)$. A time configuration is derived as for the centralized case where for e_3 is imposed the constraint that $\theta_{e_3} = 135$ ($I_{1_i}(e_3) = 80$). This implies that the input border conditions has the date of birth time-interval $\underline{dob}_{1_i}(b_1) = [85, 115]$ while $I(e_8) = [105, 125]$

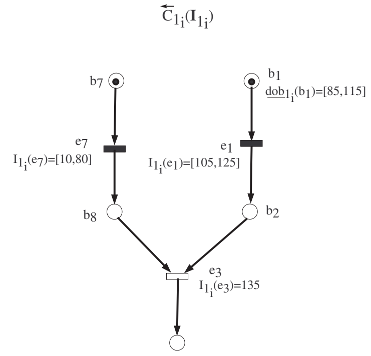

$\overleftarrow{C}_{1_i}(\mathbf{I}_{1_i})$
b7
b1
$\underline{dob}_{1_i}(b_1)$=[85,115]
e7
$I_{1_i}(e_7)$=[10,80]
e1
$I_{1_i}(e_1)$=[105,125]
b8
b2
e3
$I_{1_i}(e_3)$=135

Figure 5.29:

Considering on the other hand the case that the observation was generated by the execution of t_4, Ag_1 derives two minimal time-interval configurations namely $\overleftarrow{C}_{1_{2_i}}(\mathbf{I}_{1_{2_i}})$ (Fig. 5.30) and $\overleftarrow{C}_{2_{2_i}}(\mathbf{I}_{2_{2_i}})$ (Fig. 5.31) considering that a token enter component i via p_{15} and two minimal time-interval configurations namely $\overleftarrow{C}_{1_{3_i}}(\mathbf{I}_{1_{3_i}})$ (Fig. 5.32) and $\overleftarrow{C}_{2_{3_i}}(\mathbf{I}_{2_{3_i}})$ (Fig. 5.33) considering that a token enter component i via the border place p_1.

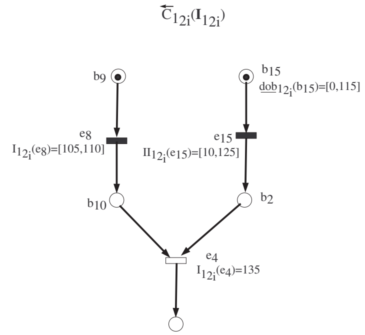

$\overleftarrow{C}_{12i}(\mathbf{I}_{12i})$
b_9
b_{15}
$\underline{dob}_{12_i}(b_{15})=[0,115]$
e_8
$I_{12_i}(e_8)=[105,110]$
e_{15}
$II_{12_i}(e_{15})=[10,125]$
b_{10}
b_2
e_4
$I_{12_i}(e_4)=135$

Figure 5.30:

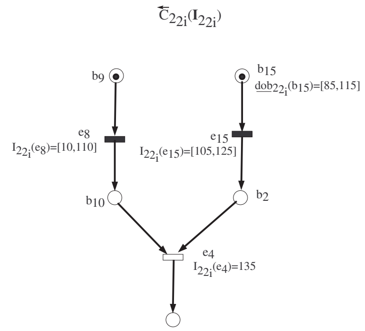

$\overleftarrow{C}_{22i}(\mathbf{I}_{22i})$
b_9
b_{15}
$\underline{dob}_{22_i}(b_{15})=[85,115]$
e_8
$I_{22_i}(e_8)=[10,110]$
e_{15}
$I_{22_i}(e_{15})=[105,125]$
b_{10}
b_2
e_4
$I_{22_i}(e_4)=135$

Figure 5.31:

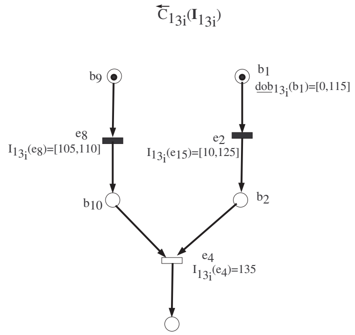

$\overleftarrow{C}_{13i}(\mathbf{I}_{13i})$
b_9
b_1
$\underline{dob}_{13_i}(b_1)=[0,115]$
e_8
$I_{13_i}(e_8)=[105,110]$
e_2
$I_{13_i}(e_{15})=[10,125]$
b_{10}
b_2
e_4
$I_{13_i}(e_4)=135$

Figure 5.32:

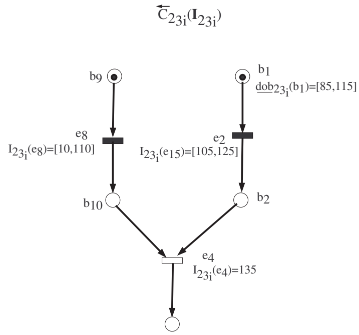

$\overleftarrow{C}_{23i}(\mathbf{I}_{23i})$
b9
b1
dob$_{23_i}$(b1)=[85,115]
e8
I$_{23_i}$(e8)=[10,110]
e2
I$_{23_i}$(e15)=[105,125]
b10
b2
e4
I$_{23_i}$(e4)=135

Figure 5.33:

At the time θ_{com} when the communication is allowed Ag_i extends the minimal time-interval configurations appending only unobservable events and imposing the condition that observable events must be executed only after the time 200.

For $\overleftarrow{C}_{1_i}(\mathbf{I}_{1_i})$ Ag_i derives $C_{1_{1_i}}(\mathbf{I}_{1_{1_i}})$ (Fig. 5.34) and $C_{2_{1_i}}(\mathbf{I}_{2_{1_i}})$ (Fig. 5.35).

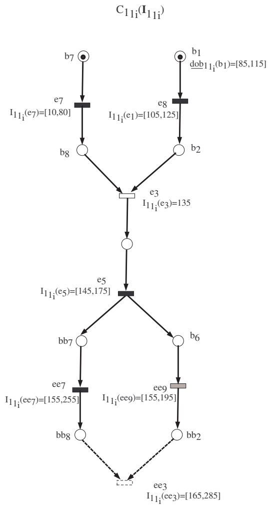

C$_{11i}$($\mathbf{I}_{11i}$)
b7
b1
dob$_{11_i}$(b1)=[85,115]
e7
I$_{11_i}$(e7)=[10,80]
e8
I$_{11_i}$(e1)=[105,125]
b8
b2
e3
I$_{11_i}$(e3)=135
e5
I$_{11_i}$(e5)=[145,175]
bb7
b6
ee7
I$_{11_i}$(ee7)=[155,255]
ee9
I$_{11_i}$(ee9)=[155,195]
bb8
bb2
ee3
I$_{11_i}$(ee3)=[165,285]

Figure 5.34:

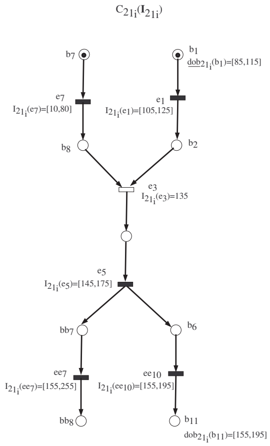
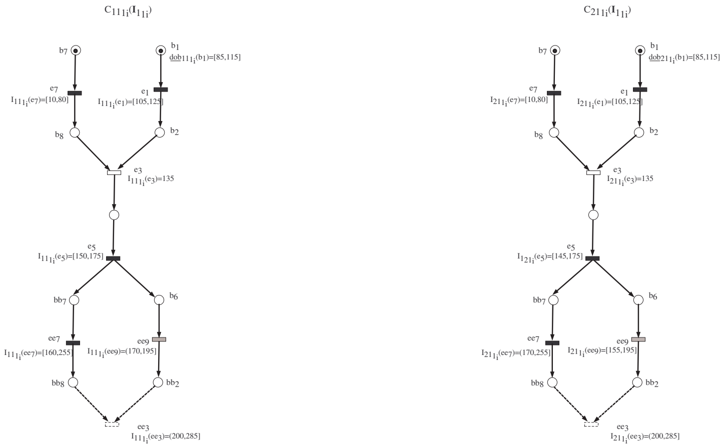

C21i(I21i)
b7
b1
dob21i(b1)=[85,115]
e7
I21i(e7)=[10,80]
e1
I21i(e1)=[105,125]
b8
b2
e3
I21i(e3)=135
e5
I21i(e5)=[145,175]
bb7
b6
ee7
I21i(ee7)=[155,255]
ee10
I21i(ee10)=[155,195]
bb8
b11
dob21i(b11)=[155,195]

Figure 5.35:

For $C_{1_{1_i}}(\mathbf{I}_{1_{1_i}})$ we have that the enabled event ee_3 may be execute before the time 200. Ag_i imposes the constraint that $\vartheta_{ee_3} > 200$ creating two time-interval configurations namely $C_{1_{1_{1_i}}}(\mathbf{I}_{1_{1_{1_i}}}))$ (5.36-l) and $C_{2_{1_{1_i}}}(\mathbf{I}_{2_{1_{1_i}}}))$ (5.36-r).

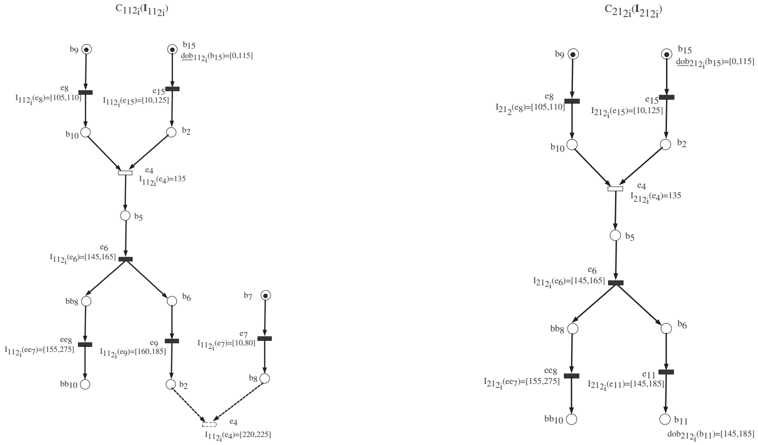
Since at least one of ee_7 and ee_9 must happen after the time 170 we have in $C_{1_{1_{1_i}}}(\mathbf{I}_{1_{1_{1_i}}}))$ that $I_{1_{1_{1_i}}}(ee_9) = [170, 195]$, $I_{1_{1_{1_i}}}(ee_5) = [150, 175]$ and, $I_{1_{1_{1_i}}}(ee_7) = [160, 255]$ while in $C_{2_{1_{1_i}}}(\mathbf{I}_{2_{1_{1_i}}}))$ that $I_{2_{1_{1_i}}}(ee_7) = [170, 255]$.



$C_{111_i}(\mathbf{I}_{11_i})$
b_7
b_1
$\underline{dob}_{111_i}(b_1)=[85,115]$
e_7
$I_{111_i}(e_7)=[10,80]$
e_1
$I_{111_i}(e_1)=[105,125]$
b_8
b_2
e_3
$I_{111_i}(e_3)=135$
e_5
$I_{111_i}(e_5)=[150,175]$
bb_7
b_6
ee_7
$I_{111_i}(ee_7)=[160,255]$
ee_9
$I_{111_i}(ee_9)=(170,195]$
bb_8
bb_2
ee_3
$I_{111_i}(ee_3)=(200,285]$
$C_{211_i}(\mathbf{I}_{11_i})$
b_7
b_1
$\underline{dob}_{211_i}(b_1)=[85,115]$
e_7
$I_{211_i}(e_7)=[10,80]$
e_1
$I_{211_i}(e_1)=[105,125]$
b_8
b_2
e_3
$I_{211_i}(e_3)=135$
e_5
$I_{121_i}(e_5)=[145,175]$
bb_7
b_6
ee_7
$I_{211_i}(ee_7)=(170,255]$
ee_9
$I_{211_i}(ee_9)=[155,195]$
bb_8
bb_2
ee_3
$I_{211_i}(ee_3)=(200,285]$

Figure 5.36:

For $\overleftarrow{C}_{1_{2_i}}(\mathbf{I}_{1_{2_i}})$ Ag_i derives $C_{1_{1_{2_i}}}(\mathbf{I}_{1_{1_{2_i}}})$ (Fig. 5.37-l) and $C_{2_{1_{2_i}}}(\mathbf{I}_{2_{1_{2_i}}})$ (Fig. 5.37-r).



$C_{112_i}(\mathbf{I}_{112_i})$
b_9
b_{15}
$\underline{dob}_{112_i}(b_{15})=[0,115]$
e_8
$I_{112_i}(e_8)=[105,110]$
e_{15}
$I_{112_i}(e_{15})=[10,125]$
b_{10}
b_2
e_4
$I_{112_i}(e_4)=135$
b_5
e_6
$I_{112_i}(e_6)=[145,165]$
bb_8
b_6
b_7
ee_8
$I_{112_i}(ee_7)=[155,275]$
e_9
$I_{112_i}(e_9)=[160,185]$
e_7
$I_{112_i}(e_7)=[10,80]$
bb_{10}
b_2
b_8
e_4
$I_{112_i}(e_4)=[220,225]$
$C_{212_i}(\mathbf{I}_{212_i})$
b_9
b_{15}
$\underline{dob}_{212_i}(b_{15})=[0,115]$
e_8
$I_{212_2}(e_8)=[105,110]$
e_{15}
$I_{212_i}(e_{15})=[10,125]$
b_{10}
b_2
e_4
$I_{212_i}(e_4)=135$
b_5
e_6
$I_{212_i}(e_6)=[145,165]$
bb_8
b_6
ee_8
$I_{212_i}(ee_7)=[155,275]$
e_{11}
$I_{212_i}(e_{11})=[145,185]$
bb_{10}
b_{11}
$dob_{212_i}(b_{11})=[145,185]$

Figure 5.37:

For $\overleftarrow{C}_{2_{2_i}}(\mathbf{I}_{2_{2_i}})$ *Ag_i derives* $C_{1_{2_{2_i}}}(\mathbf{I}_{1_{2_{2_i}}})$ *(Fig. 5.38-l) and* $C_{2_{2_{2_i}}}(\mathbf{I}_{2_{2_{2_i}}})$ *(Fig. 5.38-r).*

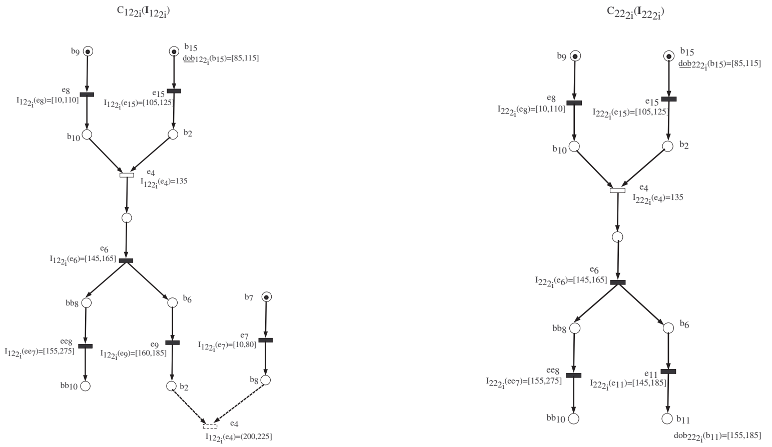


Figure 5.38:

For $\overleftarrow{C}_{1_{3_i}}(\mathbf{I}_{1_{3_i}})$ *Ag_i derives* $C_{1_{1_{3_i}}}(\mathbf{I}_{1_{1_{3_i}}})$ *(Fig. 5.39-l) and* $C_{2_{1_{3_i}}}(\mathbf{I}_{2_{1_{3_i}}})$ *(Fig. 5.39-r).*

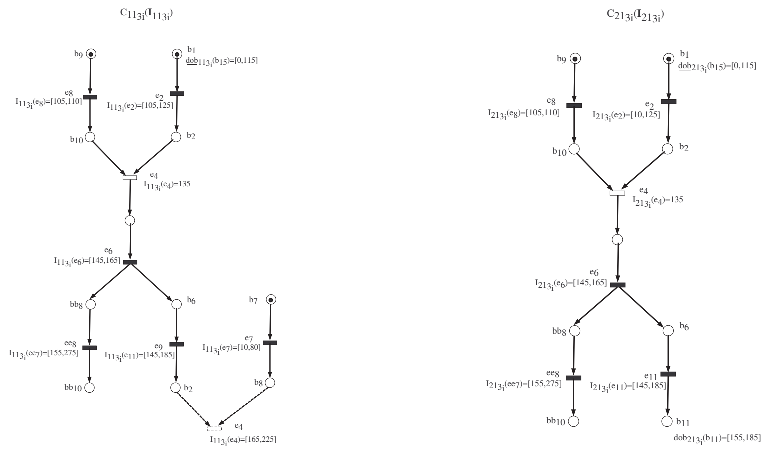


Figure 5.39:

For $\overleftarrow{C}_{2_{3_i}}(\mathbf{I}_{2_{3_i}})$ Ag_i *derives* $C_{1_{2_{3_i}}}(\mathbf{I}_{1_{2_{3_i}}})$ *(Fig. 5.40-l) and* $C_{2_{2_{3_i}}}(\mathbf{I}_{2_{2_{3_i}}})$ *(Fig. 5.40-r).*

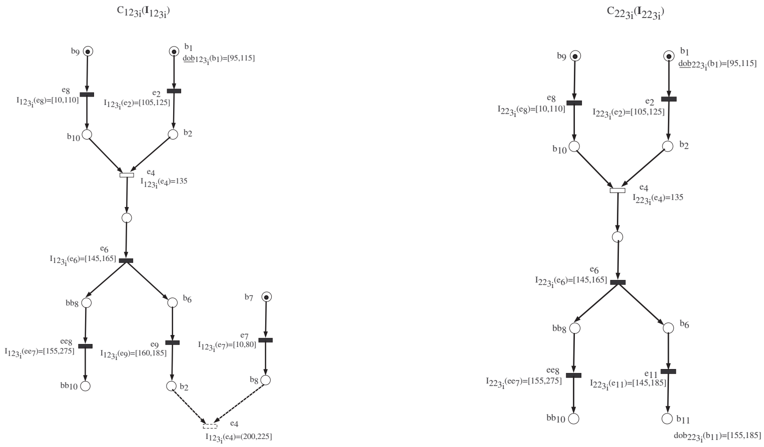


Figure 5.40:

***The preliminary local analysis of** Ag_j.*

For the first observed event at the time 80*,* Ag_j *derives two minimal time-interval configurations* $\overleftarrow{C}_{1_j}(\mathbf{I}_{1_j})$ *(Fig. 5.41-l) and* $\overleftarrow{C}_{2_j}(\mathbf{I}_{2_j})$ *(Fig. 5.41-r).*

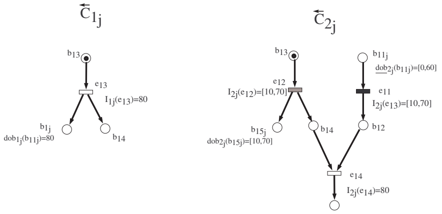


Figure 5.41:

For the second observed event at the time 200 *the local calculations of* Ag_j *is as follows.*

Ag_j *extends the configuration in Fig. 5.41-l to explain the second observation* $\overleftarrow{C}_{1_j}(\mathbf{I}_{1_j})$ *(Fig. 5.42).*

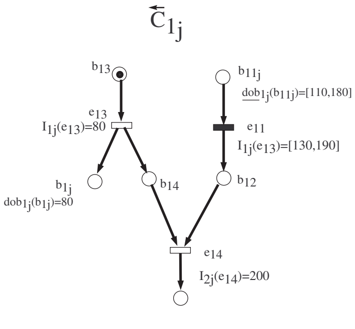


Figure 5.42:

For $\overleftarrow{C}_{2_j}(\mathbf{I}_{2_j})$ *(Fig. 5.41-r),* Ag_j *derives the time-configurations* $\overleftarrow{C}_{1_{2_j}}(\mathbf{I}_{1_{2_j}})$ *(Fig. 5.43-l) and* $\overleftarrow{C}_{2_{2_j}}(\mathbf{I}_{2_{2_j}})$ *(Fig. 5.43-r) that also explain the second observation:*

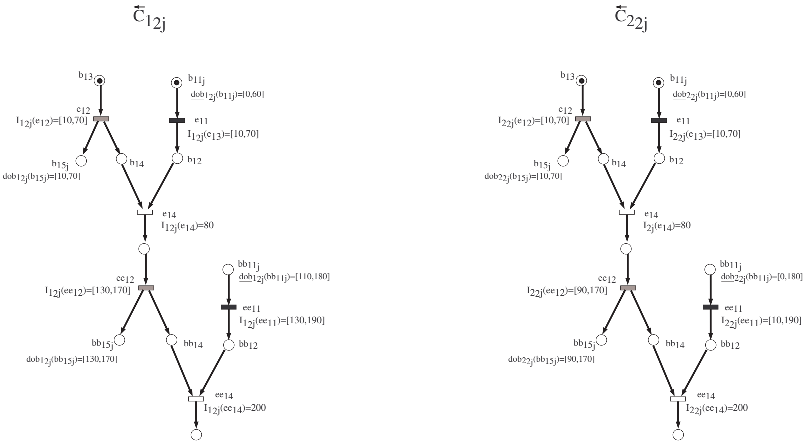

$\overleftarrow{C}_{12j}$
b_{13}
b_{11j}
$\underline{dob}_{12j}(b_{11j})=[0,60]$
e_{12}
$I_{12j}(e_{12})=[10,70]$
e_{11}
$I_{12j}(e_{13})=[10,70]$
b_{15j}
$dob_{12j}(b_{15j})=[10,70]$
b_{14}
b_{12}
e_{14}
$I_{12j}(e_{14})=80$
bb_{11j}
$\underline{dob}_{12j}(bb_{11j})=[110,180]$
ee_{12}
$I_{12j}(ee_{12})=[130,170]$
ee_{11}
$I_{12j}(ee_{11})=[130,190]$
bb_{15j}
$dob_{12j}(bb_{15j})=[130,170]$
bb_{14}
bb_{12}
ee_{14}
$I_{12j}(ee_{14})=200$
$\overleftarrow{C}_{22j}$
b_{13}
b_{11j}
$\underline{dob}_{22j}(b_{11j})=[0,60]$
e_{12}
$I_{22j}(e_{12})=[10,70]$
e_{11}
$I_{22j}(e_{13})=[10,70]$
b_{15j}
$dob_{22j}(b_{15j})=[10,70]$
b_{14}
b_{12}
e_{14}
$I_{2j}(e_{14})=80$
bb_{11j}
$\underline{dob}_{22j}(bb_{11j})=[0,180]$
ee_{12}
$I_{22j}(ee_{12})=[90,170]$
ee_{11}
$I_{22j}(ee_{11})=[10,190]$
bb_{15j}
$dob_{22j}(bb_{15j})=[90,170]$
bb_{14}
bb_{12}
ee_{14}
$I_{22j}(ee_{14})=200$

Figure 5.43:

In $\overleftarrow{C}_{1_{2_j}}(\mathbf{I}_{1_{2_j}})$ we have that $\phi(b_{15}) = \phi(bb_{15})$ and $b_{15} \parallel bb_{15}$ and also $\phi(b_1) = \phi(bb_1)$ and $b_1 \parallel bb_1$.

Since the overall model is 1-safe, Ag_j must impose that constraints that $b_1 \preceq bb_1$ and $b_{15} \preceq bb_{15}$. $b_1 \preceq bb_1$ can be achieved only if $b_1 \preceq b_{15}$ in component i while $b_{15} \preceq bb_{15}$ can be achieved only if $b_{15} \preceq bb_1$ in component i.

This is identically made also in $\overleftarrow{C}_{2_{2_j}}(\mathbf{I}_{2_{2_j}})$. The refined time configurations are displayed in Fig. 5.44.

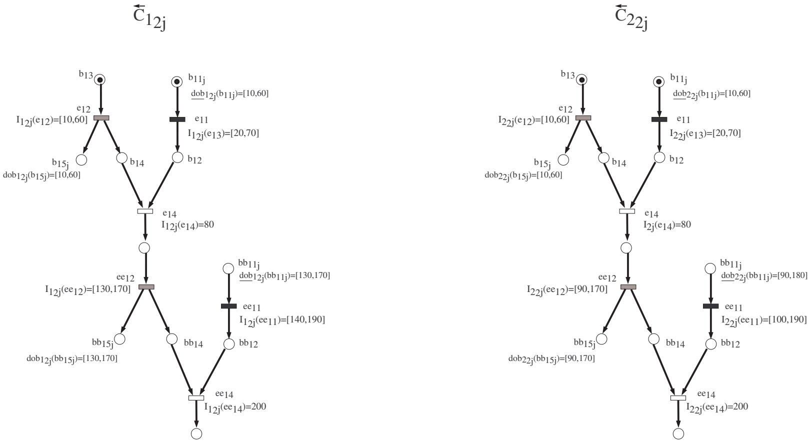

$\overleftarrow{C}_{12j}$
b_{13}
b_{11j}
$\underline{dob}_{12j}(b_{11j})=[10,60]$
e_{12}
$I_{12j}(e_{12})=[10,60]$
e_{11}
$I_{12j}(e_{13})=[20,70]$
b_{15j}
$dob_{12j}(b_{15j})=[10,60]$
b_{14}
b_{12}
e_{14}
$I_{12j}(e_{14})=80$
bb_{11j}
$\underline{dob}_{12j}(bb_{11j})=[130,170]$
ee_{12}
$I_{12j}(ee_{12})=[130,170]$
ee_{11}
$I_{12j}(ee_{11})=[140,190]$
bb_{15j}
$dob_{12j}(bb_{15j})=[130,170]$
bb_{14}
bb_{12}
ee_{14}
$I_{12j}(ee_{14})=200$
$\overleftarrow{C}_{22j}$
b_{13}
b_{11j}
$\underline{dob}_{22j}(b_{11j})=[10,60]$
e_{12}
$I_{22j}(e_{12})=[10,60]$
e_{11}
$I_{22j}(e_{13})=[20,70]$
b_{15j}
$dob_{22j}(b_{15j})=[10,60]$
b_{14}
b_{12}
e_{14}
$I_{2j}(e_{14})=80$
bb_{11j}
$\underline{dob}_{22j}(bb_{11j})=[90,180]$
ee_{12}
$I_{22j}(ee_{12})=[90,170]$
ee_{11}
$I_{22j}(ee_{11})=[100,190]$
bb_{15j}
$dob_{22j}(bb_{15j})=[90,170]$
bb_{14}
bb_{12}
ee_{14}
$I_{22j}(ee_{14})=200$

Figure 5.44:

The preliminary calculations of component i and component j are presented below:

component i	$B^{\theta}(IN_i)$	$B^{\theta}(OUT_i)$
$C_{1_{1_{1_i}}}$	$dob_{1_{1_{1_i}}}(b_1) = [85, 115]$	$\times$
$C_{2_{1_{1_i}}}$	$dob_{2_{1_{1_i}}}(b_1) = [85, 115]$	$\times$
$C_{2_{1_i}}$	$dob_{2_{1_i}}(b_{15}) = [0, 115]$	$dob_{2_{1_i}}(b_{11}) = [155, 195]$
$C_{1_{1_{2_i}}}$	$dob_{1_{1_{2_i}}}(b_{15}) = [0, 115]$	$\times$
$C_{2_{1_{2_i}}}$	$dob_{2_{1_{2_i}}}(b_{15}) = [0, 115]$	$dob_{2_{1_{2_i}}}(b_{11}) = [145, 185]$
$C_{1_{2_{2_i}}}$	$dob_{1_{2_{2_i}}}(b_{15}) = [85, 115]$	$\times$
$C_{2_{2_{2_i}}}$	$dob_{2_{2_{2_i}}}(b_{15}) = [85, 115]$	$dob_{2_{2_{2_i}}}(b_{11}) = [155, 185]$
$C_{1_{1_{3_i}}}$	$dob_{1_{1_{3_i}}}(b_1) = [0, 115]$	$\times$
$C_{2_{1_{3_i}}}$	$dob_{2_{1_{3_i}}}(b_1) = [0, 115]$	$dob_{2_{1_{3_i}}}(b_{11}) = [155, 185]$
$C_{1_{2_{3_i}}}$	$dob_{1_{2_{3_i}}}(b_1) = [95, 115]$	$\times$
$C_{2_{2_{3_i}}}$	$dob_{2_{2_{3_i}}}(b_1) = [95, 115]$	$dob_{2_{2_{3_i}}}(b_{11}) = [155, 185]$

component j	$B^{\theta}(IN_j)$	$B^{\theta}(OUT_j)$
C_{1_j}	$dob_{1_j}(b_{11}) = [110, 180]$	$dob_{1_j}(b_1) = 80$
$C_{1_{2_j}}$	$dob_{1_{2_j}}(b_{11}) = [10, 60]$ $dob_{1_{2_j}}(bb_{11}) = [130, 170]$	$dob_{1_j}(b_{15}) = [10, 60]$ $dob_{1_j}(bb_{15}) = [130, 170]$
$C_{2_{2_j}}$	$dob_{2_{2_j}}(b_{11}) = [10, 60]$ $dob_{2_{2_j}}(bb_{11}) = [110, 180]$	$dob_{2_j}(b_{15}) = [10, 60]$ $dob_{2_j}(bb_{15}) = [90, 170]$

Ag_i and Ag_j exchange information about the border conditions and to check the consistency of their local results.

In this example the only possible consistent pair is $(C_{2_{1_{3_i}}}, C_{1_j})$.

Then Ag_i imposes to $C_{2_{1_{3_i}}}$ the constraints that $dob_{2_{1_{3_i}}}(b_1) = 80$ and $dob_{2_{1_{3_i}}}(b_{11}) = [155, 180]$ while Ag_j imposes to C_{1_j} the constraint that $dob_{1_j}(b_{11}) = [155, 180]$.

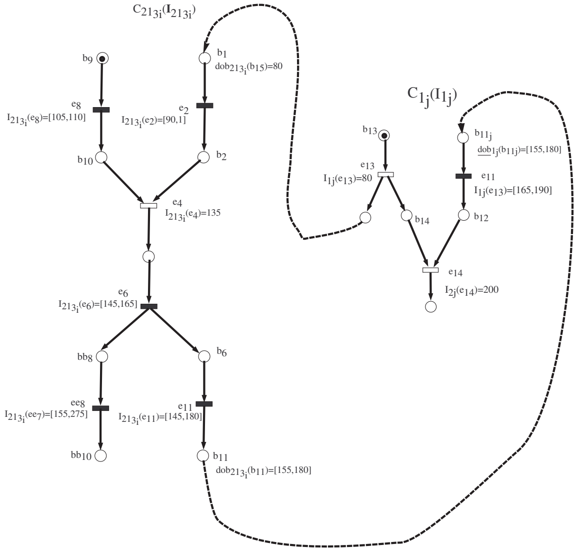


Figure 5.45:

Since neither $C_{2_{13_i}}$ nor C_{1_j} include fault events the diagnosis result for each site is normal $\mathcal{LDR}_i(\mathcal{O}^{\theta}_{\theta_{com}}) = \{\mathtt{N}_i\}$, and $\mathcal{LDR}_j(\mathcal{O}^{\theta}_{\theta_{com}}) = \{\mathtt{N}_j\}$.

At the time 200 the local state of component i is:

- either $M_{0_{1_i}} = \big\{M_{0_{1_i}}(p_8) = 1; M_{0_{1_i}}(p_9) = 1\big\}$ and $ENABLED(M_{0_{1_i}}) = \emptyset$

- or $M_{0_{2_i}} = \big\{M_{0_{1_i}}(p_8) = 1; M_{0_{1_i}}(p_{10}) = 1\big\}$ and $ENABLED(M_{0_{2_i}}) = \{t_8\}$, $FI(t_8) = [200, 275]$

At the time 200 the local state of component j is:

$M_{0_j} = \big\{M_{0_j}(p_{13}) = 1\big\}$ and $ENABLED(M_{0_j}) = \{t_{12}, t_{13}\}$ and
$FI(t_{12}) = [210, 290]$ and $FI(t_{12}) = [230, 290]$

The distributed monitoring continues in the same manner until either the plant stops or the monitoring ends.

Chapter 6

Conclusions

This research is motivated by our interest in designing distributed fault diagnosis algorithms for large and complex systems where unobservable inputs are sent/received between components placed in different sites. The diagnosis problem is viewed in this thesis as part of a broader supervisory architecture taking into account that the diagnosis result is used for taking control/isolation actions to prevent the deterioration of the plant after the occurrence of a fault. The distributed setting that we considered is very general considering that the plant comprises several components that are supervised by local agents that perform local calculation and exchange information among them. To the best of our knowledge the consideration of the case of unobservable interactions between place-bordered PNs is new. In [GL03], [GL05], [BFHJ03], [FBHJ05] the assumption is that the input and output transitions of the border places are non-deterministicly observable. A distributed setting considering unobservable interactions between components was considered in [BL99], [Su04] for distributed diagnosis of a plant modeled by a network of interacting automata. However the analysis of a component modeled as an automaton is a lot simpler that a PN with an uncertain marking.

The lack of observation of the interactions of a component with its neighbours, the unreliability of the communication channels, as well as the requirement that the local agents should be able to provide the diagnosis of their component in any situation make the diagnosis problem very difficult.

It is widely recognized in the AI community that in presence of incomplete knowledge a very efficient method is to reason from effect to causes (abductive reasoning [CMS02], [Hua02]). For the analysis of PN models under partial observation the abductive reasoning simply means to infer starting from the received observation (assumed correct) backwards in order to derive the minimal explanations of the received observation.

Beside its use for designing a distributed diagnosis algorithm we show that the backward analysis can be deployed for the centralized analysis of large PN models. It is well known that for large plants a diagnoser-automaton may become too large to be stored on a computer. This is because for a given sequence of observed events the centralized analysis requires the calculation of the entire set of complete explanations. As a novelty we proposed for the centralized analysis of large PN models the construction of a reduced observer that considers in a given state fewer markings than the classical observer such that all the markings considered by the classical observer are obtained from the markings considered by a reduced observer, by firing unobservable transitions (Section 3.2). The size of the reduced observer is in general a lot smaller than the size of the classical observer and thus it can be stored in a computer and moreover it is possible at any time if required to derive the set of markings estimated by the classical observer.

A fault in a PN model is represented by a choice transition and naturally we assumed that there is no reachable marking from where only fault events are possible to be executed.

Based on this remark we have shown that deriving backwards the set of minimal explanations of the received observation allows one to derive the plant diagnosis result that equals the centralized diagnosis result based on the set of complete explanations in the detection of the faults that for sure happened in the plant (Section 4.2). This makes possible the centralized analysis of very large plants since the complexity of the calculations does not depend on the entire plant size but on the largest sub-net that contains only unobservable events. The efficiency of the method relies on the backward calculations that in general explore unreachable states. The use of place invariants and other heuristics [FRSB02] were found applicable to drive the search.

In this thesis the backward and forward calculations for untimed PN models are performed using the unfolding technique [McM93], [AIN00]. Beside a more efficient calculation of the PN models, a configuration in an unfolding also represents the causality between the events that are assumed executed in a given trace. This allows for checking consistency of local results in the distributed setting.

The backward search is used in the distributed algorithm for deriving the preliminary local calculation of a component (Section 4.3.2). The set of minimal explanations of the local observation includes, beside the sequence of events that must have happened before the observed events, also the minimum number of tokens that must have entered a local component in order to enable the observed events. We showed that if every oriented path that starts in an input place of a component and ends in one of its output places includes at least one observable event then every fault that is detected by a centralized agent to have happened for sure in a component is also detected in the preliminary local diagnosis of the component. If this condition on the path between input and output places is not satisfied then the local preliminary diagnosis can fail to detect some faults that would have been derived by a centralized agent that for sure happened but these faults must be located on the unobservable paths that link the input and the output places of a component. Thus under normal specifications the local control actions that are mandatory to be taken in absence of communication should not be sensitive to this lack of detection.

In Section 4.3.3 we have designed a distributed protocol that allows the local agents to recover the centralized diagnosis result based on local calculation and information exchange. The centralized diagnosis is recovered by consistent pairs of local diagnosis results. For the case of more than two components we have shown that in general the global consistency of the local results cannot be achieved if the information exchange between two neighboring agents includes information only about their common border. This is because they cannot detect circular dependencies between their local results that appear due the presence of unobservable circuits that cross more than two components. However if all the unobservable circuits in the overall plant contain transitions of at most two components then we prove that the distributed diagnosis algorithm terminates after a finite number of iterations providing the centralized diagnosis result as a pair of consistent local diagnosis results (Section 4.3.5).

The results derived for untimed PN were then extended to TPN models. The reason is that timing information allows for more accurate models and consequently it allows for a more accurate diagnosis. However the analysis becomes more complicated because of the explicit consideration of the time as a continuous parameter. Considering the analysis of TPN models based on atomic state classes [YR98] we have presented a monitoring algorithm were the exact plant observation is taken into account by adding extra linear inequalities to the characteristic system of a path in the state class graph that obey the observation (Section 5.4.2).

The timing information may reduce the number of untimed traces in the PN model. However the total number of interleavings of the concurrent events can still be high and the analysis based on atomic state class graph can be intractable even for TPN models of reasonable size. To overcome this limitation we proposed diagnosis algorithms for TPN based on time-configurations (Section 5.5). A time configuration (time process [AL97]) is an untimed configuration endowed with a valuation function that associates with each event in the configuration its execution time. A time configuration is valid if it represents a trace in the original TPN. To check the validity of a valuation requires to check if the execution times of the events in the configurations satifies the characteristic system of inequalities. The characteristic system of a configuration is a system of $(max, +)$-linear inequalities. Thus to derive the set of all valid time processes requires to calculate the entire set of solutions of a system of $(max, +)$-linear inequalities. Solving such a system requires to explicitly consider the possible interleavings between concurrent events that have common successors exactly what was intended to avoid.

We have proposed a more efficient way to derive the set of all valid time-interval configurations. The set of all solutions of the characteristic system of a configuration (the set of all valid times) is obtained as a cover of subsets of solutions such that each sub-set of solutions has the time independence property for the concurrent events in the configuration. The time independence property of a subset of solutions of the characteristic system of a configuration says that: *given any set of concurrent events in the configuration and fixing the execution times of their predecessors, their executions times belong to a hyper-rectangle in high dimensional space.* The execution time-intervals for the events are obtained from the smallest hyperbox (of dimension equal with number of events in the configuration) that includes a given subset of solutions of the characteristic system. The algorithm that derives such a partition of the solution set of the characteristic system of a configuration is based on the propagation of temporal constraints on the execution time intervals of the events in the configuration. The partial order relation of the events in the configuration is exploited. We have presented on-line monitoring algorithm that can handle the addition of extra inequalities (constraints) whenever an observation is received.

The distributed diagnosis algorithm for TPN models (Section 5.7) considers a distributed setting very similar to the one that we consider for the untimed models. However some simplifying assumptions are needed. The overall TPN model is assumed untimed 1-safe and free-choice. These conditions are required in order to meet the requirement that the preliminary local calculations can be used for taking some control/isolation actions.

As for the untimed PN models, the preliminary local calculations of the TPN model of a component give rise to a major difficulty namely the analysis of a model with uncertain initial conditions. We adapted the backward unfolding method to Time Petri Net models (Section 5.6) and then we showed for the case of two components that the distributed algorithm recovers the diagnosis result of a centralized agent by consistent pairs of local diagnosis results (Section 5.7.5).

Future work

We plan to further extend the results of this thesis in the following ways:

1. to include probabilistic information in order to have a classification of likelihood of the faults whose detection is uncertain. [Vol02], [BFH03], [Haa03] propose probabilistic models of discrete events systems where the concurrency is filtered out. Probabilities are assigned to transitions in a PN, and a probabilistic measure is derived over the space of all possible configurations in the PN model. Probabilistic methods were proposed in [Vol02], [BFH03] for free-choice PNs, and then extended

in [Haa03] for general PNs. The solution in [Haa03] is to decompose the PN model in clusters and to consider a probability distribution over the set of *"words"* in the cluster. The methods consider a centralized setting. However the distributed diagnosis of PN models that include probabilistic information is an open problem even for a simplified setting.

2. to relax the assumption made in the distributed setting for TPN that all the components are free-choice TPNs, by finding adequate conditions that allow for performing local preliminary calculations in absence of communication.

3. to investigate the problem of minimization of the information exchanged between the local agents [BvS02], [RLL03]. When the plant is supervised by a large number of agents the question *"to whom to send a message ?"* becomes very important. A solution for this problem may be to incorporate some knowledge to each local agent regarding the plant structure in its vicinity, e.g. abstract models of its neighbouring components and then to investigate the trade off between the extra knowledge of a local agent versus the reduction in the amount of information that is exchanged.

4. to investigate the application of the distributed diagnosis in electrical power systems considering a broader architecture that includes control/isolation modules [JB04b].

5. to consider the case of more complex interactions between the local components (e.g. via logical guards [NAH^{+}98], [JB06]) or via common sub-nets with unobservable transitions.

6. to investigate the application of the Extended Linear Complementarity Problem [SM95], [SM96] to derive the entire set of solutions of the characteristic system of a configuration as an alternative of the method that we have proposed and then to identify sub-classes of TPN models to whom each of the two methods is appropriate.

Chapter 7

Pseudo-code Algorithms

7.1 Algorithm for forward reachability

Algorithm 20 Reach_Tree(M_0) *(Carp-Miller algorithm)*

```
Require: $\langle \mathcal{N}, M_0 \rangle$
Ensure: $\mathcal{RT}_{\mathcal{N}}(M_0)$
initialize $x_0 = M_0$ {the root node}
$SET = \{x_0\}; VISIT = \emptyset$
while $SET \neq \emptyset$ do
  pick up $x \in SET$
  if $ENABLED(x) = \emptyset$ then
    $x$ is a terminal node
  else
    for all $t \in ENABLED(x)$ do
      create a new node $x'$ and draw an arc from $x$ to $x'$ labeled $t$
      if $x' \notin VISIT$ then
        $SET = SET \cup \{x'\}$
        draw an arc from $x$ to $x'$ labeled $t$
      else
        merge $x'$ with $x'' \in VISIT$ s.t. $x' = x''$
        draw an arc from $x$ to $x''$ labeled $t$
      end if
    end for
    $VISIT = VISIT \cup \{x\}$
  end if
end while
```

7.2 Algorithm for backward reachability

Algorithm 21 Back_Reach - *(Main Program)*

```
Require: $\mathcal{N}$, $M_{ini}$, $M_{fin}$
Ensure: $\mathcal{UC}_{\mathcal{N}}(M_{fin}, M_{ini}); \mathcal{UL}_{\mathcal{N}}(M_{fin}, M_{ini})$
$mark(node\_root) = M_{fin}$ ; $Pred(node\_root) = \emptyset$; $Succ(node\_root) = nil$
create $node\_root = (mark, Pred, Succ)$
$SET = \{node\_root\}; TERMIAL = \emptyset;$
$VISIT\_SOL = \emptyset; VISIT\_NOSOL = \emptyset; VISIT\_UKW = \emptyset;$
$\mathcal{UC}_{\mathcal{N}}(M_{fin}, M_{ini}) = \emptyset; \mathcal{UL}_{\mathcal{N}}(M_{fin}, M_{ini}) = \emptyset$
while $SET \neq \emptyset$ do
  $choose\_node\_cur(SET)$
  if $break = false$ then
    $check\_solution(node\_cur)$
    if $new\_node = true$ then
      $make\_new\_nodes(node\_cur)$
    end if
    $sort(SET)$
  else
    $SET = \emptyset$
  end if
end while
for all $node\_ter \in TERMINAL$ do
  $mark(node\_root) \overset{\sigma_{uo}}{\rightsquigarrow} mark(node\_ter)$
  $\mathcal{UC}_{\mathcal{N}}(M_{fin}, M_{ini}) = \mathcal{UC}_{\mathcal{N}}(M_{fin}, M_{ini}) \cup \{mark(node\_ter)\}$
  $\mathcal{UL}_{\mathcal{N}}(M_{fin}, M_{ini}) = \mathcal{UL}_{\mathcal{N}}(M_{fin}, M_{ini}) \cup \{\sigma_{uo}\}$
end for
```

Algorithm 22 Procedure *choose_node_cur*

```
Require: $SET$
Ensure: $node\_cur, break$
for all $node \in SET$ do
  $condition(node)$
  if $chosen = true$ then
    $node\_cur = node$
    $break = false$
    $exit\_loop\_for$
  else
    $break = true$
  end if
end for
```

Algorithm 23 Procedure *condition*

```
Require: $node$
Ensure: $chosen$
for all $node' \in VISIT\_UKW$ do
    if $mark(node') < mark(node)$ then
        $chosen = false$
        $exit\_loop\_for$
    end if
end for
$chosen = true$
```

Algorithm 24 Procedure *check_sol*

```
Require: $node\_cur$
Ensure: $new\_node$
if $mark(node\_cur) \leq M_0$ then
    add $node\_cur$ to $TERMIAL$
    add $node\_cur$ to $VISIT\_SOL$
    $propagate\_sol(node\_cur)$
    if $B\_ENABLE(mark(node\_cur)) \cap \mathcal{T}_{uo} = \emptyset$ then
        $new\_node = false$
    else
        $new\_node = true$
    end if
else
    if $B\_ENABLE(mark(node\_cur)) \cap \mathcal{T}_{uo} = \emptyset$ then
        add $node\_cur$ to $VISIT\_NOSOL$
        $propagate\_no\_sol(node\_cur)$
    else
        add $node\_cur$ to $VISIT\_UKW$
        $new\_node = true$
    end if
end if
remove $node\_cur$ from $SET$
```

Algorithm 25 Procedure $propagate\_sol$

```
Require: $node\_cur$
$PROP\_SOL = \{node\_cur\}$
while $PROP\_SOL \neq \emptyset$ do
    node=HEAD(PROP_SOL)
    for all $node' \in Pred(node)$ do
        add $node'$ to $VISIT\_SOL$
        remove $node'$ from $VISIT\_UKW$
        add $node'$ to $PROP\_SOL$
    end for
    remove $node$ from $PROP\_SOL$
end while
```

Algorithm 26 Procedure $propagate\_no\_sol$

```
Require: $node\_cur$
$PROP\_NOSOL = \{node\_cur\}$
while $PROP\_NOSOL \neq \emptyset$ do
    node=HEAD(PROP_NOSOL)
    for all $node' \in VISIT\_UKW$ do
        if $mark(node) \leq mark(node')$ then
            add $node'$ to $PROP\_NOSOL$
        end if
    end for
    $REMOVE\_UP = \{node\}$
    while $REMOVE\_UP \neq \emptyset$ do
        for all $node' \in Succ(node)$ do
            remove $node$ from $Pred(node')$
            if $Pred(node') = \emptyset$ then
                add $node'$ to $REMOVE\_UP$
            end if
        end for
    end while
    $REMOVE\_DOWN = \{node\}$
    while $REMOVE\_DOWN \neq \emptyset$ do
        for all $node' \in Pred(node)$ do
            remove $node$ from $Succ(node')$
            if $Succ = \emptyset$ then
                add $node'$ to $REMOVE\_DOWN$
            end if
        end for
        remove $node$ from $REMOVE\_DOWN$
    end while
    remove $node$ from $PROP\_NOSOL$
end while
```

Algorithm 27 Procedure $make\_new\_node$

```
Require: $node\_cur$
for all $t \in B\_Enabled(mark(node\_curr)) \cap \mathcal{T}_{uo}$ do
    $mark\_new = mark(node\_cur) \ominus Post(t, \cdot) + Pre(\cdot, t)$
    $fail = false$
    for all $node \in VISIT\_NOSOL$ do
        if $mark(node\_new) > mark(node)$ then
            $fail = true$
            $exit\_loop\_for$
        end if
    end for
    for all $node \in VISIT\_UKW$ do
        if $mark(node\_new) = mark(node)$ then
            add $node\_cur$ to $Pred(node)$
            add $node$ to $Succ(node\_cur)$
            $fail = true$
            $exit\_loop\_for$
        end if
    end for
    if $fail = false$ then
        create $node\_new$
        $mark(node\_new) = mark\_new$
        $Succ(node\_cur) = node\_new$
        $Pred(node\_new) = node\_cur$
        add $node\_new$ to $SET$
    end if
end for
```

7.3 Minimal explanations for PNs with unobservable trap circuits

Algorithm 28 Procedure Min_Conf

```
Require: $t_{cur_\nu}$
Ensure: $\underline{C}_\nu(t^o)$
create an event $e = \phi(t_{cur_\nu})$
$\underline{C}_\nu = \underline{C}_\nu \cup e \cup \{e^\bullet\}$
for all $p \in \{{}^\bullet\phi(e)\}$ do
  if $\exists b \in AVAILABLE[\nu]$ s.t. $\phi(b) = p$ then
    create arc from $b$ to $e$ and remove $b$ from $AVAILABLE[\nu]$
  else
    add $b$ ($\phi(b) = p$) to $SET[\nu]$ s.t. $b = HEAD(SET[\nu])$
  end if
end for
while $SET[\nu] \neq \emptyset$ and $abort[\nu] \neq true$ do
  $b = HEAD(SET[\nu])$
  for all $t' \in {}^\bullet\phi(b) \cap \mathcal{T}_{uo}$ s.t. $t' \notin \phi(Pred[\nu])$ do
    add $t'$ to $B\_BRANCH$
  end for
  if $B\_BRANCH = \emptyset$ then
    $abort = true$
  else
    choose $t \in B\_BRANCH$ remove $t$ from $B\_BRANCH$ and make $t_{cur_\nu} = t$
    while $B\_BRANCH \neq \emptyset$ do
      $\nu_{max} = \nu_{max+1}$; choose $t' \in B\_BRANCH$; $t_{cur_{\nu_{max}}} = t'$; remove $t'$
      $\underline{C}_{\nu_{max}} = \underline{C}_\nu$; $SET[\nu_{max}] = SET[\nu]$;
      $AVAILABLE[\nu_{max}] = AVAILABLE[\nu]$; $Pred[\nu_{max}] = Pred[\nu]$;
    end while
    create an event $e = \phi(t_{cur_\nu})$
    $\underline{C}_\nu = \underline{C}_\nu \cup e$
    find the largest subset $X^{con}_{B_{C_\nu}} \subset SET[\nu]$ s.t. $\phi(X^{con}_{B_{C_\nu}}) \subset t^\bullet$ and $b \in X^{con}_{B_{C_\nu}}$
    draw an arc from $e$ to each $b \in X^{con}_{B_{C_\nu}}$
    remove $X^{con}_{B_{C_\nu}}$ from $SET[\nu]$
    for all $p \in t^\bullet \setminus \phi(X^{con}_{B_{C_\nu}})$ do
      $b' = \phi(p)$, $\underline{C}_\nu = \underline{C}_\nu \cup \{b'\}$
      draw an arc from $e$ to $b'$ and add $b'$ to $AVAILABLE[\nu]$
    end for
    for all $p \in \phi({}^\bullet e)$ do
      if $\exists\, b \in AVAILABLE[\nu]$ s.t. $\phi(b) = p$ and ${}^\bullet b \notin PRED[\nu]$ then
        draw an arc from $b$ to $e$ and remove $b$ from $AVAILABLE[\nu]$
      else
        add $b$ to $SET[\nu]$ s.t. $HEAD(SET[\nu]) = b$
        $NEW = true$
      end if
    end for
    if $AVAILABLE[\nu] = \emptyset$ then
      $abort[\nu] = true$
    else
      if $NEW = false$ then
        remove $\phi(e)$ from $Pred[\nu]$
      end if
    end if
  end if
end while
```

Bibliography

[AIN00] P.A. Abdulla, S.P. Iyer, and A. Nylen. On unfolding unbounded petri nets. In Springer-Verlag, editor, *Computer Aided Verification, LNCS*, volume 1855, 2000.

[AL97] T. Aura and J. Lilius. Time processes of time petri nets. *18th International Conference on Application and Theory of Petri Nets (ATPN'97) - LNCS*, 1248:136–155, 1997.

[All83] J. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26:832–843, 11 1983.

[BFH03] A. Benvensite, E. Fabre, and S. Haar. Markov nets: Probabilistic models for distributed and concurrent systems. *IEEE Transactions on Automatic Control*, 48(11):1936–1950, 2003.

[BFHJ03] A. Benvensite, E. Fabre, S. Haar, and C. Jard. Diagnosis of asynchronous discrete event systems, a net unfolding approach. *IEEE Transactions on Automatic Control*, 48(5), 2003.

[BJ03a] R.K. Boel and G. Jiroveanu. Modular reachability analysis for time petri nets with guarded transitions. In *ADSH IFAC Conference*, St.-Malo, France, 2003.

[BJ03b] R.K. Boel and G. Jiroveanu. Petri nets model-based fault section detection and diagnosis system in electrical power networks. In *6th IPEC Conference*, Singapore, 2003.

[BJ04] R.K. Boel and G. Jiroveanu. Distributed contextual diagnosis for very large systems. In *7th Workshop on Discrete Event Systems (WODES'04)*, Reims, France, 2004.

[BL99] P. Baroni and G. Lamperti. Diagnosis of large active systems. *Artificial Intelligence*, 101(1):135–183, 1999.

[BM83] B. Berthomieu and M. Menasche. An enumerative approach for analyzing time petri nets. *IFIP Congess, Paris*, 1983.

[BSC02] A. Boufaied, A. Subias, and M. Combacau. Chronicle modeling by petri nets for distributed detection of process failures. In *IEEE Conference on Systems, Management and Cybernetics*, Hammamet, Tunisie, 2002.

[BV02] B. Berthomieu and F. Vernadat. State class construction for branching analysis of time petri nets. *LAAS Report 02130*, March 2002.

[BvS02] R.K. Boel and J.H. van Schuppen. Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers. In *Workshop on Discrete Event Systems (WODES'02)*, Zaragoza, Spain, 2002.

[CJ05] T. Chatain and C. Jard. Time supervision of concurrent systems using symbolic unfoldings of time petri nets. *Int. Conference on Formal Modeling and Analysis of Time Systems*, Uppsala, Sweden, 2005.

[CKV95] J. Cardoso, L.A. Kunzle, and R. Valette. Petri net based reasoning for the diagnosis of dynamic discrete event systems. In *6th International Fuzzy Systemes Association World Congress*, pages 333–336, July 1995.

[CL99] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kuwer Academic Publishers, 1999.

[CMS02] A. Ciampolini, P. Mello, and S. Storari. Distributed medical diagnosis with abductive logic agents. In *European Conference in Artificial Intelligence (ECAI'02)*, Reims, France, 2002.

[DE95] J. Desel and J. Esparza. *Free Choice Petri Nets*. Cambridge University Press, 1995.

[DLT00] R. Debouk, S. Lafortune, and D. Teneketzis. Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Journal of Discrete Event Dynamic Systems*, 10(1-2):33–86, January 2000.

[DLT03] R. Debouk, S. Lafortune, and D. Teneketzis. On the effect of communication delays in failure diagnosis of descentralized discrete event systems. *Journal of Discrete Event Dynamic Systems*, 3:263–289, 2003.

[DRvB04] G. Delzanno, J-F. Raskin, and L. van Begin. Covering sharing trees: A compact data structure for parameterized verification. *Software Tools for Technology Transfer*, 5(2):268–297, 2004.

[Eng91] J. Engelfriet. Branching processes of petri nets. *Acta Informatica*, 28(6):575–591, 1991.

[ERV96] J. Esparza, S. Romer, and W. Volger. An improvement of mcmillan's unfolding algorithm. *LNCS*, 1055:87–106, March 1996. Springer-Verlag.

[Esp94] J. Esparza. Model checking using net unfoldings. *Science of Computer Programming*, 23(2):151–194, 1994.

[FBHJ05] E. Fabre, A. Benvensite, S. Haar, and C. Jard. Distributed monitoring of concurrent and asynchronous systems. *Journal of Discrete Event Dynamic Systems*, 15(1):33–84, March 2005.

[Fra90] P. Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - a survey and some new results. *Automatica*, 26(3):459–474, 1990.

[FRSB02] A. Finkel, J-F. Raskin, M. Samuelidis, and L. Van Begin. Monotonic extensions of petri nets: forward and backward search revisited. *Electronic Notes on Theoretical Computer Science*, 68(6), 2002.

[GBT05] M. Ghazel, M. Bigand, and A. Toguyéni. A temporal-contraint based approach for monitoring of DESs under partial observation. In *16th IFAC Triennial World Congress*, Prague, 2005.

[GCS05] A. Giua, D. Corona, and C. Seatzu. State estimation of λ-free labeled petri nets with contact-free nondeterministic transitions. *Journal of Discrete Event Dynamic Systems*, 15(1):85–108, March 2005.

[GL03] S. Genc and S. Lafortune. Distributed diagnosis for DES using petri nets. In *Conference on Applications and Theory of Petri Nets (ATPN'03)*, Eindhoven, The Nederlands, 2003.

[GL05] S. Genc and S. Lafortune. A distributed algorithm for on-line diagnosis of place-bordered petri nets. In *16th IFAC Triennial World Congress*, Prague, 2005.

[GS02] A. Giua and C. Seatzu. Observability of place/transition nets. *IEEE Transactions On Automatic Control*, 47(9):1424–1437, 2002.

[GW93] P. Godefroid and P. Wolper. Using partial orders for efficient verification of deadlock freedom and safety properties. In *Formal Methods in Systems Design*, volume 2(2), pages 149–164, 1993.

[HA02] L.E. Holloway and J. Ashley. Diagnosis of condition systems using causal structure. In *American Control Conference*, Alaska, USA, 2002.

[Haa03] S. Haar. Probabilistic cluster unfoldings for petri nets. Technical Report 1517, IRISA, Rennes, France, 2003.

[HB95] H. Hulgaard and S.M. Burns. Efficient timing analysis of a class of petri nets. *Computer Aided Verification*, 1995.

[Hua02] Y.-C. Huang. Abductive reasoning network based diagnosis system for fault section estimation in power systems. *IEEE Transaction on Power Delivery*, 17(2):369–374, 2002.

[HV99] C.N. Hadjicostis and G.C. Verghese. Monitoring discrete event systems using petri net embeddings. *Application and Theory of Petri Nets - LNCS*, 1639:188–207, 1999. Springer-Verlag.

[HV00] C.N. Hadjicostis and G. Verghese. Power systems monitoring based on relay and circuit breaker. In *IEE Procedings on Generation, Transmission and Distribibution*, volume 147, pages 299–303, 2000.

[JB03] G. Jiroveanu and R.K. Boel. A distributed approach for fault detection and diagnosis based on petri nets. In *CESA'03*, Lille, France, 2003.

[JB04a] G. Jiroveanu and R. K. Boel. Contextual analysis of petri nets for distributed applications. In *MTNS'04*, Leuven, Belgium, 2004.

[JB04b] G. Jiroveanu and R.K. Boel. A common architecture for distributed diagnosis and wide-area back-up protection. In *Developments in Power System Protection, Eighth IEE International Conference*, Amsterdam, The Netherlands, 2004.

[JB05a] G. Jiroveanu and R.K. Boel. Distributed diagnosis for petri net models with unobservable interactions via common places. In *44th Conference on Decision and Control (CDC'05)*, Sevilla, Spain, 2005.

[JB05b] G. Jiroveanu and R.K. Boel. Distributed diagnosis of large interacting systems. In *16th International Workshop on Principles of Diagnosis (DX'05)*, Monterrey, CA, USA, 2005.

[JB06] G. Jiroveanu and R.K. Boel. A distributed approach for fault detection and diagnosis based on time petri nets. *Mathematics and Computers in Simulation*, 2006. (to appear).

[JK04] S. Jiang and R. Kumar. Failure diagnosis of discrete event systems with linear-time temporal logic specifications. *IEEE Transactions on Automatic Control*, 49, 2004.

[KKSW02] J. Kurine, X. Koutsoukos, R. Su, and W.M. Wonham. Distributed diagnosis for qualitative systems. In *Workshop on Discrete Event Systems (WODES'02)*, Zaragoza, Spain, 2002.

[KKZ01] J. Kurine, X. Koutsoukos, and F. Zhao. Distributed diagnosis of networked, embedded systems. In *12th International Workshop on Principles of Diagnoses (DX'01)*, USA, 2001.

[KTKT95] A. Kondratyev, A. Taubin, M. Kishinevsky, and S. Ten. Analysis of petri nets by ordering relations. *Technical Report*, TR:95-0-002, 1995. University of Aizu.

[LA94] L.Portinale and C. Anglano. B-w analysis: a backward reachability analysis for diagnostic problem solving suitable to parallel implementation. In *15th Int. Conference on Application and Theory of Petri Nets, LNCS*, volume 815, pages 39–58, Zaragoza, Spain, 1994.

[Lil98] J. Lilius. Efficient state space search for time petri nets. *Electronic Notes in Theoretical Computer Science*, 18, 1998. Elsevier Science.

[Lun00] J. Lunze. Diagnosis of quantised systems by means of timed discrete-event representations. *IEEE Transactions of Systems, Managment and Cybernetics*, A(30):322–335, 2000.

[LZ02] G. Lamperti and M. Zanella. Diagnosis of discrete-event system from uncertain temporal observations. *Artificial Intelligence*, 138:91–137, 2002.

[LZ03] G. Lamperti and M. Zanella. *Diagnosis of active systems*. Kluwer Academic Publisher, 2003.

[McI98] S. McIlraith. Explanatory diagnosis: Conjecturing actions to explain observation. In 6^{th} *Int. Conf. on Principles of Knowledge Representation and Reasoning*, 1998.

[McM93] K.L. McMillan. *Symbolic model checking*. Kluwer Academic Publishers, Dordrecht, 1993.

[MDHM04] S.D.J. McArthur, E.M. Davidson, J.A. Hossack, and J.R. McDonald. Automating power systems fault diagnosis through multi-agent systems. In *37th Annual Hawaii International Conference on System Science*, pages 59–66, 2004.

[Mer74] P. Merlin. *A study of recoverability of computer science*. PhD thesis, University of California, Irvine, USA, 1974.

[MLRV02] C. Mancel, P. Lopez, N. Riviére, and R. Valette. Relationships between petri nets and constraint graphs: application to manufacturing. In *15th IFAC Triennial World Congress*, 2002.

[Mur89] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings IEEE*, 77(4):541–580, April 1989.

[NAH+98] J.L. Nielsen, H.R. Andersen, H. Hulgaard, G. Behrmann, K. Kristoffersen, and K.G. Larsen. Verification of large state/event systems using compositionality and dependency analysis. In *International Conference on Tools and Algorithms for Construction and Analysis*, pages 201–216, 1998.

[OW90] C.M. Ozvern and A.S. Willsky. Observability of discrete event systems. *IEEE Transactions on Automatic Control*, 35(7):797–806, 1990.

[PCR01] Y. Pencolé, M.-O. Cordier, and L. Rozé. Incremental decentralized diagnosis approach for the supervision of a telecommunication network. In *Workshop on Principles of Diagnosis (DX'01)*, Italy, 2001.

[Pen00] Y. Pencolé. Decentralized diagnoser approach: application to telecommunication networks. In *In 11th International Workshop on Principles of Diagnosis - DX'00*, pages 185–192, 2000. Morelia, MX.

[Pro02] G. Provan. A model-based diagnosis framework for distributed systems. In *13th International Workshop on Principles of Diagnoses (DX'02)*, USA, 2002.

[Ram74] C. Ramchandani. Analysis of asynchronous concurrent systems by timed petri nets. Technical report, Massachussets Institute of Technology, 1974.

[Rei87] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[RLL03] K. Rudie, S. Lafortune, and F. Lin. Minimal communication in a distributed discrete-event systems. *IEEE Transactions On Automatic Control*, 48(6):957–975, 2003.

[SJ94] V.S. Srinivasan and M.A. Jafari. Fault detection/monitoring using timed petri nets. *IEEE Transactions On Systems Managment and Cybernetics*, 23(4):1155–1162, 1994.

[SM95] B. De Schutter and B. De Moor. The extended linear complementarity problem. *Mathematical Programming*, 71(3):289–325, Dec. 1995.

[SM96] B. De Schutter and B. De Moor. A method to find all solutions of a system of multivariable polynomial equalities and inequalities in the max algebra. *Discrete Event Dynamic Systems: Theory and Applications*, 6(2):115–138, Mar. 1996.

[SR92] P.D. Stotts and J.C. Ruiz. Synthesizing a global net state from synchronized local pieces. Technical Report TR-92-13, University of Florida, 1992.

[SSL$^+$95] M. Sampath, R. Sengupta, S. Lafortune, S. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete event systems. *IEEE Transactions On Automatic Control*, 40(9):1555–1575, 1995.

[Su04] R. Su. *Distributed diagnosis for Discrete Event Systems*. PhD thesis, University of Toronto, 2004.

[SW04] R. Su and W.M. Wonham. A model of component consistency in distributed diagnosis. In *7th Workshop on Discrete Event Systems (WODES'04)*, Reims, France, 2004.

[SY96] A. Semenov and A. Yakovlev. Verification of asynchronous curcuits using time petri-net unfolding. *Proceedings ACM/IEEE Design Automation Conference*, 1996.

[Val90] A. Valmari. Stubborn sets for reduced state space generation. In Lecture Notes in Computer Science, editor, *In Advances in Petri Nets*, volume 483, pages 491–515. Springer Verlag, 1990.

[Val94] A. Valmari. Compositional analysis with place-bordered subnets. In *15th International Conference on Application and Theory of Petri Nets, LNCS*, volume 815, pages 531–547, 1994.

[Vic01] E. Vicario. Static analysis and dynamic steering of time-dependent systems. *IEEE Transactions on Software Engineering*, Vol. 27(8):728–748, August 2001.

[Vol02] H. Volzer. Randomized non-sequential processes and distributed adversaries. Technical report, The University of Queensland, July 2002. Technical Report.

[WD00] J. Wang and Y. Deng. Reachability analysis of real-time systems using time petri nets. *IEEE Transactions On Systems Managment and Cybernetics*, 30(5):725–736, 2000.

[YOYS92] C.L. Yang, H. Onamoto, A. Yokoyama, and Y. Sekine. Expert system for fault section estimation of power systems using time sequence information. *International Journal of Electrical Power and Energy Systems*, 14(2/3):225–232, 1992.

[YR98] T. Yoneda and H. Ryuba. Ctl model checking of time petri nets using geometric regions. *EICE Transaction on Information & Systems*, E99-D:1–11, 1998.

[YS97] T. Yoneda and H. Schlingloff. Effcient verification of parallel real-time systems. *Journal of Formal Methods and Design*, 11(2):187–215, 1997.

[ZKW03] S.H. Zad, R.H. Kwong, and W.M. Wonham. Fault diagnosis in discrete-event systems: Framework and model reduction. *IEEE Transactions On Automatic Control*, 48(7), 2003.

List of Notation

$\mathbb{N}$	the set of natural numbers including 0	13
$\mathbb{N}_+$	the set of natural numbers except 0	13
$\mathbb{Q}_+$	the set of rational numbers except 0	13
X_μ	multi-set over the set X	13
$\preceq$	partial order relation	14
Σ	a finite set of symbols	15
ϵ	the empty sequence	15
Σ^*	Kleene closure of the alphabet Σ	15
$\mathcal{L}$	a language over a set of symbols	15
$Pwr(X)$	the power set of a set X	15
$\Sigma(s)$	the set of symbols that appear in a string s	15
$\Sigma_\mu(s)$	the multi-set over the set of symbols of a string s	15
$\equiv_{\Sigma_\mu}$	equivalence relation between strings	16
$(\Sigma, \preceq)$	partial ordered set	16
$\langle\Sigma\rangle_\preceq$	the set of linearizations of $(\Sigma, \preceq)$	16
G	deterministic automaton	17

$mark(C)$	the marking that corresponds with $CUT(C)$	32
$\mathcal{U}_{\mathcal{N}}(M_0)$	the net unfolding of a PN $\langle\mathcal{N}, M_0\rangle$	33
$\underline{C}(e)$	the minimal configuration of an event e in $\mathcal{U}_{\mathcal{N}}(M_0)$	33
$UR_{\mathcal{N}}(M)$	the unobservable reach of a marking M	49
$\mathtt{CO}$	classical observer for a PN $\langle\mathcal{N}, M_0\rangle$	50
$\mathcal{L}_{\mathtt{CO}}$	the language of a classical observer-automaton	50
$\mathtt{RO}$	reduced observer for a PN $\langle\mathcal{N}, M_0\rangle$	52
$\underline{C}(\mathcal{O}_n)$	minimal configuration that explains $\mathcal{O}_n$	55
$\underline{\mathcal{C}}(\mathcal{O}_n)$	the set of minimal configuration that explain $\mathcal{O}_n$	55
$\underline{\mathcal{L}}_{\mathcal{N}}(\mathcal{O}_n)$	the set of minimal explanations of the observation $\mathcal{O}_n$	55
$\overset{t}{\rightsquigarrow}$	backward firing of a transition t	55
$\overleftarrow{\mathfrak{O}}$	reverse occurrence net	60
$\overleftarrow{B}$	the set of conditions in $\overleftarrow{\mathfrak{O}}$	60
$\overleftarrow{E}$	the set of events in $\overleftarrow{\mathfrak{O}}$	60
$\overleftarrow{\sharp}$	backward conflict relation	61
$\overleftarrow{\sharp}_{ac}$	backward auto-conflict relation	61
$\overleftarrow{C}$	a configuration in the occurrence net $\overleftarrow{\mathfrak{O}}$	62
$\overleftarrow{\mathcal{C}}$	the set of all configurations in $\overleftarrow{\mathfrak{O}}$	62
$\overleftarrow{\mathcal{U}}_{\mathcal{N}}$	the backward net unfolding of a PN $\langle\mathcal{N}, M_0\rangle$	62
$\mathcal{D}_{\mathcal{N}}(\mathcal{O}_n)$	centralized diagnosis of a PN $\langle\mathcal{N}, M_0\rangle$	71

$\mathcal{DR}_{\mathcal{N}}$	centralized diagnosis result of a PN $\langle \mathcal{N}, M_0 \rangle$	71
$\underline{\mathcal{D}}_{\mathcal{N}}(\mathcal{O}_n)$	centralized diagnosis based on minimal explanations	73
$\underline{\mathcal{DR}}_{\mathcal{N}}$	diagnosis result based on minimal explanations	73
J	the set of components	88
$\mathcal{P}_{IN_i}$	the set of input places of component i	89
$\mathcal{P}_{OUT_i}$	the set of output places of component i	89
$\theta_{t_{k_i}}$	the time when the observed event $t^o_{k_i}$ is executed in component i	90
θ_{com}	the time when the communication between the local agents is allowed	99
$\mathcal{C}_i(\mathcal{O}^i_{\theta_{com}})$	the set of preliminary configurations in component i	101
$\mathcal{E}_i(\mathcal{O}^i_{\theta_{com}})$	the set of preliminary explanations in component i	101
$\mathcal{LPD}_i$	the local preliminary diagnosis of component i	101
$\psi_{\ell_{\nu_i \nu_j}}$	the interpretation function of the border-conditions of C_{ν_i} and C_{ν_j}	105
$\Psi_{\nu_i \nu_j}$	the set of all interpretation functions of C_{ν_i} and C_{ν_j}	105
$\mathcal{LD}^{gcon}_i$	the global consistent local diagnosis of component i	115
$\mathcal{N}^\theta$	a Time Petri Net	132
I^s_t	the static interval of transition t	132
$\langle \mathcal{N}^\theta, M^\theta_0 \rangle$	a TPN $\mathcal{N}^\theta$ with an initial marking M^θ_0	132
θ^{en}_t	the time when transition t becomes enabled	132
S_θ	a state in a TPN $\langle \mathcal{N}^\theta, M^\theta_0 \rangle$	132
θ_t	the time when transition t fires	132

FI	the firing interval function	132
τ^{θ}	a time trace in a TPN	133
$\mathcal{S}$	the set of all reachable states in a TPN	133
$\mathcal{L}^{\theta}_{\mathcal{N}^{\theta}}(M^{\theta}_{0})$	the set of all legal time traces in a TPN	134
$\mathcal{L}_{\mathcal{N}^{\theta}}(M^{\theta}_{0})$	the untimed support language of a TPN	134
$\mathcal{O}^{\theta}_{\xi}$	the observation of a TPN up to the time ξ	138
$\mathcal{O}^{\theta}_{n}$	the observation after the n^{th} observed event	138
$\mathcal{D}(\mathcal{O}^{\theta}_{\xi})$	the diagnosis of a TPN at the time ξ	138
K_{τ}	the characteristic system of a trace τ	140
SC_{τ}	the linear state class corresponding with the trace τ	142
$\mathcal{SC}$	the set of all linear state classes	142
$LSCG$	the linear state class graph	142
$\mathcal{ASC}$	the set of all atomic state classes	145
$ASCG$	the atomic state class graph	145
ρ	a path in the atomic state class graph $ASCG$	145
Θ	the valuation of the execution times of the events within a configuration	151
$K_{C^{\theta}}$	the characteristic system of a configuration C	167
$C(\mathbf{I})$	time-interval configuration in a free-choice TPN	167
$I(e)$	the execution time-interval of an event e in $C(\mathbf{I})$	167
$\widehat{\theta}_{v}$	the discarding time of $\widehat{C}_{v}(\mathbf{I}_{v})$	169

$\mathcal{TC}(\mathcal{O}^{\theta}_{\xi})$	the set of time interval configurations at the time ξ that obey the plant observation $\mathcal{O}^{\theta}_{\xi}$	172
$\breve{e}$	a conflicting event of a configuration	191
$\breve{E}_C$	the set of conflicting events of a configuration C	191
$C(\mathbf{Y})$	time-interval configuration in a general TPN	192