**UNIVERSITÉ LIBRE DE BRUXELLES**
**Faculté des Sciences**
**Département d'Informatique**

# Distribution of reactive systems

**MEUTER Cédric**

Mémoire présenté vue de l'obtention du diplome d'étude approfondie en informatique

**Année académique 2002-2003**

# Contents

# Acknowledgement

This work has been supervised by Prof. Jean-François Raskin, in collaboration with Dr. Franck Cassez. We wish to address them many thanks for their invaluable support. We would also like to thank the members of our jury, i.e. Prof. Thierry Massart and Prof. Jean Cardinal.

This document is a revised version, where several mistakes and typos have been corrected. We would like to thank Prof. Raymond Devillers for a careful review of the original document that allowed to highlight these mistakes and typos.

This work has been funded by the *Région de Bruxelles Capitale*[1] (grand no. RBC-BR 227/3298) and by *Macq Electronique*, a leading company the design of industrial process control[2].

This document has been typeset under LaTeX2$\varepsilon$, using the `ulbthesis` document class, and all figures were made using MetaPost.

[1] `http://www.bruxelles.irisnet.be/`
[2] `http://www.macqel.be`

# Introduction

In this paper, we focus on reactive systems. Reactive systems react to certain events by performing some actions, in order to keep the controlled environment in a desirable state. For instance, a temperature control system will react to the temperature being too low by turning on a heater. In simple systems, the events monitored by the systems and the actions to be performed are centralized. However, in most industrial systems, the sensors and actuators (inputs and outputs) are geographically distributed. In the design of such distributed systems, some error may be introduced by the distributed nature of these inputs and outputs. One approach to deal with this is to specify the system in a centralized manner, where the inputs and outputs are all accessible in one central location, and then automatically synthesize a distributed systems that is behaviorally equivalent to the initial centralized one. Then if the synthesis process is correct, any property verified by the centralized system will also be verified by the distributed one.

This problem of distributing a centralized specification has already been studied. In [Mor99], the problem is examined on the model of asynchronous systems. An asynchronous system is basically a deterministic labelled transition system along with an independence relation over its alphabet. The problem studied there is to find a synchronized product of deterministic labelled transition systems that is isomorphic to the centralized asynchronous system. In [CMT99], they study the distribution problem on the more general model of labelled transition systems (not necessarily deterministic). They study the problem of finding a synchronized product of labelled transition system that is bisimilar to the centralized model. Note that, in both [Mor99] and [CMT99], the synchronized product operates on common events. In [SEM03], they examine the following problem. Given a centralized specification of acceptable behavior, represented as a language accepted by a deterministic labelled transition system, they try to find an asynchronous automaton accepting a sublanguage of that language. A asynchronous automaton is a finite automaton that can easily be distributed into communicating local automata. However, the communication paradigm used between these local automata is more restrictive than simple synchronization on common events. Indeed, in order to take a transition, the local automata have to communicate not only the event they synchronize on, but also the local states they are in. Finally note that some other related problems of decentralized observability and controllability are studied in [RW92, RW95, Tri01b] on the model of finite transition systems.

In this paper, we introduce a new model for reactive systems where the inputs and outputs are clearly separated and study the automatic distribution on this model. Given a centralized model of a reactive system, and given the locations of the inputs and outputs, we propose a method, based on [CMT99], to automatically synthesize a distributed version of this model that is behaviorally equivalent to the original specification. The components of the distributed systems will communicate through synchronization on common events, and the behavioral equivalence we will use is bisimulation.

### Plan of the work

The paper is organized as follows. We start, in chapter 1, by introducing the model of Action Mealy Machines that we use to model reactive systems. We follows, in chapter 2, by presenting the problem of distribution in its most general form. In chapters 3 and 4 we study two particular subclasses of this problem where the distribution is only based respectively on the location of outputs, and on the location of the inputs. Finally, in chapter 5 we explain how the methods developed in those two chapters can be combined to give a solution to the general problem.

# Chapter 1

# Action Mealy Machines

In this chapter, we introduce a new model for reactive systems : Action Mealy Machines. This model was inspired by the model of FIFFO-AUTOMATON that we studied in [Meu02]. This model of Action Mealy Machines is more abstract than the one we defined in [Meu02]. The reason for that, is that FIFFO-AUTOMATON are too specific, and we wanted a lighter model to address the distribution problem. However, we strongly believe that every result presented in this document can be adapted to FIFFO-AUTOMATON. The rest of this chapter is organized as follows. First in section 1.1, we define the model of Action Mealy Machines. Then, in section 1.2, we present how Action Mealy Machine can be combined using synchronized product. Finally, in section 1.3, we discuss behavior equivalences over this model.

## 1.1 Action Mealy Machines

To model reactive systems, we introduce Action Mealy Machines, a special class of Mealy Machines. Intuitively, a Mealy Machine is a finite automaton for which each transition is triggered by an input and produces outputs. In the case of Action Mealy Machines (AMM), an input models the occurrence of an event received from the environment and the corresponding output produced is a finite set of actions to be performed as a reaction to the event occurrence. The order in which those actions are performed is not important (i.e the actions are totally independent). We formalize this as follows.

**Definition 1.1 - Action Mealy Machine**
An Action Mealy Machine (AMM) is a tuple $\mathcal{M} = (S_{\mathcal{M}}, s^0_{\mathcal{M}}, \Sigma_{\mathcal{M}}, A_{\mathcal{M}}, \delta_{\mathcal{M}}, \lambda_{\mathcal{M}})$ where:

1. $S_{\mathcal{M}}$ is a finite set of states,
2. $s^0_{\mathcal{M}}$ is the initial state,
3. $\Sigma_{\mathcal{M}}$ is a finite set of events,

4. $A_{\mathcal{M}}$ is a finite set of actions,

5. $\delta_{\mathcal{M}} : S_{\mathcal{M}} \times \Sigma_{\mathcal{M}} \rightarrow S_{\mathcal{M}}$ is a partial transition function,

6. $\lambda_{\mathcal{M}} : S_{\mathcal{M}} \times \Sigma_{\mathcal{M}} \rightarrow 2^{A_{\mathcal{M}}}$ is a partial output function; $\lambda_{\mathcal{M}}(s, e)$ is defined if and only if $\delta_{\mathcal{M}}(s, e)$ is defined.

When the control is in a state $s \in S_{\mathcal{M}}$, if an event $e \in \Sigma_{\mathcal{M}}$ occurs, if $\delta_{\mathcal{M}}(s, e)$ is defined, the control moves to $\delta_{\mathcal{M}}(s, e)$ and produces a set of actions given by $\lambda_{\mathcal{M}}(s, e)$. We will note $s \xrightarrow{e}_{\mathcal{M}}$ the fact that $\delta_{\mathcal{M}}(s, e)$ is defined. Similarly, we will note $s \xrightarrow{e/\mathcal{A}}_{\mathcal{M}} s'$ the fact that $\delta_{\mathcal{M}}(s, e) = s'$ and that $\lambda_{\mathcal{M}}(s, e) = \mathcal{A}$. We will also note $out_{\mathcal{M}}(s) = \{e \in \Sigma_{\mathcal{M}} | s \xrightarrow{e}_{\mathcal{M}}\}$ the set of events accepted in state $s$. A run of an AMM $\mathcal{M}$ is a sequence of transitions $\sigma = s^0_{\mathcal{M}} \xrightarrow{e_1/\mathcal{A}_1}_{\mathcal{M}} s_1 \xrightarrow{e_2/\mathcal{A}_2}_{\mathcal{M}} s_2 \ldots s_{n-1} \xrightarrow{e_n/\mathcal{A}_n}_{\mathcal{M}} s_n$. A state $s$ of $S_{\mathcal{M}}$ is reachable if there exists a run of $\mathcal{M}$ ending in $s$.

Note that Action Mealy Machines are deterministic by definition. The determinism hypothesis seems reasonable in our case, because we are dealing with industrial controllers, which are deterministic by nature. Therefore, the Action Mealy Machines can be used to model a wide variety of useful systems.

**Example 1.1**
Figure 1.1, presents an AMM $\mathcal{M}$, modeling a temperature control system. When the control system is activated, we assume that the heater is off. In the initial state $s_1$, if the sensor indicates that the temperature is too low, the heater is turned on and the control moves to state $s_2$. This is modeled by the transition $s_1 \xrightarrow{\text{too\_low}/\{\text{heater\_on}\}}_{\mathcal{M}} s_2$. In state $s_2$, where the heater is on, if the sensor indicates that a certain acceptable temperature is reached, the heater is turned off and the control moves to the initial state $s_1$. This is modeled by the transition $s_2 \xrightarrow{\text{too\_high}/\{\text{heater\_off}\}}_{\mathcal{M}} s_1$.

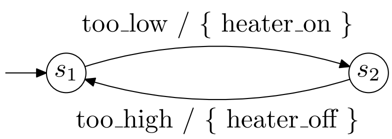



Figure 1.1: AMM modeling a temperature control system

## 1.2 Synchronized product

Because we will be dealing with distributed systems, we must define the behavior of two AMM's $\mathcal{M}_1$ and $\mathcal{M}_2$ reacting in parallel to the same environment. This behavior can be modeled by a single AMM which is the synchronized product of $\mathcal{M}_1$ and $\mathcal{M}_2$. As usual, the synchronizations between both AMM's are made on the events they share. But, in our model, when a synchronized transition is taken, both $\mathcal{M}_1$ and $\mathcal{M}_2$ will produce actions. Thus, in the synchronized product, it is natural to produce the actions produced by both $\mathcal{M}_1$ and $\mathcal{M}_2$. The synchronized product is formalized hereafter.

**Definition 1.2 - Synchronized product**
The synchronized product of two AMM's $\mathcal{M}_1$ and $\mathcal{M}_2$ noted $\mathcal{M}_1 \times \mathcal{M}_2$ is an AMM

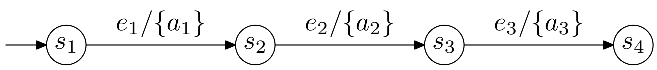
$$(S_{\mathcal{M}_1} \times S_{\mathcal{M}_2}, (s^0_{\mathcal{M}_1}, s^0_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_1} \cup \Sigma_{\mathcal{M}_2}, A_{\mathcal{M}_1} \cup A_{\mathcal{M}_2}, \delta_{\mathcal{M}_1 \times \mathcal{M}_2}, \lambda_{\mathcal{M}_1 \times \mathcal{M}_2})$$

where $\forall (s_1, s_2) \in S_{\mathcal{M}_1} \times S_{\mathcal{M}_2}, \forall e \in \Sigma_{\mathcal{M}_1} \cup \Sigma_{\mathcal{M}_2}$:

$$\delta_{\mathcal{M}_1 \times \mathcal{M}_2}((s_1, s_2), e) = \begin{cases} (\delta_{\mathcal{M}_1}(s_1, e), \delta_{\mathcal{M}_2}(s_2, e)) & \text{if } e \in out_{\mathcal{M}_1}(s_1) \cap out_{\mathcal{M}_2}(s_2) \\ (\delta_{\mathcal{M}_1}(s_1, e), s_2) & \text{if } e \in out_{\mathcal{M}_1}(s_1) \wedge e \notin \Sigma_{\mathcal{M}_2} \\ (s_1, \delta_{\mathcal{M}_2}(s_2, e)) & \text{if } e \in out_{\mathcal{M}_2}(s_2) \wedge e \notin \Sigma_{\mathcal{M}_1} \\ undefined & \text{otherwise} \end{cases}$$

$$\lambda_{\mathcal{M}_1 \times \mathcal{M}_2}((s_1, s_2), e) = \begin{cases} \lambda_{\mathcal{M}_1}(s_1, e) \cup \lambda_{\mathcal{M}_2}(s_2, e) & \text{if } e \in out_{\mathcal{M}_1}(s_1) \cap out_{\mathcal{M}_2}(s_2) \\ \lambda_{\mathcal{M}_1}(s_1, e) & \text{if } e \in out_{\mathcal{M}_1}(s_1) \wedge e \notin \Sigma_{\mathcal{M}_2} \\ \lambda_{\mathcal{M}_2}(s_2, e) & \text{if } e \in out_{\mathcal{M}_2}(s_2) \wedge e \notin \Sigma_{\mathcal{M}_1} \\ undefined & \text{otherwise} \end{cases}$$

**Example 1.2**
Figure 1.2 presents an example of synchronized product. Figures 1.2(a) and 1.2(b) presents two AMM's $\mathcal{M}_1$ and $\mathcal{M}_2$ with respective alphabets $\Sigma_{\mathcal{M}_1} = \{e_1, e_2, e_3\}$ and $\Sigma_{\mathcal{M}_2} = \{e_1, e_4, e_3\}$. In the synchronized product $\mathcal{M}_1 \times \mathcal{M}_2$, presented in figure 1.2(c), the synchronizations are made on common events, that is $\Sigma_{\mathcal{M}_1} \cap \Sigma_{\mathcal{M}_2} = \{e_1, e_2\}$. For the actions, we can, for instance, observe that from state $(s_1, s_5)$, on event $e_1$, both $a_1$ from $\mathcal{M}_1$ and $a'_1$ from $\mathcal{M}_2$ are performed.

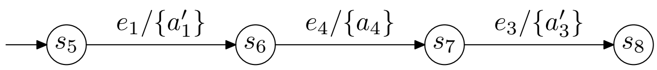



(a) $\mathcal{M}_1$

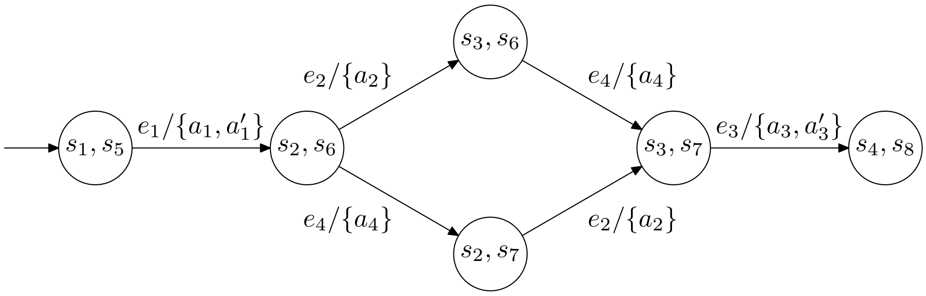
Figure 1.2: Example of synchronized product

## 1.3 Behavior equivalences

To address the distribution problem, it is essential that we define some sort of behavioral equivalence between AMM's. Indeed, we need a way to determine that the distributed system is correct, that




(b) $\mathcal{M}_2$




(c) $\mathcal{M}_1 \times \mathcal{M}_2$

Figure 1.2: Example of synchronized product (cont'd)

is behaviorally equivalent to the centralized system. One well known form of behavior equivalence is language equivalence (or trace equivalence). It can be formalized as follows.

**Definition 1.3 - Language**

Let $\mathcal{M}$ be a AMM. A trace of $\mathcal{M}$ is a word $w = (e_1, \mathcal{A}_1) \cdot (e_2, \mathcal{A}_2) \cdot ... \cdot (e_n, \mathcal{A}_n)$ on $(\Sigma_{\mathcal{M}} \times A_{\mathcal{M}})^*$ such that there exists a run $s^0_{\mathcal{M}} \xrightarrow{e_1/\mathcal{A}_1}_{\mathcal{M}} s_1 \xrightarrow{e_2/\mathcal{A}_2}_{\mathcal{M}} s_2 \ ... \ s_{n-1} \xrightarrow{e_n/\mathcal{A}_n}_{\mathcal{M}} s_n$ in $\mathcal{M}$. We define the language of $\mathcal{M}$, noted $\mathcal{L}_{\mathcal{M}}$ as the set of all traces of $\mathcal{M}$.

Two AMM's $\mathcal{M}_1$ and $\mathcal{M}_2$ are language equivalent if and only if $\mathcal{L}_{\mathcal{M}_1} = \mathcal{L}_{\mathcal{M}_2}$. Note that for any AMM $\mathcal{M}$, by definition, $\mathcal{L}_{\mathcal{M}}$ is prefix-closed and therefore always contains the empty trace $\epsilon$. For a word $w = (e_1, \mathcal{A}_1) \cdot (e_2, \mathcal{A}_2) \cdot ... \cdot (e_n, \mathcal{A}_n)$ on $(\Sigma_{\mathcal{M}} \times A_{\mathcal{M}})^*$, the length of $w$, noted $|w|$, is $n$, and we note, $s \xrightarrow{w}_{\mathcal{M}} s'$ the fact that there exists a sequence of transitions $\sigma = s \xrightarrow{e_1/\mathcal{A}_1}_{\mathcal{M}} s_1 \xrightarrow{e_2/\mathcal{A}_2}_{\mathcal{M}} s_1 \ ... \ s_{n-1} \xrightarrow{e_n/\mathcal{A}_n}_{\mathcal{M}} s'$.

On finite automata, another well known form of behavior equivalence is bisimulation. We extend this notion to AMM's. We introduce action bisimulation which is slightly more restrictive than a traditional bisimulation, in the sense that it forces two action bisimilar states not only to react to the same events as in traditional bisimulation, but also to produce the same sets of actions when reacting to those events. The action bisimulation can be formalized as follows.

**Definition 1.4 - Action Bisimulation**

Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two AMM's. A binary relation $\mathcal{B} \subseteq S_{\mathcal{M}_1} \times S_{\mathcal{M}_2}$ is an action bisimulation relation if and only if, $\forall s_1 \in S_{\mathcal{M}_1}$, $\forall s_2 \in S_{\mathcal{M}_2}$, if $\mathcal{B}(s_1, s_2)$, then:

1. $\forall e \in \Sigma_{\mathcal{M}_1}$, if $s_1 \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1} s_1'$, then $s_2 \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_2} s_2'$ and $\mathcal{B}(s_1', s_2')$
2. $\forall e \in \Sigma_{\mathcal{M}_2}$, if $s_2 \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_2} s_2'$, then $s_1 \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1} s_1'$ and $\mathcal{B}(s_1', s_2')$

Two AMM's $\mathcal{M}_1$ and $\mathcal{M}_2$ are action bisimilar, noted $\mathcal{M}_1 \equiv_B \mathcal{M}_2$, if and only if $\Sigma_{\mathcal{M}_1} = \Sigma_{\mathcal{M}_2}$ and if there exists an action bisimulation relation $\mathcal{B}$ such that $\mathcal{B}(s^0_{\mathcal{M}_1}, s^0_{\mathcal{M}_2})$. Actually, since AMM's are deterministic by nature, action bisimulation and language equivalence coïncide. Given two AMM's $\mathcal{M}_1$ and $\mathcal{M}_2$, $\mathcal{M}_1 \equiv_B \mathcal{M}_2$ if and only if $\mathcal{L}_{\mathcal{M}_1} = \mathcal{L}_{\mathcal{M}_2}$. Therefore, in the rest of the paper, we will use both forms of behavior equivalence interchangeably.

# Chapter 2

# General distribution

In this chapter, we present the main problem addressed in this document, the general distribution problem. This chapter is organized as follows. First we recall some fundamental notions, in section 2.1. We follow, in section 2.2, by formalizing and discussing the general distribution problem. In this problem, we only consider two execution sites. That is why, in section 2.3, we explain how any distribution method over two execution sites can be used to distribute a system over more than two execution sites. Finally, in section 2.4, we discuss our approach to solve the problem.

## 2.1 Preliminaries

Before we introduce the general distribution problem, we need to recall the notion of partition.

**Definition 2.1 - Partition**
Given a set $S$, a partition of $S$ is a non-empty set of subsets of $S$, $P = \{S_1, S_2, ..., S_n\}$ such that $\forall i, j \in \{1..n\}$, if $i \neq j$, then $S_i \cap S_j = \emptyset$, and such that $\bigcup_{i \in \{1..n\}} S_i = S$.

We note $[s]_P$ the partition $S_i \in P$ containing $s$. Moreover, given a set $S$, an equivalence relation $\mathcal{E}$ over $S$ defines a partition $P_{\mathcal{E}}$ given by the set of all equivalence classes of $\mathcal{E}$, that is for all $s \in S$ $[s]_{\mathcal{E}} = \{ \ s' \mid \mathcal{E}(s, s') \ \}$.

## 2.2 The problem

Let us consider a reactive system modeled by an AMM $\mathcal{M}$. In $\mathcal{M}$, all events and actions are implicitly centralized. However, as stated in the introduction, most of the time, those events and actions are physically distributed over several execution sites. Thus, we would like to distribute the system described by $\mathcal{M}$ over several execution sites. Of course, it is imperative for the distribution process to be correct, that is, the distributed system must be behaviorally equivalent to the system described by $\mathcal{M}$. We can formulate the problem a follows.

**Problem 2.1 - General distribution problem**
Given an AMM $\mathcal{M}$, a partition $\{A_1, A_2\}$ of $A_{\mathcal{M}}$ and two subsets $\Sigma_1$, $\Sigma_2$ of $\Sigma_{\mathcal{M}}$ such that $\Sigma_1 \cup \Sigma_2 = \Sigma_{\mathcal{M}}$, do there exist two AMM $\mathcal{M}_1$ and $\mathcal{M}_2$ with $A_{\mathcal{M}_1} = A_1$, $A_{\mathcal{M}_2} = A_2$, $\Sigma_{\mathcal{M}_1} = \Sigma_1$, $\Sigma_{\mathcal{M}_2} = \Sigma_2$ such that $\mathcal{M}_1 \times \mathcal{M}_2 \equiv_B \mathcal{M}$.

In problem 2.1, $A_1$ and $A_2$ represent the sets actions to be executed on the two execution sites. Most of the times, the location of an action is forced by the location of the outputs it uses (executing an action might even simply be to set an output to a certain value). Thus, $A_1$ and $A_2$ reflect, in some sense, the location of the outputs of the systems. Similarly, $\Sigma_1$ and $\Sigma_2$ represent the sets of events that can be monitored on the two execution sites. Thus, they reflect the location of the inputs of the system. Note that $\Sigma_1$ and $\Sigma_2$ are not assumed disjoint. Indeed, most of the times, the distributed AMM's will need to communicate (synchronize) in order for the distributed system to be behaviorally equivalent to the original centralized one. Of course, in the real physical implementation, an input can only be located on one site. For the actions, since they are not involved in any communication, it is not necessary to allow one action to be executed by both execution sites. Therefore, we assume $A_1$ and $A_2$ to be disjoint. Note that we used action bisimulation as a behavior equivalence. But, as explained in chapter 1, we can used language equivalence in the same manner.

## 2.3 More than two sites

In the general distribution problem defined in the previous section, we only consider two execution sites. If there are more than two execution sites, we can use the following result.

**Theorem 2.1**
The synchronized product of AMM's is congruent to action bisimulation. Given four AMM's $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$, $\mathcal{M}_4$, if $\mathcal{M}_1 \equiv_B \mathcal{M}_2$ and $\mathcal{M}_3 \equiv_B \mathcal{M}_4$ then $\mathcal{M}_1 \times \mathcal{M}_3 \equiv_B \mathcal{M}_2 \times \mathcal{M}_4$.

**Proof**
Let $\mathcal{B}_{(1,2)}$ be a bisimulation relation between $\mathcal{M}_1$ and $\mathcal{M}_2$ and $\mathcal{B}_{(3,4)}$, a bisimulation relation between $\mathcal{M}_3$ and $\mathcal{M}_4$. Let $\mathcal{B} \subseteq (S_{\mathcal{M}_1 \times \mathcal{M}_3} \times S_{\mathcal{M}_2 \times \mathcal{M}_4})$ be such that $\mathcal{B}((s_1, s_3), (s_2, s_4))$ if and only if $\mathcal{B}_{(1,2)}(s_1, s_2)$ and $\mathcal{B}_{(3,4)}(s_3, s_4)$. We prove that $\mathcal{B}$ is a bisimulation relation between $\mathcal{M}_1 \times \mathcal{M}_3$ and $\mathcal{M}_2 \times \mathcal{M}_4$. Indeed, $\forall (s_1, s_3) \in S_{\mathcal{M}_1 \times \mathcal{M}_3}$, $(s_2, s_4) \in S_{\mathcal{M}_2 \times \mathcal{M}_4}$ such that $\mathcal{B}((s_1, s_3), (s_2, s_4))$, we have that:

1. $\forall e \in \Sigma_{\mathcal{M}_1 \times \mathcal{M}_3}$, if $(s_1, s_3) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1 \times \mathcal{M}_3} (s_1', s_3')$, then depending on $e$ there are there are three possibilities:

   (i) if $e \in \Sigma_{\mathcal{M}_1} \cap \Sigma_{\mathcal{M}_3}$, then by construction, we have $s_1 \xrightarrow{e/\mathcal{A}'}_{\mathcal{M}_1} s_1'$ and $s_3 \xrightarrow{e/\mathcal{A}''}_{\mathcal{M}_3} s_3'$ with $\mathcal{A}' \cup \mathcal{A}'' = \mathcal{A}$. Since $\mathcal{B}((s_1, s_3), (s_2, s_4))$, we have $\mathcal{B}_{(1,2)}(s_1, s_2)$ and $\mathcal{B}_{(3,4)}(s_3, s_4)$. It follows that $s_2 \xrightarrow{e/\mathcal{A}'}_{\mathcal{M}_2} s_2'$ with $\mathcal{B}_{(1,2)}(s_1', s_2')$, and that $s_4 \xrightarrow{e/\mathcal{A}''}_{\mathcal{M}_4} s_4'$ with $\mathcal{B}_{(3,4)}(s_3', s_4')$. Finally, by construction, we have $(s_2, s_4) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_2 \times \mathcal{M}_4} (s_2', s_4')$, and by definition of $\mathcal{B}$, we have $\mathcal{B}((s_1', s_3'), (s_2', s_4'))$.

(ii) if $e \in \Sigma_{\mathcal{M}_1} \setminus \Sigma_{\mathcal{M}_3}$, then by construction, we have $s_1 \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1} s_1'$ and $s_3 = s_3'$. Since $\mathcal{B}((s_1, s_3), (s_2, s_4))$, we have $\mathcal{B}_{(1,2)}(s_1, s_2)$ and $\mathcal{B}_{(3,4)}(s_3, s_4)$. It follows that $s_2 \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_2} s_2'$ with $\mathcal{B}_{(1,2)}(s_1', s_2')$. Finally, by construction, we have $(s_2, s_4) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_2 \times \mathcal{M}_4} (s_2', s_4)$, and by definition of $\mathcal{B}$, we have $\mathcal{B}((s_1', s_3), (s_2', s_4))$.

(iii) if $e \in \Sigma_{\mathcal{M}_3} \setminus \Sigma_{\mathcal{M}_1}$, the proof is symmetrical to the previous case.

2. $\forall e \in \Sigma_{\mathcal{M}_2 \times \mathcal{M}_4}$, if $(s_2, s_4) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_2 \times \mathcal{M}_4} (s_2', s_4')$, we can prove that $(s_1, s_3) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1 \times \mathcal{M}_3} (s_1', s_3')$ symmetrically to the proof of (1).

Finally, we have $\mathcal{B}((s^0_{\mathcal{M}_1}, s^0_{\mathcal{M}_3}), (s^0_{\mathcal{M}_2}, s^0_{\mathcal{M}_4}))$ ■

The previous theorem allows us to use any distribution method developed for two execution sites to distribute the centralized system over more than two execution sites. The idea is as follows. Let us consider an AMM $\mathcal{M}$, a partition $\{A_1, A_2, ..., A_n\}$ of $A_{\mathcal{M}}$ and $n$ subsets $\Sigma_1, \Sigma_2, ..., \Sigma_n$ such that $\bigcup_{i \in \{1..n\}} \Sigma_i = \Sigma_{\mathcal{M}}$. First, we distribute $\mathcal{M}$ over $A_1$, $\Sigma_1$ on one execution site, and $\bigcup_{i \in \{2..n\}} A_i$, $\bigcup_{i \in \{2..n\}} \Sigma_i$ on the other, we will obtain two AMM's $\mathcal{M}_1$ and $\mathcal{M}_1'$. Then we can distribute $\mathcal{M}_1'$ over $A_1, \Sigma_2$ on one execution site and $\bigcup_{i \in \{3..n\}} A_i$, $\bigcup_{i \in \{3..n\}} \Sigma_i$ on the other. We can repeat this process until we are left with only two execution sites. Theorem 2.1 assures us that the synchronized product of all the $\mathcal{M}_i$ obtained along the way will be bisimilar to $\mathcal{M}$.

## 2.4 Towards a solution

As stated in the previous section, the distribution process is constrained by two things (1) the locations of the actions and (2) the locations of the events. In order to solve the general distribution problem, we chose to first study those constraints separately and then combine the developed methods to solve the general problem. Therefore, before studying the general distribution problem, we will concentrate on two subclasses of this problem. In the first one, we will only consider the constraints on the actions. In this action driven distribution problem, studied in chapter 3, we will make no assumption on the events. As a matter of fact, we will assume that every event can be monitored everywhere ($\Sigma_1 = \Sigma_2 = \Sigma$). The distribution will only depend on the location of the actions. Symmetrically, in the second problem, we will only consider the constraints on the events. In this event driven distribution problem, studied in chapter 4, we will only examine AMM's without actions ($A_{\mathcal{M}} = \emptyset$). Then, in chapter 5, we will combine and extend the methods developed in chapters 3 and 4 to solve the general distribution problem.

# Chapter 3

# Action driven distribution

In this chapter we will consider a particular class of the general distribution problem, in which the distribution is driven by the location of the actions to be performed. This chapter is organized as follows. We start by formulating the problem in section 3.1. We follow by showing how it can simply be solved in section 3.2. However we will see that this solution can be greatly improved. That is why, in section 3.3, in order to improve this solution, we show how we can transform the simple solution by merging some states together, while respecting the correctness of the distribution. We follow, in section 3.4, by discussing the optimality of a solution. Finally, in section 3.5, we conclude by presenting a heuristic approach to the problem of finding this optimal solution.

## 3.1 The problem

In the action driven distribution problem, we split the set of actions in two, just like in the general problem, but here we assume that all events can be monitored everywhere. Let us first formalize the problem.

**Problem 3.1 - Action driven distribution problem**
Given an AMM $\mathcal{M}$ and a partition $\{A_1, A_2\}$ of $A_{\mathcal{M}}$, does there exist two AMM's $\mathcal{M}_1$ and $\mathcal{M}_2$ with $A_{\mathcal{M}_1} = A_1$, $A_{\mathcal{M}_2} = A_2$, $\Sigma_{\mathcal{M}_1} = \Sigma_{\mathcal{M}_2} = \Sigma_{\mathcal{M}}$ such that $\mathcal{M}_1 \times \mathcal{M}_2 \equiv_B \mathcal{M}$.

## 3.2 Restriction

Since all events can be monitored everywhere, a simple way to solve the problem 3.1 is to duplicate the centralized system on each execution site and restrict the actions of the duplicates to respectively actions of $A_1$ and $A_2$. We formalize the notion of restriction as follows.

**Definition 3.1 - Restriction of AMM**

Given an AMM $\mathcal{M}$, the restriction of $\mathcal{M}$ to a subset $A$ of $A_{\mathcal{M}}$, noted $\rho(\mathcal{M}, A)$, is an AMM

$$(S_{\mathcal{M}}, s^0_{\mathcal{M}}, \Sigma_{\mathcal{M}}, A, \delta_{\mathcal{M}}, \lambda_{\rho(\mathcal{M},A)})$$

where $\lambda_{\rho(\mathcal{M},A)}$ is defined $\forall s \in S_{\mathcal{M}}, \forall e \in out_{\mathcal{M}}(s,e)$ as follows:

$$\lambda_{\rho(\mathcal{M},A)}(s,e) = \lambda_{\mathcal{M}}(s,e) \cap A$$

We can also adapt the definition of restriction to traces.

**Definition 3.2 - Restriction of a trace**

Given an AMM $\mathcal{M}$ and a trace $w$ of $\mathcal{M}$, the restriction $w$ to a subset $A$ of $A_{\mathcal{M}}$, noted $\rho(w, A)$ is defined recursively as follows:

1. $\rho(\epsilon, A) = \epsilon$
2. $\rho((e, \mathcal{A}) \cdot w', A) = (e, \mathcal{A} \cap A) \cdot \rho(w', A)$

The definition of restriction of traces can easily be extended to sets of traces (languages): $\rho(\mathcal{L}, A) = \{ \rho(w, A) \mid w \in \mathcal{L} \}$. Given these definitions, it is easy to see that $\rho(\mathcal{L}_{\mathcal{M}}, A) = \mathcal{L}_{\rho(\mathcal{M},A)}$.

Intuitively, it seems quite natural that given an AMM $\mathcal{M}$, the synchronized product of the respective restriction of $\mathcal{M}$ on $A_1$ and $A_2$ is bisimilar to $\mathcal{M}$. Indeed, apart from the actions, the restricted AMM are almost identical to $\mathcal{M}$. We prove this formally hereafter.

**Theorem 3.1**

Given an AMM $\mathcal{M}$ and a partition $\{A_1, A_2\}$ of $A_{\mathcal{M}}$, if $\mathcal{M}_1 = \rho(\mathcal{M}, A_1)$ and $\mathcal{M}_2 = \rho(\mathcal{M}, A_2)$, then $\mathcal{M}_1 \times \mathcal{M}_2 \equiv_B \mathcal{M}$.

**Proof**

We define a relation $\mathcal{B} \subseteq S_{\mathcal{M}_1 \times \mathcal{M}_2} \times S_{\mathcal{M}}$ such that $\mathcal{B}((s_1, s_2), s)$ if and only if $s = s_1 = s_2$, and we prove that $\mathcal{B}$ is an action bisimulation relation. Indeed, $\forall s \in S_{\mathcal{M}}, \forall (s_1, s_2) \in S_{\mathcal{M}_1 \times \mathcal{M}_2}$, if $\mathcal{B}((s_1, s_2), s)$ then:

1. $\forall e \in \Sigma_{\mathcal{M}}$, if $s \xrightarrow{e/\mathcal{A}}_{\mathcal{M}} s'$, by construction $s \xrightarrow{e/\mathcal{A} \cap A_1}_{\mathcal{M}_1} s'$ and $s \xrightarrow{e/\mathcal{A} \cap A_2}_{\mathcal{M}_2} s'$. Consequently, since $(\mathcal{A} \cap A_1) \cup (\mathcal{A} \cap A_2) = \mathcal{A}$, $(s, s) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1 \times \mathcal{M}_2} (s', s')$ by definition of $\times$ and $\mathcal{B}((s', s'), s')$ holds trivially.
2. $\forall e \in \Sigma_{\mathcal{M}_1 \times \mathcal{M}_2}$, if $(s_1, s_2) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1 \times \mathcal{M}_2} (s'_1, s'_2)$ then by definition of $\times$, $s_1 \xrightarrow{e/\mathcal{A}_1}_{\mathcal{M}_1} s'_1$ and $s_2 \xrightarrow{e/\mathcal{A}_2}_{\mathcal{M}_2} s'_2$ with $\mathcal{A}_1 \cup \mathcal{A}_2 = \mathcal{A}$. Then, since $s = s_1 = s_2$, $s \xrightarrow{e/\mathcal{A}}_{\mathcal{M}} s'$ with $s' = s'_1 = s'_2$ and $\mathcal{B}((s'_1, s'_2), s')$ holds trivially.

Finally, $\mathcal{B}((s^0_{\mathcal{M}_1}, s^0_{\mathcal{M}_1}), s^0_{\mathcal{M}})$ holds by construction and $\mathcal{M}_1 \times \mathcal{M}_2 \equiv_B \mathcal{M}$. ∎

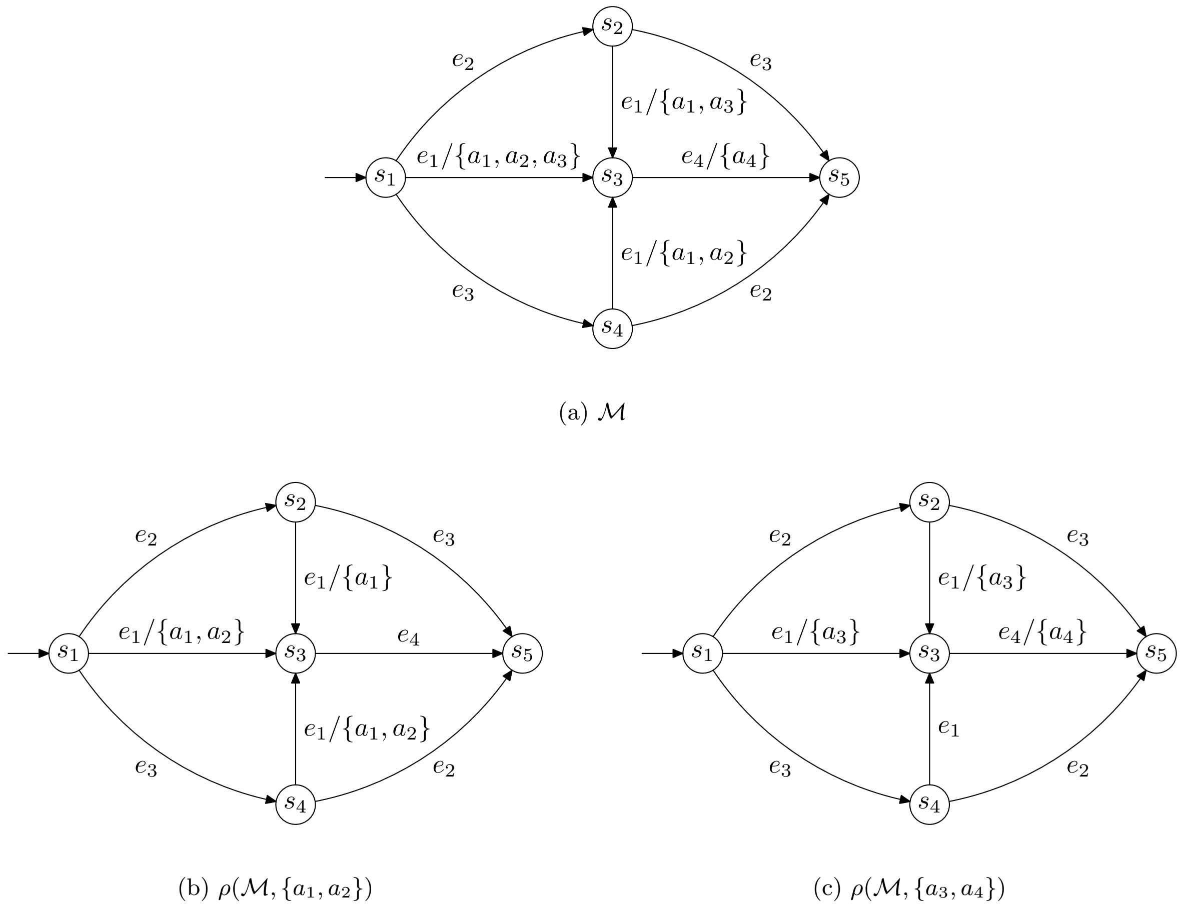



Figure 3.1: Distribution using restriction

Note that this theorem is adapted from [Meu02]. However, the proof has been greatly simplified. It formally proves that the restriction gives us a (trivial) solution to the general distribution problem.

**Example 3.1**
Figure 3.1 present the distribution process using restriction. The original centralized AMM is presented in figure 3.1(a). The set of action $A_{\mathcal{M}} = \{a_1, a_2, a_3, a_4\}$. The distribution is made by restricting $\mathcal{M}$ to respectively $\{a_1, a_2\}$ and $\{a_3, a_4\}$.

## 3.3 Merging

In the previous section, we have presented a solution to the action driven distribution problem. However, in this solution, both distributed AMM's have the same number of states as the initial centralized AMM. It is therefore reasonable to wonder about the existence of a smaller solution in terms of number of states. This is why we will attempt to merge some states of the distributed AMM. First, we formalize the notion of merging.

**Definition 3.3 - Merging**
Given an AMM $\mathcal{M}$ and a partition $P = \{S_1, S_2, ..., S_n\}$ of $S_\mathcal{M}$, the merging of $\mathcal{M}$ by $P$ noted $\mu(\mathcal{M}, P)$ is an AMM:

$$(P, [s^0_\mathcal{M}]_P, \Sigma_\mathcal{M}, A_\mathcal{M}, \delta_{\mu(\mathcal{M},P)}, \lambda_{\mu(\mathcal{M},P)})$$

where $\forall S_i \in P, \forall e \in \Sigma_\mathcal{M}$ as follows:

$$\delta_{\mu(\mathcal{M},P)}(S_i, e) = \begin{cases} S_j & \text{if } \bigcup_{s\in S_i | e \in out_\mathcal{M}(s)} [\delta_\mathcal{M}(s,e)]_P \text{ is a singleton } \{S_j\} \\ undefined & \text{otherwise} \end{cases}$$

$$\lambda_{\mu(\mathcal{M},P)}(S_i, e) = \begin{cases} \cup_{s\in S_i | e \in out_\mathcal{M}(s)} \lambda_\mathcal{M}(s,e) & \text{if } \bigcup_{s\in S_i | e \in out_\mathcal{M}(s)} \{[\delta_\mathcal{M}(s,e)]_P\} \text{ is a singleton } \{S_j\} \\ undefined & \text{otherwise} \end{cases}$$

We need to be careful before merging states together. Indeed, the model of AMM is deterministic by nature and we need to keep this determinism. If there exists two states $s, s'$ in the same subset of the partition $P$, such that $e$ is accepted in both, but for which the reached states are in different subsets, merging those states would not be safe because it would produce a non-deterministic automaton. That is why, in definition 3.3, $\delta_{\mu(\mathcal{M},P)}(S_i, e)$ and $\lambda_{\mu(\mathcal{M},P)}(S_i, e)$ are left undefined if more that one subset is reached from a state $s$ of $S_i$ on event $e$, in other words, if $\bigcup_{s\in S_i | e \in out_\mathcal{M}(s)} \{[\delta_\mathcal{M}(s,e)]_P\}$ is not a singleton. Before we can use this transformation to improve the solution given by the restriction, we need to characterize the partitions for which the merging is safe. Therefore, we introduce the notion of deterministic partition.

**Definition 3.4 - Deterministic partition**
Given an AMM $\mathcal{M}$, a partition $P = \{S_1, S_2, ..., S_n\}$ of $S_\mathcal{M}$ is deterministic with respect to $\mathcal{M}$ if $\forall S_i \in P, \forall e \in \Sigma, \forall s, s' \in S_i$:

$$(e \in out_\mathcal{M}(s) \wedge e \in out_\mathcal{M}(s')) \rightarrow ([\delta_\mathcal{M}(s,e)]_P = [\delta_\mathcal{M}(s',e)]_P)$$

Now, of course, it is essential that we keep the correctness of the solution. Indeed, by merging states together in the distributed (i.e. restricted) AMM's, we might introduce some new transitions in their synchronized product, which would violate the action bisimulation with the initial centralized AMM. Therefore, before merging states together, we need to check that this does not happen. We need a way to characterize under which conditions, states can be merged together while respecting the correctness of the distribution. This is given by the following theorem.

**Theorem 3.2**

Given an AMM $\mathcal{M}$, a partition $\{A_1, A_2\}$ of $A_{\mathcal{M}}$ and two partitions $P_1$ and $P_2$ of $S_{\mathcal{M}}$, both deterministic w.r.t $\mathcal{M}$, let $\mathcal{M}_1 = \mu(\rho(\mathcal{M}, A_1), P_1)$ and $\mathcal{M}_2 = \mu(\rho(\mathcal{M}, A_2), P_2)$, $\mathcal{M}_1 \times \mathcal{M}_2 \equiv_B \mathcal{M}$ if and only if for all reachable states $(S_1, S_2) \in S_{\mathcal{M}_1 \times \mathcal{M}_2}$, for all $s \in S_1 \cap S_2$,

(i) $out_{\mathcal{M}}(s) = out_{\mathcal{M}_1}(S_1) \cap out_{\mathcal{M}_2}(S_2)$

(ii) $\forall e \in out_{\mathcal{M}}(s)$, $\lambda_{\mathcal{M}}(s, e) = \lambda_{\mathcal{M}_1}(S_1, e) \cup \lambda_{\mathcal{M}_2}(S_2, e)$

**Proof**

($\rightarrow$) We know that $\mathcal{M}_1 \times \mathcal{M}_2 \equiv_B \mathcal{M}$. This implies that there exists an action bisimulation relation $\mathcal{B} \subseteq S_{\mathcal{M}_1 \times \mathcal{M}_2} \times S_{\mathcal{M}}$ such that $\mathcal{B}((S^0_{\mathcal{M}_1}, S^0_{\mathcal{M}_2}), s^0_{\mathcal{M}})$. We first prove that for all reachable states $(S_1, S_2)$ of $\mathcal{M}_1 \times \mathcal{M}_2$, there exists $s \in S_1 \cap S_2$ such that $\mathcal{B}((S_1, S_2)s)$

- For the initial state, $(S^0_{\mathcal{M}_1}, S^0_{\mathcal{M}_2}) = ([s^0_{\mathcal{M}}]_{P_1}, [s^0_{\mathcal{M}}]_{P_2})$. It follows that $s^0_{\mathcal{M}} \in S^0_{\mathcal{M}_1} \cap S^0_{\mathcal{M}_2}$ and by definition of $\mathcal{B}$, we have $\mathcal{B}((S^0_{\mathcal{M}_1}, S^0_{\mathcal{M}_2}), s^0_{\mathcal{M}})$.
- For any reachable state $(S_1, S_2)$ of $\mathcal{M}_1 \times \mathcal{M}_2$ such that there exists $s \in S_1 \cap S_2$ with $\mathcal{B}((S_1, S_2), s)$, if $(S_1, S_2) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1 \times \mathcal{M}_2} (S_1', S_2')$, since $\mathcal{B}((S_1, S_2), s)$, $s \xrightarrow{e/\mathcal{A}}_{\mathcal{M}} s'$ with $\mathcal{B}((S_1', S_2'), s')$. By construction, since $s \in S_1$, $s' \in S_1'$ and since $s \in S_2$, $s' \in S_2'$. It follows that $s' \in S_1' \cap S_2'$ with $\mathcal{B}((S_1', S_2'), s')$.

Then we prove that for all reachable states $s$ of $\mathcal{M}$, we have $\mathcal{B}(([s]_{P_1}, [s]_{P_2}), s)$:

- For the initial state $s^0_{\mathcal{M}}$, we have $\mathcal{B}(([s^0_{\mathcal{M}}]_{P_1}, [s^0_{\mathcal{M}}]_{P_2}), s^0_{\mathcal{M}})$ by definition of $\mathcal{B}$.
- For any reachable state $s$ of $\mathcal{M}$ such that $\mathcal{B}(([s]_{P_1}, [s]_{P_2}), s)$, if $s \xrightarrow{e/\mathcal{A}}_{\mathcal{M}} s'$ then by construction $[s]_{P_1} \xrightarrow{e/\mathcal{A} \cap A_1}_{\mathcal{M}_1} [s']_{P_1}$ and $[s]_{P_2} \xrightarrow{e/\mathcal{A} \cap A_2}_{\mathcal{M}_2} [s']_{P_2}$ since $P_1$ and $P_2$ are deterministic w.r.t. $\mathcal{M}$. Consequently, since $(\mathcal{A} \cap A_1) \cup (\mathcal{A} \cap A_2) = \mathcal{A}$, we have that $([s]_{P_1}, [s]_{P_2}) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1 \times \mathcal{M}_2} ([s']_{P_1}, [s']_{P_2})$ by definition of $\times$. And since $\mathcal{B}(([s]_{P_1}, [s]_{P_2}), s)$, we also have that for $s'$, $\mathcal{B}(([s']_{P_1}, [s']_{P_2}), s')$.

We can deduce that for every reachable states $(S_1, S_2) \in S_{\mathcal{M}_1 \times \mathcal{M}_2}$, for all state $s \in S_1 \cap S_2$, we have $\mathcal{B}((S_1, S_2), s)$. Then, it is easy to see that $\mathcal{B}((S_1, S_2), s)$ implies (i) and (ii).

($\leftarrow$) Let $\mathcal{B} \subseteq S_{\mathcal{M}_1 \times \mathcal{M}_2} \times S_{\mathcal{M}}$ such that $\mathcal{B}((S_1, S_2), s)$ if and only if $s \in S_1 \cap S_2$. We prove that $\mathcal{B}$ is an action bisimulation relation. Indeed, for all reachable states $(S_1, S_2) \in S_{\mathcal{M}_1 \times \mathcal{M}_2}$, for all $s \in S_{\mathcal{M}}$, if $\mathcal{B}((S_1, S_2), s)$ then:

1. $\forall e \in \Sigma_{\mathcal{M}}$, if $s \xrightarrow{e/\mathcal{A}}_{\mathcal{M}} s'$, then by construction $[s]_{P_1} \xrightarrow{e/\mathcal{A} \cap A_1}_{\mathcal{M}_1} [s']_{P_1}$ and $[s]_{P_2} \xrightarrow{e/\mathcal{A} \cap A_2}_{\mathcal{M}_2} [s']_{P_2}$ since $P_1$ and $P_2$ are deterministic w.r.t $\mathcal{M}$. Consequently, since $(\mathcal{A} \cap A_1) \cup (\mathcal{A} \cap A_2) = \mathcal{A}$, $([s]_{P_1}, [s]_{P_2}) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1 \times \mathcal{M}_2} ([s']_{P_1}, [s']_{P_2})$ by definition of $\times$ and $\mathcal{B}(([s']_{P_1}, [s']_{P_2}), s')$ holds trivially.

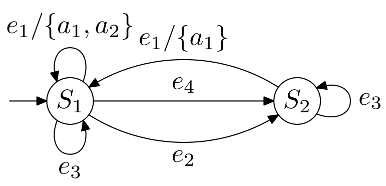
2. $\forall e \in \Sigma_{\mathcal{M}_1 \times \mathcal{M}_2}$, if $(S_1, S_2) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1 \times \mathcal{M}_2} (S'_1, S'_2)$ then by definition of times, $S_1 \xrightarrow{e/\mathcal{A}_1}_{\mathcal{M}_1} S'_1$ and $S_2 \xrightarrow{e/\mathcal{A}_2}_{\mathcal{M}_2} S'_2$ with $\mathcal{A}_1 \cup \mathcal{A}_2 = \mathcal{A}$. Then by (i) $s \xrightarrow{e/\mathcal{A}'}_{\mathcal{M}} s'$ and by (ii) $\mathcal{A}' = \mathcal{A}$. By construction, since $S_1 \xrightarrow{e/\mathcal{A}_1}_{\mathcal{M}_1} S'_1$ with $s \in S_1$, we have $s' \in S'_1$. Similarly $s' \in S'_2$ and $\mathcal{B}((S'_1, S'_2), s')$.

Finally by construction, we have $\mathcal{B}(([s^0_{\mathcal{M}}]_{P_1}, [s^0_{\mathcal{M}}]_{P_2}), s^0_{\mathcal{M}})$, thus $\mathcal{M}_1 \times \mathcal{M}_2 \equiv_B \mathcal{M}$. ■

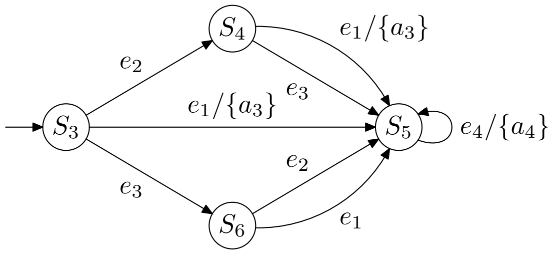
Theorem 3.2 gives a necessary and sufficient condition to check that, given two deterministic partitions, the merging of the restricted AMM's using those partitions keeps the correctness of the distribution. Note that we can check (i) and (ii) linearly in the number of states in $\mathcal{M}$. Indeed, from the proof above, we know that the set of reachable states of $\mathcal{M}_1 \times \mathcal{M}_2$ is given by $\{([s]_{P_1}, [s]_{P_2}) | s \in S_{\mathcal{M}}\}$.

**Example 3.2**

To illustrate how we can obtain a better solution by merging some states together, consider the AMM of example 3.1 and the distribution obtained using only restriction. Figure 3.2 presents a solution with less states but that is still correct. This solution is obtained by merging $s_1, s_3, s_4$ and $s_2, s_5$ in the first restricted AMM, and by merging $s_3, s_5$ in the second restricted AMM. The synchronized product of the distributed AMM is presented in figure 3.2(c). One can easily check this synchronized product is bisimilar to the original AMM of figure 3.1(a). Another way to check the correctness of this distribution, is by using theorem 3.2. For instance, for the initial state $(S_1, S_3)$, $\{s_1, s_3, s_4\} \cap \{s_1\} = \{s_1\}$. We have that $out_{\mathcal{M}}(s_1) = \{e_1, e_2, e_3\}$ and that $out_{\mathcal{M}_1}(S_1) \cap out_{\mathcal{M}_1}(S_3) = \{e_1, e_2, e_3, e_4\} \cap \{e_1, e_2, e_3\} = \{e_1, e_2, e_3\}$. For the actions, for $e_1$, we have $\lambda_{\mathcal{M}}(s_1, e_1) = \{a_1, a_2, a_3\}$ and $\lambda_{\mathcal{M}_1}(S_1, e_1) \cup \lambda_{\mathcal{M}_2}(S_3, e_1) = \{a_1, a_2\} \cup \{a_3\} = \{a_1, a_2, a_3\}$, and for $e_2, e_3$ there are no actions from $s_1$, $S_1$ and $S_3$.

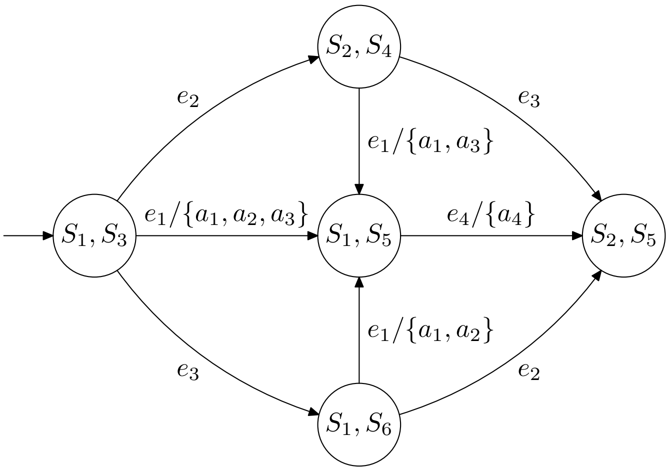



(a) $\mathcal{M}_1 = \mu(\rho(\mathcal{M}, \{a_1, a_2\}), \{S_1, S_2\})$, $S_1 = \{s_1, s_3, s_4\}, S_2 = \{s_2, s_5\}$




(b) $\mathcal{M}_2 = \mu(\rho(\mathcal{M}, \{a_3, a_4\}), \{S_3, S_4, S_5, S_6\})$, $S_3 = \{s_1\}, S_4 = \{s_2\}, S_5 = \{s_3, s_5\}, S_6 = \{s_4\}$

Figure 3.2: Example of a better solution




(c) $\mathcal{M}_1 \times \mathcal{M}_2$

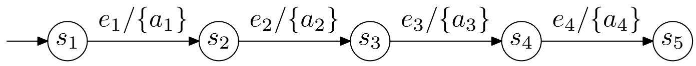
Figure 3.2: Example of better solution (cont'd)

## 3.4 Minimizing state space

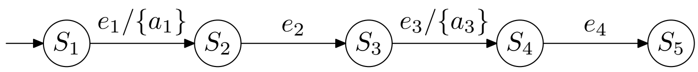
Now that we can characterize correct solutions, we can focus on what we mean exactly by minimizing the solution. An obvious criteria to classify the solution is the number of states in both distributed AMM's. We could simply try to find the solution which minimizes the total number of states $|S_{\mathcal{M}_1}| + |S_{\mathcal{M}_2}|$. However, as shown in example 3.3, this could lead to very odd solutions, where every states are merged in one of the restricted AMM and nothing is merged in the other. This kind of solution is of course not interesting. This is specially true if the amount of memory on the implementation platform is critical. Therefore, in order to balance the size of both distributed AMM's, we can try to find a solution that minimizes the number of states of both AMM's at the same time. One way to achieve this, is by minimizing $max(|S_{\mathcal{M}_1}|, |S_{\mathcal{M}_2}|)$. We formulate the problem as follows.

**Problem 3.2 - State minimal partitioning problem**
Given an AMM $\mathcal{M}$, we would like to build two partition $P_1$ and $P_2$ of $S_{\mathcal{M}}$ such that if $\mathcal{M}_1 = \mu(\rho(\mathcal{M}, A_1), P_1)$ and $\mathcal{M}_2 = \mu(\rho(\mathcal{M}, A_2), P_2)$, $\mathcal{M} \equiv_{\mathcal{B}} \mathcal{M}_1 \times \mathcal{M}_2$ and such that $max(|S_{\mathcal{M}_1}|, |S_{\mathcal{M}_2}|)$ is minimal.

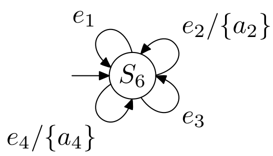
**Example 3.3**
Figure 3.3 shows an example of distribution when trying to minimize the total number of states $|S_{\mathcal{M}_1}| + |S_{\mathcal{M}_2}|$. The AMM's presented respectively in figure 3.3(a) and 3.3(b) indeed minimize the total number of states. However, nothing is merged in $\mathcal{M}_1$, while everything is merged in $\mathcal{M}_2$. Note that merging




(a) $\mathcal{M}$




(b) $\mathcal{M}_1 = \mu(\rho(\mathcal{M}, \{a_1, a_3\}), \{S_1, S_2, S_3, S_4, S_5\})$,$S_1 = \{s_1\}$, $S_2 = \{s_2\}, S_3 = \{s_3\}, S_4 = \{s_4\}, S_5 = \{s_5\}$




(c) $\mathcal{M}_2 = \mu(\rho(\mathcal{M}, \{a_2, a_4\}), \{S_6\})$, $S_6 = \{s_1, s_2, s_3, s_4, s_5\}$

Figure 3.3: A solution where everything is merged in $\mathcal{M}_1$

everything together in $\mathcal{M}_2$ is possible because this does not introduce non-determinism. But it might not necessarily be the case.

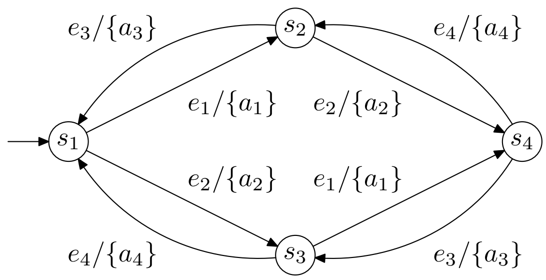
Now the question is, how do we find a solution to problem 3.2. We can take advantage of theorem 3.2. Since there is a finite number of states in $S_{\mathcal{M}}$, we can enumerate all partitions of $S_{\mathcal{M}}$. Let $P_{\mathcal{M}}$ be the set of all partitions of $S_{\mathcal{M}}$ deterministic w.r.t $\mathcal{M}$. We can exhaustively check for all $(P_1, P_2) \in P_{\mathcal{M}} \times P_{\mathcal{M}}$ that the conditions of theorem 3.2 are verified, as explained earlier, and keep the couple of partitionings $(P_1^+, P_2^+)$ such that $max(|P_1^+|, |P_2^+|)$ is minimal. The solution is then given by $\mathcal{M}_1 = \mu(\rho(\mathcal{M}, A_1), P_1^+)$ and $\mathcal{M}_2 = \mu(\rho(\mathcal{M}, A_2), P_2^+)$. Note that this method is exponential in the size of $S_{\mathcal{M}}$. The worst case complexity of the state minimal partitioning remains unknown.

## 3.5 Heuristic framework

We do not know, presently if there exists a polynomial method to solve problem 3.2. However, we conjecture that it is not the case. Thus, we have developed a heuristic framework to avoid an exhaustive search among all couples of partitions. It works as follows. We start with $\mathcal{M}_1 = \mu(\rho(\mathcal{M}, A_1), \{\{s\}|s \in S_{\mathcal{M}}\})$, and $\mathcal{M}_2 = \mu(\rho(\mathcal{M}, A_2), \{\{s\}|s \in S_{\mathcal{M}}\})$. In other words, we start with the restrictions of the initial machine to the respective sets of actions. In this case, since nothing is merged, theorem 3.1 assures us that the distribution is correct. One can easily check that, for this distribution, conditions of theorem 3.2 are verified. Then we examine the distributed machines to see if we can merge some states $S_i$ and $S_j$ of $S_{\mathcal{M}_1}$ ($S_{\mathcal{M}_2}$) together. This is the case if and only if (1) the merging would not introduce non-determinism in $\mathcal{M}_1$ ($\mathcal{M}_2$) and if (2) the conditions of theorem 3.2 would still hold after the merging. For (2), we could check the conditions for all reachable states in $\mathcal{M}_1 \times \mathcal{M}_2$, but this is not necessary. Indeed, not all the reachable states are affected by the merging of $S_i$ and $S_j$, and since the conditions were verified before the merging, we only need to check the conditions for the states that are affected by the merging $\{([s]_{S_{\mathcal{M}_1}}, [s]_{S_{\mathcal{M}_2}})|s \in S_i \cup S_j\}$. We then choose one of the possible candidates according to some criterion and repeat the process, until no candidates are left. At each step there are several mergings possible, and the order in which the mergings are made has an influence on the resulting solution. Indeed, merging two states together can render two previously mergeable (unmergeable) states unmergeable (mergeable).

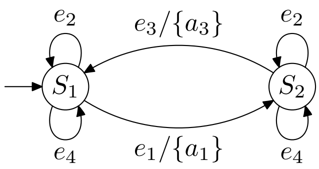
We thought at first that if at each step, instead of choosing one merging, we explored every possible mergings (i.e. backtracking), we would come up with the optimal solution. However, this is not always the case. Indeed, as shown in example 3.4, in some AMM's, even though $P_1^+$ and $P_2^+$ are deterministic, in all sequences of merging leading to this solution there is a merging leading to a non-deterministic partition. Therefore, in this case, the optimal solution cannot be found by backtracking on all the possible mergings. Therefore whichever criteria we use to chose a candidate, we will never find the optimal solution for all instances of the problem.

**Example 3.4**
Figure 3.4(a) is an example of AMM where the optimal solution can not be found by exploring all possible mergings at each step in the heuristic framework. Figures 3.4(b) and 3.4(c) present the optimal solution while figures 3.4(d) and 3.4(e) present the solution found exploring all possible mergings at each step. In this solution, $\max(|S_{\mathcal{M}_1}|, |S_{\mathcal{M}_2}|) = 3$, while in the optimal solution $\max(|S_{\mathcal{M}_1}|, |S_{\mathcal{M}_2}|) = 2$. It is the determinism that forbids us from finding the optimal solution. Indeed, in the first step of the heuristic framework, when selecting candidates in the restricted AMM's, we can not chose to merge neither $s_1$ with $s_3$ nor $s_2$ with $s_4$ as done in figure 3.4(b), and neither $s_1$ with $s_3$ nor $s_2$ with $s_1$ as done in figure 3.4(c), because it would lead to a non-deterministic partition. These candidates are therefore not taken into account. The mergings left lead us to the solution presented in figures 3.4(d) and 3.4(e)

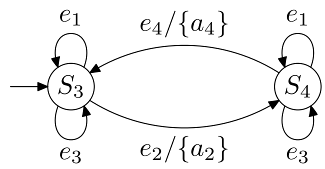



(a) $\mathcal{M}$

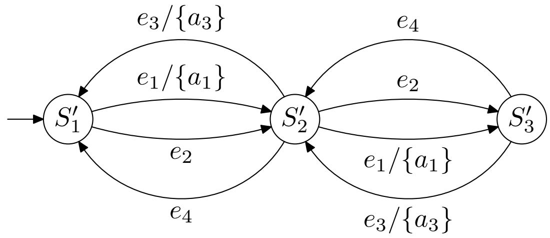
Figure 3.4: Determinism can cause problems - $A_1 = \{a_1, a_3\}$, $A_2 = \{a_2, a_4\}$



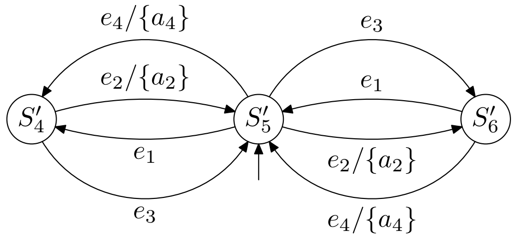

(b) $\mathcal{M}_1 = \mu(\rho(\mathcal{M}, \{a_1, a_3\}), \{S_1, S_2\})$,
$S_1 = \{s_1, s_3\}, S_2 = \{s_2, s_4\}$




(c) $\mathcal{M}_2 = \mu(\rho(\mathcal{M}, \{a_2, a_4\}), \{S_3, S_4\})$,
$S_3 = \{s_1, s_2\}, S_4 = \{s_3, s_4\}$




(d) $\mathcal{M}'_1 = \mu(\rho(\mathcal{M}, \{a_1, a_3\}), \{S'_1, S'_2, S'_3\})$,
$S'_1 = \{s_1\}, S'_2 = \{s_2, s_3\}, S'_3 = \{s_4\}$




(e) $\mathcal{M}'_2 = \mu(\rho(\mathcal{M}, \{a_2, a_4\}), \{S'_4, S'_5, S'_6\})$,
$S'_4 = \{s_2\}, S'_5 = \{s_1, s_4\}, S'_6 = \{s_3\}$

Figure 3.4: Determinism can cause problems - $A_1 = \{a_1, a_3\}$, $A_2 = \{a_2, a_4\}$ (cont'd)

In the heuristic framework we have presented, we still need a criterion to chose one merging among all the possible candidates. We have studied several criteria. The best we found is based on the impact a merging has on the conditions of theorem 3.2. If we merge two states $S_i, S_j$ with the same outgoing transitions, the resulting state will be identical to $S_i$ and $S_j$ w.r.t. the conditions of theorem 3.2. The conditions will be unaffected and the distribution will still be correct. In some sense, the difference between $S_i$ and $S_j$ in terms of outgoing transitions reflects the influence the merging will have on the rest of the distributed AMM's. Therefore, at each step, we chose to merge two states $S_i, S_j$ (of either $S_{\mathcal{M}_1}$ of $S_{\mathcal{M}_2}$) such that the difference between $S_i$ and $S_j$ in terms of outgoing transitions is minimal. A good approximation of this difference is given by $|(out_{\mathcal{M}_1}(S_i) \setminus out_{\mathcal{M}_1}(S_j)) \cup (out_{\mathcal{M}_1}(S_j) \setminus out_{\mathcal{M}_1}(S_i))|$ (if the merging is made in $\mathcal{M}_1$). Moreover, in an effort to balance the number of merging in both distributed AMM's, we keep count of the number of mergings made in each distributed AMM's. Then, and at each step, if there are mergings minimizing the difference in both distributed AMM's, we chose one in the AMM where the least number of mergings were made. Experimentally, on relatively small example, this criterion seems to give good results compared to the optimal solution. On bigger examples, we could not compare our result with the optimal solution, because the exhaustive search does not terminate in a reasonable amount of time. That is why we compared it to a method were the candidates are picked at random. Our heuristic is always better than the random pick. However, the gain over the random pick method is fairly small.

# Chapter 4

# Event driven distribution

In this chapter we will consider another particular case of the general distribution problem, in which the distribution is driven by the locations of the events to be monitored. Contrarily to the problem we examined in chapter 3, we will see that this problem does not always have a solution. This chapter is organized as follows. First, in section 4.1, we formulate the problem. Then, in section 4.2, we give some intuition on why some AMM's cannot be distributed over certain pairs of alphabets. We follow, in section 4.3, by presenting a solution to the problem adapted from [CMT99], and in section 4.4, we give a construction implementing this solution.

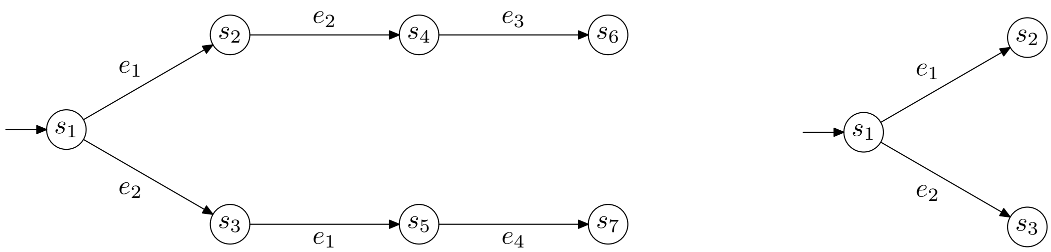
## 4.1 The problem

In the event driven distribution, we specify which events can be monitored on each execution sites just like in the general problem. However, in here, we will only consider AMM's without any actions. We will call these AMM's action free AMM's. For the sake of readability, we will simplify the notation and drop $A_{\mathcal{M}}$ and $\lambda_{\mathcal{M}}$ of the standard definition of AMM. We will also simplify the notations of traces and languages by dropping the actions. We can now formulate the problem.

**Problem 4.1 - Event driven distribution problem**
Given an action free AMM $\mathcal{M}$ and two subsets $\Sigma_1, \Sigma_2$ of $\Sigma_{\mathcal{M}}$ such that $\Sigma_1 \cup \Sigma_2 = \Sigma_{\mathcal{M}}$, does there exist two action free AMM's $\mathcal{M}_1$ and $\mathcal{M}_2$ with $\Sigma_{\mathcal{M}_1} = \Sigma_1$, $\Sigma_{\mathcal{M}_2} = \Sigma_2$ such that $\mathcal{M}_1 \times \mathcal{M}_2 \equiv_B \mathcal{M}$.

## 4.2 Distributability

As stated in the introduction of this chapter, contrarily to the action driven distribution problem, in the event driven distribution problem, there exist some AMM's that cannot be distributed, that is, for which there exists no synchronized products bisimilar to them. This problem arise because of the way the AMM's communicate: via synchronization on common events. This implies that for




(a) $\Sigma_1 = \{e_1, e_3, e_4\}$ and $\Sigma_2 = \{e_2, e_3, e_4\}$

(b) $\Sigma_1 = \{e_1\}$ and $\Sigma_2 = \{e_2\}$

Figure 4.1: Undistributable AMM's

one of the distributed AMM to allow or forbid the other to react to a certain event, it must observe that event. However, in the event driven distribution, the fact that an AMM can or cannot observe an event is fixed in the instance of the problem. Therefore for some instances of the problem, the centralized AMM cannot be distributed. In example 4.1, we present two typical cases where the centralized AMM cannot be distributed over $\Sigma_1$ and $\Sigma_2$, and try give some intuition on why those AMM cannot be distributed.

**Example 4.1**

Figures 4.1(a) and 4.1(b) present two action free AMM's that cannot be distributed, given $\Sigma_1, \Sigma_2$. In the first AMM of figure 4.1(a) if $\Sigma_1 = \{e_1, e_3, e_4\}$ and $\Sigma_2 = \{e_2, e_3, e_4\}$, after having treated events $e_1$ and $e_2$, both distributed AMM's should be able to decide weather to accept $e_3$ or $e_4$ depending on the order in which $e_1$ and $e_2$ occurred. However, they can only see $e_1$ and respectively $e_2$. Therefore, since they have no way to communicate their observations, they cannot take that decision. In the second AMM of figure 4.1(b), if $\Sigma_1 = \{e_1\}$ and $\Sigma_2 = \{e_2\}$, the problem comes from the fact that the distributed AMM's should forbid one another to accept their respective event when they have accepted their own. For instance, after $e_1$ has occurred, the first distributed AMM should be able to forbid $e_2$ from being accepted in the other distributed AMM. Again, this cannot be because of the way they communicate.

## 4.3 Partial event projection and product language

As explained in section 4.1, in the event driven distribution problem, we will only consider action free AMM's. Those AMM's can be viewed as traditional deterministic transition systems. A solution to the problem of distributing such transition systems modulo bisimulation has been studied in [CMT99]. In this section we will present this solution adapted to action free AMM's. We first give some definitions and then present the main results.

### 4.3.1 Partial event projection

Partial event projection is used to describe a trace of the centralized system as observed if only a subset of events can be monitored. We can formalize this as follows.

**Definition 4.1 - Partial event projection of a trace of an action free AMM**
The partial event projection of a trace $w$ of an action free AMM to a set of events $\Sigma$, noted $\pi(w, \Sigma)$ is defined recursively as follows:

1. $\pi(\epsilon, \Sigma) = \epsilon$
2. $\pi(e \cdot w', \Sigma) = \begin{cases} e \cdot \pi(w', \Sigma) & \text{if } e \in \Sigma \\ \pi(w', \Sigma) & \text{if } e \notin \Sigma \end{cases}$

Informally, $\pi(w, \Sigma)$ is the trace $w$ where all events not in $\Sigma$ are erased. This definition naturally extends to sets of traces (languages): $\pi(\mathcal{L}, \Sigma) = \{ \pi(w, \Sigma) \mid w \in \mathcal{L} \}$. Using partial event projection, the language accepted by the synchronized product of two action free AMM's can be formalized as follows.

**Lemma 4.1 - [CMT99]**
Given two action free AMM's $\mathcal{M}_1, \mathcal{M}_2$, $\mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2} = \{w \mid \pi(w, \Sigma_{\mathcal{M}_1}) \in \mathcal{L}_{\mathcal{M}_1} \wedge \pi(w, \Sigma_{\mathcal{M}_2}) \in \mathcal{L}_{\mathcal{M}_2}\}$ ■

Informally, the previous lemma states that, given two action free AMM's $\mathcal{M}_1, \mathcal{M}_2$, a trace $w$ is accepted by their synchronized product $\mathcal{M}_1 \times \mathcal{M}_2$ if and only if the partial event projection of $w$ on $\Sigma_{\mathcal{M}_1}$ is accepted by $\mathcal{M}_1$ and the partial event projection of $w$ on $\Sigma_{\mathcal{M}_2}$ is accepted by $\mathcal{M}_2$.

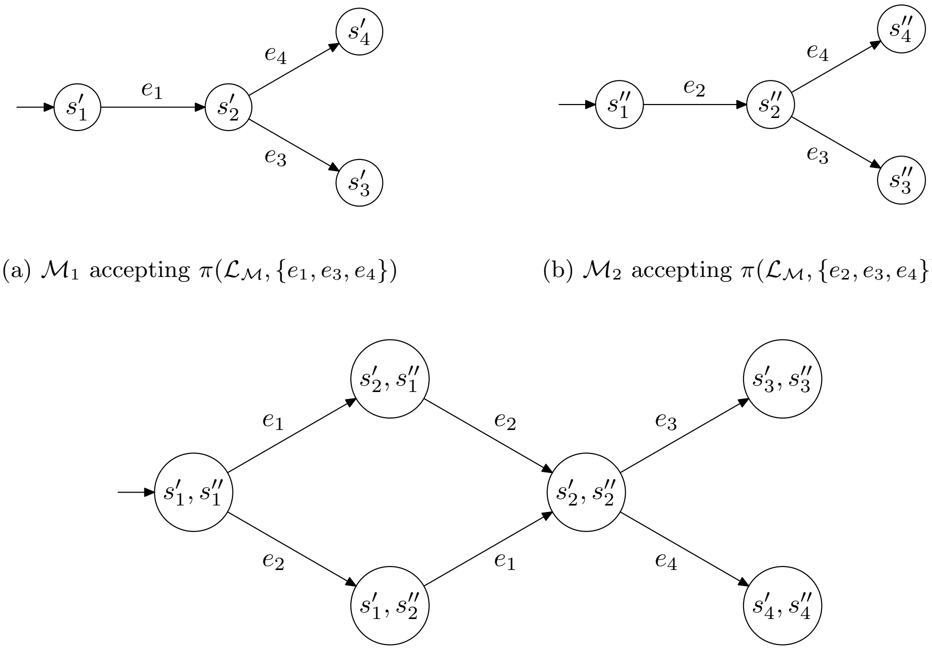
### 4.3.2 Product language

In order to characterize the action free AMM's that are bisimilar to a synchronized product of action free AMM's, we need to introduce product languages.

**Definition 4.2 - Product Language ([CMT99])**
Given an action free AMM $\mathcal{M}$, and two subsets $\Sigma_1, \Sigma_2$ of $\Sigma_{\mathcal{M}}$ such that $\Sigma_1 \cup \Sigma_2 = \Sigma_{\mathcal{M}}$, $\mathcal{L}_{\mathcal{M}}$ is a product language w.r.t $\Sigma_1$, $\Sigma_2$ if there exists two languages $\mathcal{L}_1 \subseteq \Sigma_1^*, \mathcal{L}_2 \subseteq \Sigma_2^*$ such that $\mathcal{L}_{\mathcal{M}} = \{ w \mid \pi(w, \Sigma_1) \in \mathcal{L}_1 \wedge \pi(w, \Sigma_2) \in \mathcal{L}_2 \}$

In order to check if the language accepted by an action free AMM is a product language, we can use the fact that a product language (and only a product language) is equal to the language accepted by the synchronized product of AMM's accepting its respective partial event projections on $\Sigma_1$ and $\Sigma_2$. This is formalized in the following lemma.

**Lemma 4.2 - [CMT99]**
Given an action free AMM $\mathcal{M}$, and two subsets $\Sigma_1, \Sigma_2$ of $\Sigma_{\mathcal{M}}$ such that $\Sigma_1 \cup \Sigma_2 = \Sigma_{\mathcal{M}}$, $\mathcal{L}_{\mathcal{M}}$ is a product language w.r.t $\Sigma_1, \Sigma_2$ if and only if $\mathcal{L}_{\mathcal{M}} = \{w \mid \pi(w, \Sigma_{\mathcal{M}_1}) \in \pi(\mathcal{L}_{\mathcal{M}}, \Sigma_1) \wedge \pi(w, \Sigma_{\mathcal{M}_2}) \in \pi(\mathcal{L}_{\mathcal{M}}, \Sigma_2)\}$■




(a) $\mathcal{M}_1$ accepting $\pi(\mathcal{L}_\mathcal{M}, \{e_1, e_3, e_4\})$

(b) $\mathcal{M}_2$ accepting $\pi(\mathcal{L}_\mathcal{M}, \{e_2, e_3, e_4\})$

(c) $\mathcal{M}_1 \times \mathcal{M}_2$

Figure 4.2: Partial event projection and product language at work

### 4.3.3 Checking for distributability

We now present the main result adapted from theorem 6.2 of [CMT99], providing us with a necessary and sufficient condition to check if an action free AMM can be distributed.

**Theorem 4.1 - [CMT99]**
Given an AMM $\mathcal{M}$ with $A_\mathcal{M} = \emptyset$, and two subsets $\Sigma_1, \Sigma_2$ of $\Sigma_\mathcal{M}$ such that $\Sigma_1 \cup \Sigma_2 = \Sigma_\mathcal{M}$, there exists two action free AMM's $\mathcal{M}_1, \mathcal{M}_2$ with $\Sigma_{\mathcal{M}_1} = \Sigma_1$ and $\Sigma_{\mathcal{M}_2} = \Sigma_1$ such that $\mathcal{M}_1 \times \mathcal{M}_2 \equiv_B \mathcal{M}$ if and only if $\mathcal{L}_\mathcal{M}$ is a product language w.r.t $\Sigma_1, \Sigma_2$ ■

As specified in [CMT99], theorem 4.1 and lemmata 4.1 and 4.2 yield an effective procedure for checking that an action free AMM $\mathcal{M}$ is distributable over $\Sigma_1$ and $\Sigma_2$. Indeed, we have to first construct two AMM's, $\mathcal{M}_1, \mathcal{M}_2$ accepting respectively $\pi(\mathcal{L}_\mathcal{M}, \Sigma_1)$ and $\pi(\mathcal{L}_\mathcal{M}, \Sigma_2)$, and then verify that $\mathcal{L}_\mathcal{M} = \mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2}$. If $\mathcal{L}_\mathcal{M} = \mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2}$, the solution is provided by $\mathcal{M}_1$ and $\mathcal{M}_2$. If, on the contrary, $\mathcal{L}_\mathcal{M} \neq \mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2}$, theorem 4.1 assures us that $\mathcal{M}$ is not distributable over $\Sigma_1$ and $\Sigma_2$.

**Example 4.2**
Let us consider the AMM $\mathcal{M}$ presented in figure 4.1(a). The language accepted by $\mathcal{M}$ is given by $\mathcal{L}_\mathcal{M} = \{\epsilon, e_1, e_1 \cdot e_2, e_2 \cdot e_1, e_1 \cdot e_2 \cdot e_3, e_2 \cdot e_1 \cdot e_4\}$. The respective partial event projection of $\mathcal{L}_\mathcal{M}$

on $\Sigma_1 = \{e_1, e_3, e_4\}$ and $\Sigma_2 = \{e_2, e_3, e_4\}$ are given by $\pi(\mathcal{L}_\mathcal{M}, \Sigma_1) = \{\epsilon, e_1, e_1 \cdot e_3, e_1 \cdot e_4\}$ and $\pi(\mathcal{L}_\mathcal{M}, \Sigma_2) = \{\epsilon, e_2, e_2 \cdot e_3, e_2 \cdot e_4\}$. Figures 4.2(a) and 4.2(b) present two AMM's $\mathcal{M}_1$ and $\mathcal{M}_2$ accepting those languages. Then, figure 4.2(c) presents their synchronized product $\mathcal{M}_1 \times \mathcal{M}_2$. We can clearly observe that $\mathcal{L}_\mathcal{M} \neq \mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2}$. For example, $e_1 \cdot e_2 \cdot e_4$ is in $\mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2}$ but not in $\mathcal{L}_\mathcal{M}$. Therefore, $\mathcal{L}_\mathcal{M}$ is not a product language by lemmata 4.1 and 4.2. It follows, by theorem 4.1 that $\mathcal{M}$ is not bisimilar to a synchronized product of AMM's on $\Sigma_1$ and $\Sigma_2$.

## 4.4 Computing the solution

As we have seen in the previous section, in order to solve problem 4.1, given an action free AMM and a subset $\Sigma$ of $\Sigma_\mathcal{M}$, we need a way to construct an AMM accepting the partial event projection of $\mathcal{L}_\mathcal{M}$ on $\Sigma$. A simple and natural idea is to replace all transitions $s \xrightarrow{e}_\mathcal{M} s'$ in $\mathcal{M}$ with $e \notin \Sigma$ by $\epsilon$-transitions $s \xrightarrow{\epsilon}_\mathcal{M} s'$, and then determinize the resulting AMM. This basic idea used in classical automata theory leads us to the following construction.

**Definition 4.3 - Partial event projection of action free AMM**
Given an action free AMM $\mathcal{M}$, and a subset $\Sigma$ of $\Sigma_\mathcal{M}$, the partial event projection of $\mathcal{M}$ to $\Sigma$, noted $\pi(\mathcal{M}, \Sigma)$ is defined by an action free AMM

$$(2^{S_\mathcal{M}}, \Sigma\text{-}closure(s^0_\mathcal{M}), \Sigma, \delta_{\pi(\mathcal{M},\Sigma)})$$

where $\forall S \in 2^{S_\mathcal{M}}, \forall e \in \Sigma$:

$$\Sigma\text{-}closure(s) = \{s' \in S_\mathcal{M} \mid \exists w \in (\Sigma_\mathcal{M} \setminus \Sigma)^*,\ s \xrightarrow{w}_\mathcal{M} s'\}$$

$$\delta_{\pi(\mathcal{M},\Sigma)}(S, e) = \begin{cases} \bigcup_{\{s \in S \mid e \in out_\mathcal{M}(s)\}} \Sigma\text{-}closure(\delta_\mathcal{M}(s, e)) & \text{if } \{s \in S \mid e \in out_\mathcal{M}(s)\} \neq \emptyset \\ undefined & \text{otherwise} \end{cases}$$

This construction is very similar to the one used for determinizing a non-deterministic finite automaton with $\epsilon$ transitions. In this case, instead of using $\epsilon\text{-}closure(s)$, we use $\Sigma\text{-}closure(s)$ which is the set of all the states that can be reached from $s$ by accepting only events not in $\Sigma$. Each state of the projected AMM is a set of states from the original AMM. We start with the initial state given by $\Sigma\text{-}closure(s^0_\mathcal{M})$. Then, from a given state $S$ in the projected AMM, we build a transition accepting an event $e$ if there is a state $s \in S$ in which $e$ is accepted in the original AMM. The reached state is then given by the set of all states reached from a state of $S$ by accepting $e$ and their $\Sigma\text{-}closure$'s. From this definition, the partial event projection of an AMM $\mathcal{M}$ on a subset $\Sigma$ of $\Sigma_\mathcal{M}$ should accept exactly the partial event projection of $\mathcal{L}_\mathcal{M}$ on $\Sigma$. We prove this hereafter.

**Theorem 4.2**
Given an action free AMM $\mathcal{M}$ and a subset $\Sigma$ of $\Sigma_\mathcal{M}$, $\mathcal{L}_{\pi(\mathcal{M},\Sigma)} = \pi(\mathcal{L}_\mathcal{M}, \Sigma)$.

**Proof**

The proof is divided in two parts. We first prove that $\mathcal{L}_{\pi(\mathcal{M},\Sigma)} \subseteq \pi(\mathcal{L}_{\mathcal{M}},\Sigma)$, or in other words that $\forall w \in \mathcal{L}_{\pi(\mathcal{M},\Sigma)}, \exists w' \in \mathcal{L}_{\mathcal{M}}$ such that $\pi(w',\Sigma) = w$. We prove by induction on $|w|$ that $\forall S \in S_{\pi(\mathcal{M},\Sigma)}$ if $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{w}_{\pi(\mathcal{M},\Sigma)} S$, then $\forall s \in S, \exists w' \in \mathcal{L}_{\mathcal{M}}$ such that $s^0_{\mathcal{M}} \xrightarrow{w'}_{\mathcal{M}} s$ with $\pi(w',\Sigma) = w$.

- For the base case, $|w| = 0$, and $w = \epsilon$. By construction, $S^0_{\pi(\mathcal{M},\Sigma)} = \Sigma\text{-}closure(s^0_{\mathcal{M}})$, so $\forall s \in S^0_{\pi(\mathcal{M},\Sigma)}, \exists w' \in \mathcal{L}_{\mathcal{M}}$ such that $s^0_{\mathcal{M}} \xrightarrow{w'}_{\mathcal{M}} s$. Since $w' \in (\Sigma_{\mathcal{M}} \setminus \Sigma)^*$, we have $\pi(w',\Sigma) = \epsilon = w$.
- For the inductive step, we assume as inductive hypothesis that with $|w| = n$, we have that if $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{w}_{\pi(\mathcal{M},\Sigma)} S$ then $\forall s \in S, \exists w' \in \mathcal{L}_{\mathcal{M}}$ such that $s^0_{\mathcal{M}} \xrightarrow{w'}_{\mathcal{M}} s$ with $\pi(w',\Sigma) = w$. Now, if $|w| = n+1$, we can write $w = u \cdot e$. So we have $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{u}_{\pi(\mathcal{M},\Sigma)} S \xrightarrow{e}_{\pi(\mathcal{M},\Sigma)} S'$. By construction, $\forall s' \in S', \exists s'' \in S', \exists s \in S$ such that $s \xrightarrow{e}_{\mathcal{M}} s'' \xrightarrow{w'}_{\mathcal{M}} s'$ with $w' \in (\Sigma_{\mathcal{M}} \setminus \Sigma)^*$. We can conclude that $\forall s' \in S', \exists s'' \in S', \exists s \in S$ such that $s^0_{\mathcal{M}} \xrightarrow{u'}_{\mathcal{M}} s \xrightarrow{e}_{\mathcal{M}} s'' \xrightarrow{w'}_{\mathcal{M}} s'$ with $\pi(u' \cdot e \cdot w',\Sigma) = \pi(u',\Sigma) \cdot \pi(e,\Sigma) \cdot \pi(w',\Sigma)$. Since $e \in \Sigma$ and $w' \in (\Sigma_{\mathcal{M}} \setminus \Sigma)^*$, we have that $\pi(e,\Sigma) = e$ and $\pi(w',\Sigma) = \epsilon$. It follows directly that $\pi(u' \cdot e \cdot w',\Sigma) = u \cdot e = w$.

Then, we prove that $\pi(\mathcal{L}_{\mathcal{M}},\Sigma) \subseteq \mathcal{L}_{\pi(\mathcal{M},\Sigma)}$, or in other words that $\forall w \in \mathcal{L}_{\mathcal{M}}, \pi(w,\Sigma) \in \mathcal{L}_{\pi(\mathcal{M},\Sigma)}$. We prove by induction on $|w|$, that if $s^0_{\mathcal{M}} \xrightarrow{w}_{\mathcal{M}} s$ then $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{\pi(w,\Sigma)}_{\pi(\mathcal{M},\Sigma)} S$ with $s \in S$.

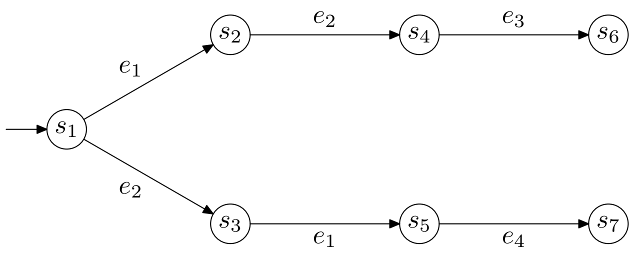
- For the base case, $|w| = 0$, and $w = \epsilon$. We have $\pi(w,\Sigma) = \epsilon$ accepted in $s^0_{\mathcal{M}}$. By construction $\epsilon$ accepted in $S^0_{\pi(\mathcal{M},\Sigma)}$ and that $s^0_{\mathcal{M}} \in S^0_{\pi(\mathcal{M},\Sigma)} = \Sigma\text{-}closure(s^0_{\mathcal{M}})$.
- For the induction step, we assume as inductive hypothesis that with $|w| = n$, if $s^0_{\mathcal{M}} \xrightarrow{w}_{\mathcal{M}} s$, $\exists S \in S_{\pi(\mathcal{M},\Sigma)}$ such that $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{\pi(w,\Sigma)} S$ and $s \in S$. Now, if $|w| = n+1$, we can write $w = u \cdot e$. So we have $s^0_{\mathcal{M}} \xrightarrow{u}_{\mathcal{M}} s \xrightarrow{e}_{\mathcal{M}} s'$. By inductive hypothesis, $\exists S \in S_{\pi(\mathcal{M},\Sigma)}$ such that $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{\pi(u,\Sigma)}_{\pi(\mathcal{M},\Sigma)} S$ and $s \in S$. If $e \notin \Sigma$, we have $\pi(u \cdot e,\Sigma) = \pi(u,\Sigma)$ and $s' \in \Sigma\text{-}closure(s) \subseteq S$. If $e \in \Sigma$, then by construction, $S \xrightarrow{e}_{\pi(\mathcal{M},\Sigma)} S'$. So we have that $\pi(u \cdot e,\Sigma) = \pi(u,\Sigma) \cdot e$ and by construction $s' \in S'$ with $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{\pi(u,\Sigma)}_{\pi(\mathcal{M},\Sigma)} S \xrightarrow{e}_{\pi(\mathcal{M},\Sigma)} S'$.■

**Corollary 4.1**

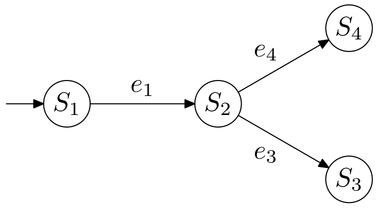
Given an action free AMM $\mathcal{M}$ and two subsets $\Sigma_1, \Sigma_2$ of $\Sigma_{\mathcal{M}}$ such that $\Sigma_1 \cup \Sigma_2 = \Sigma_{\mathcal{M}}$, if $\exists \mathcal{M}_1, \mathcal{M}_2$ with $\Sigma_{\mathcal{M}_1} = \Sigma_1$ and $\Sigma_{\mathcal{M}_2} = \Sigma_2$ such that $\mathcal{M}_1 \times \mathcal{M}_2 \equiv_B \mathcal{M}$, then $\pi(\mathcal{M},\Sigma_1) \times \pi(\mathcal{M},\Sigma_2) \equiv_B \mathcal{M}$.

**Proof**

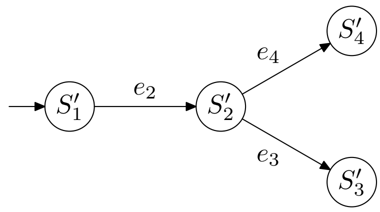
This corollary is a direct consequence of theorems 4.1, 4.2 and lemmata 4.1, 4.2. ■




(a) $\mathcal{M}$




(b) $\pi(\mathcal{M}, \{e_1, e_3, e_4\})$, $S_1 = \{s_1, s_3\}$, $S_2 = \{s_2, s_4, s_5\}, S_3 = \{s_6\}, S_4 = \{s_7\}$




(c) $\pi(\mathcal{M}, \{e_2, e_3, e_4\})$, $S_1 = \{s_1, s_2\}$, $S_2 = \{s_3, s_4, s_5\}, S_3 = \{s_6\}, S_4 = \{s_7\}$

**Example 4.3**

To illustrate the partial event projection of an action free AMM, let us take back the AMM from example 4.1. We recall this AMM $\mathcal{M}$ in figure 4.3(a). Figures 4.3(b) and 4.3(c) present the partial event projection of $\mathcal{M}$ on respectively $\{e_1, e_3, e_4\}$ and $\{e_2, e_3, e_4\}$. We can observe that those two AMM's accept the partial event projection of $\mathcal{L}_\mathcal{M}$ on respectively $\{e_1, e_3, e_4\}$ and $\{e_2, e_3, e_4\}$. Indeed, they are bisimilar (event isomorphic in this case) to the AMM's we presented in figures 4.2(a) and 4.2(b).

# Chapter 5

# Back to general distribution

In chapters 3 we studied a first subclass of the general distribution problem driven only by the location of the actions, and saw how restriction allowed us to solve that problem. In the chapter 4, we studied a second subclass of the general distribution problem driven only by the location of the events, and saw how partial event projection allowed us to solve the problem. In this chapter, we explain how those methods (i.e. restriction and partial event projection) can be adapted and combined to solve the general distribution problem. This chapter is organized as follows. First, in section 5.1, we extend the partial event projection to AMM's with actions. Then, in section 5.2, we explain how this extended partial event projection can be combined with the restriction to give a solution to the general distribution problem we presented in chapter 2. Finally, in section 5.3, we discuss our method.

## 5.1 Partial event projection with actions

We saw in chapter 4 that in order to solve the event driven distribution problem, we had to use partial event projection. However, in this problem, only action free AMM's were considered. In the general distribution problem, this is not the case anymore. Therefore, if we want to use this to solve the general distribution problem, we need to extend partial event projection to AMM's with actions. We formalize this hereafter.

**Definition 5.1 - Partial event projection of a trace**
The partial event projection of a trace $w$ of an AMM to a set of events $\Sigma$, noted $\pi(w, \Sigma)$ is defined recursively as follows:

1. $\pi(\epsilon, \Sigma) = \epsilon$
2. $\pi((e, \mathcal{A}) \cdot w', \Sigma) = \begin{cases} (e, \mathcal{A}) \cdot \pi(w', \Sigma) & \text{if } e \in \Sigma \\ \pi(w', \Sigma) & \text{if } e \notin \Sigma \end{cases}$

The only difference between this definition and the one presented in chapter 4 concerns the actions involved in the trace. These actions are simply erased along with the events not in $\Sigma$.

Later in chapter 4, we presented a construction which, given an action free AMM $\mathcal{M}$, allowed to build an action free AMM accepting the partial event projection $\mathcal{L}_{\mathcal{M}}$ to a subset $\Sigma$ of $\Sigma_{\mathcal{M}}$. Again we need to extend this construction in order to use it for the general distribution problem. We formalize this hereafter.

**Definition 5.2 - Partial event projection of AMM**

Given an AMM $\mathcal{M}$ and a subset $\Sigma$ of $\Sigma_{\mathcal{M}}$, the partial event projection of $\mathcal{M}$ to $\Sigma$, noted $\pi(\mathcal{M}, \Sigma)$ is defined by an AMM

$$(2^{S_{\mathcal{M}}}, \Sigma\text{-}closure(s^0_{\mathcal{M}}), \Sigma, A_{\mathcal{M}}, \delta_{\pi(\mathcal{M},\Sigma)}, \lambda_{\pi(\mathcal{M},\Sigma)})$$

where $\forall S \in 2^{S_{\mathcal{M}}}, \forall e \in \Sigma$:

$$\Sigma\text{-}closure(s) = \{s' \in S_{\mathcal{M}} \mid \exists w \in ((\Sigma_{\mathcal{M}} \setminus \Sigma) \times 2^{A_{\mathcal{M}}})^*,\ s \xrightarrow{w}_{\mathcal{M}} s'\}$$

$$\delta_{\pi(\mathcal{M},\Sigma)}(S, e) = \begin{cases} \bigcup_{\{s \in S \ \mid \ e \in out_{\mathcal{M}}(s)\}} \Sigma\text{-}closure(\delta_{\mathcal{M}}(s, e)) & \text{if } \{s \in S \mid e \in out_{\mathcal{M}}(s)\} \neq \emptyset \\ undefined & \text{otherwise} \end{cases}$$

$$\lambda_{\pi(\mathcal{M},\Sigma)}(S, e) = \begin{cases} \bigcup_{\{s \in S \ \mid \ e \in out_{\mathcal{M}}(s)\}} \lambda(s, e) & \text{if } \{s \in S \mid e \in out_{\mathcal{M}}(s)\} \neq \emptyset \\ undefined & \text{otherwise} \end{cases}$$

This definition is almost identical to the one in the action free case. Each state of the projected AMM's is a set of state of the original AMM, and a transition of the projected AMM represents several transitions of the original AMM. The difference here lies in the added $\lambda_{\pi(\mathcal{M},\Sigma)}$. Given a state $S$ of the projected AMM, and en event $e$ of $\Sigma$, we define $\lambda_{\pi(\mathcal{M},\Sigma)}(S, e)$ as the union of all the actions that are triggered by $e$ from a state $s$ of $S$ in the original AMM only if $\delta_{\pi(\mathcal{M},\Sigma)}(S, e)$ is defined and we leave it undefined otherwise.

In chapter 4, we proved that in the action free case $\mathcal{L}_{\pi(\mathcal{M},\Sigma)} = \pi(\mathcal{L}_{\mathcal{M}}, \Sigma)$. However, in the present case, it is not always true. The problem does not come from the construction we presented, but from the sub-alphabet $\Sigma$ onto which $\mathcal{M}$ is projected. Indeed, as illustrated in example 5.1, for some AMM's, the partial event projection of $\mathcal{L}_{\mathcal{M}}$ on some sub-alphabets $\Sigma$ of $\Sigma_{\mathcal{M}}$ cannot be accepted by any AMM at all.

**Example 5.1**

Consider the AMM presented in figure 5.1. If we denote this AMM $\mathcal{M}$, the language of $\mathcal{M}$ is given by $\mathcal{L}_{\mathcal{M}} = \{\epsilon, (e_1, \{a_1\}), (e_1, \{a_1\}) \cdot (e_2, \{a'_2\}), (e_2, \{a_2\})\}$. If $\Sigma = \{e_2\}$, the partial event projection of $\mathcal{L}_{\mathcal{M}}$ onto $\Sigma$ is given by $\pi(\mathcal{L}_{\mathcal{M}}, \Sigma) = \{\epsilon, (e_2, \{a'_2\}), (e_2, \{a_2\})\}$. This language cannot be accepted by an AMM because of the determinism of the model. Indeed, suppose that there exists an AMM $\mathcal{M}'$ accepting $\pi(\mathcal{L}_{\mathcal{M}}, \Sigma)$. From the initial state of $\mathcal{M}'$, $e_2$ should be accepted. If the set actions produced is $\{a'_2\}$, then $(e_2, \{a_2\}) \notin \mathcal{L}_{\mathcal{M}'}$. On the other hand if the set actions produced on $e_2$ is $\{a_2\}$, $(e_2, \{a''_2\}) \notin \mathcal{L}_{\mathcal{M}'}$. Therefore, no AMM can accept both $(e_2, \{a'_2\})$ and $(e_2, \{a_2\})$ at the same time.

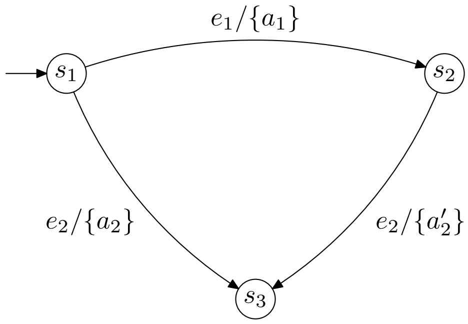



Figure 5.1: Problem with partial event projection, $\Sigma = \{e_2\}$

The problem illustrated in example 5.1 arise if there exists two words $w_1$ and $w_2$ of $\mathcal{L}_\mathcal{M}$ observed in the same manner when only events of $\Sigma$ are monitored, and if those two words can be prolonged by an event $e$ of $\Sigma$ such that the actions triggered by the $e$ after $w_1$ and after $w_2$ are different. We can formalize this as follows.

**Definition 5.3 - Ambiguous alphabet**
Given an AMM $\mathcal{M}$ and a subset $\Sigma$ of $\Sigma_\mathcal{M}$, we say that $\Sigma$ is ambiguous w.r.t $\mathcal{L}_\mathcal{M}$ if $\exists w_1, w_2 \in \mathcal{L}_\mathcal{M}, e \in \Sigma$ such that $\pi(w_1, \Sigma) = \pi(w_2, \Sigma)$, and such that $w_1 \cdot (e, \mathcal{A}_1) \in \mathcal{L}_\mathcal{M}$, $w_2 \cdot (e, \mathcal{A}_2) \in \mathcal{L}_\mathcal{M}$ with $\mathcal{A}_1 \neq \mathcal{A}_2$.

We now prove that if $\Sigma$ is ambiguous w.r.t. $\mathcal{L}_\mathcal{M}$, then there exists no AMM accepting $\pi(\mathcal{L}_\mathcal{M}, \Sigma)$

**Theorem 5.1**
Given an AMM $\mathcal{M}$ and a subset $\Sigma$ of $\Sigma_\mathcal{M}$, if $\Sigma$ is ambiguous w.r.t $\mathcal{L}_\mathcal{M}$, then there exists no AMM accepting $\pi(\mathcal{L}_\mathcal{M}, \Sigma)$.

**Proof**
If $\exists w_1, w_2 \in \mathcal{L}_\mathcal{M}, e \in \Sigma$ such that $\pi(w_1, \Sigma) = \pi(w_2, \Sigma)$, $w_1 \cdot (e, \mathcal{A}_1) \in \mathcal{L}_\mathcal{M}$ and $w_2 \cdot (e, \mathcal{A}_2) \in \mathcal{L}_\mathcal{M}$ with $\mathcal{A}_1 \neq \mathcal{A}_2$, then $\pi(w_1, \Sigma) \cdot (e, \mathcal{A}_1) \in \pi(\mathcal{L}_\mathcal{M}, \Sigma)$ and $\pi(w_2, \Sigma) \cdot (e, \mathcal{A}_2) \in \pi(\mathcal{L}_\mathcal{M}, \Sigma)$. Let us assume an AMM $\mathcal{M}'$ accepting $\pi(\mathcal{L}_\mathcal{M}, \Sigma)$. Since AMM are deterministic by definition, the state reached in $\mathcal{M}'$ after $\pi(w_1, \Sigma)$ is the same as the one reached after $\pi(w_2, \Sigma)$. Let this state be $s$. If $\lambda_{\mathcal{M}'}(s, e) = \mathcal{A}_1$ then $\pi(w_2, \Sigma) \cdot (e, \mathcal{A}_2) \notin \pi(\mathcal{L}_\mathcal{M}, \Sigma)$ and $\mathcal{L}_{\mathcal{M}'} \neq \pi(\mathcal{L}_\mathcal{M}, \Sigma)$. If, on the other hand, $\lambda_{\mathcal{M}'}(s, e) = \mathcal{A}_2$ then $\pi(w_1, \Sigma) \cdot (e, \mathcal{A}_1) \notin \pi(\mathcal{L}_\mathcal{M}, \Sigma)$ and $\mathcal{L}_{\mathcal{M}'} \neq \pi(\mathcal{L}_\mathcal{M}, \Sigma)$. If $\mathcal{M}'$ accepts $w_1 \cdot (e, \mathcal{A}_1)$, it cannot accept $w_2 \cdot (e, \mathcal{A}_2)$, and if $\mathcal{M}'$ accepts $w_2 \cdot (e, \mathcal{A}_2)$, it cannot accept $w_1 \cdot (e, \mathcal{A}_1)$. Since both $w_1 \cdot (e, \mathcal{A}_1)$ and $w_2 \cdot (e, \mathcal{A}_2)$ are in $\pi(\mathcal{L}_\mathcal{M}, \Sigma)$, there exists no AMM accepting $\pi(\mathcal{L}_\mathcal{M}, \Sigma)$. ■

In example 5.1, we have $w_1 = \epsilon$, $w_2 = (e_1, \{a_1\})$ with $\pi(\epsilon, \Sigma) = \pi((e_1, \{a_1\}), \Sigma) = \epsilon$ and $(e_2, \{a_2\}) \in \mathcal{L}_\mathcal{M}$, $(e_1, \{a_1\}) \cdot (e_2, \{a_2'\}) \in \mathcal{L}_\mathcal{M}$, with $\{a_2\} \neq \{a_1\}$. So we have $\Sigma$ ambiguous w.r.t. $\mathcal{L}_\mathcal{M}$ and by theorem 5.1, there exists no AMM accepting $\pi(\mathcal{L}_\mathcal{M}, \{e_2\})$.

**Corollary 5.1**

Given an AMM $\mathcal{M}$ and a subset $\Sigma$ of $\Sigma_{\mathcal{M}}$, if $\Sigma$ is ambiguous w.r.t $\mathcal{L}_{\mathcal{M}}$, then there exist no AMM accepting any superset $\mathcal{L}$ of $\pi(\mathcal{L}_{\mathcal{M}}, \Sigma)$.

**Proof**

If $\Sigma$ is ambiguous w.r.t $\mathcal{L}_{\mathcal{M}}$, from the proof of theorem 5.1, we can see that the problem is caused by $\pi(w_1, \Sigma) \cdot (e, \mathcal{A}_1)$ and $\pi(w_2, \Sigma) \cdot (e, \mathcal{A}_2)$ of $\pi(\mathcal{L}_{\mathcal{M}}, \Sigma)$ that cannot be accepted by the same AMM. However, every superset $\mathcal{L}$ of $\mathcal{L}_{\mathcal{M}}$ contains those two problematic words as well. Therefore, for any superset $\mathcal{L}$ of $\mathcal{L}_{\mathcal{M}}$, there exist no AMM accepting $\mathcal{L}$. ■

Theorem 5.1 characterizes the problematic cases. However, we still need to prove that if $\Sigma$ is not ambiguous w.r.t $\mathcal{L}_{\mathcal{M}}$, then our extended partial event projection construction allows us to build an AMM accepting the partial event projection of $\mathcal{L}_{\mathcal{M}}$ onto $\Sigma$. This is formalized in the following theorem.

**Theorem 5.2**

Given an AMM $\mathcal{M}$ and a subset $\Sigma$ of $\Sigma_{\mathcal{M}}$, if $\Sigma$ is not ambiguous w.r.t $\mathcal{L}_{\mathcal{M}}$ then $\mathcal{L}_{\pi(\mathcal{M},\Sigma)} = \pi(\mathcal{L}_{\mathcal{M}}, \Sigma)$.

**Proof**

The proof is divided in two parts. We first prove that if $\Sigma$ is not ambiguous w.r.t $\mathcal{L}_{\mathcal{M}}$ then $\mathcal{L}_{\pi(\mathcal{M},\Sigma)} \subseteq \pi(\mathcal{L}_{\mathcal{M}}, \Sigma)$, or in other words that $\forall w \in \mathcal{L}_{\pi(\mathcal{M},\Sigma)}, \exists w' \in \mathcal{L}_{\mathcal{M}}$ such that $\pi(w', \Sigma) = w$. We prove by induction on $|w|$ that $\forall S \in S_{\pi(\mathcal{M},\Sigma)}$ if $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{w}_{\pi(\mathcal{M},\Sigma)} S$, then $\forall s \in S, \exists w' \in \mathcal{L}_{\mathcal{M}}$ such that $s^0_{\mathcal{M}} \xrightarrow{w'}_{\mathcal{M}} s$ with $\pi(w', \Sigma) = w$.

- For the base case, $|w| = 0$, and $w = \epsilon$. By construction, $S^0_{\pi(\mathcal{M},\Sigma)} = \Sigma\text{-}closure(s^0_{\mathcal{M}})$, so $\forall s \in S^0_{\pi(\mathcal{M},\Sigma)}, \exists w' \in \mathcal{L}_{\mathcal{M}}$ such that $s^0_{\mathcal{M}} \xrightarrow{w'}_{\mathcal{M}} s$. Since $w' \in ((\Sigma_{\mathcal{M}} \setminus \Sigma) \times 2^{A_{\mathcal{M}}})^*$, we have $\pi(w', \Sigma) = \epsilon = w$.

- For the inductive step, we assume as inductive hypothesis that with $|w| = n$, we have that if $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{w}_{\pi(\mathcal{M},\Sigma)} S$ then $\forall s \in S, \exists w' \in \mathcal{L}_{\mathcal{M}}$ such that $s^0_{\mathcal{M}} \xrightarrow{w'}_{\mathcal{M}} s$ with $\pi(w', \Sigma) = w$. Now, if $|w| = n + 1$, we can write $w = u \cdot (e, \mathcal{A})$. So we have $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{u}_{\pi(\mathcal{M},\Sigma)} S \xrightarrow{e/\mathcal{A}}_{\pi(\mathcal{M},\Sigma)} S'$. By inductive hypothesis, $\forall s \in S, \exists u' \in \mathcal{L}_{\mathcal{M}}$, such that $s^0_{\mathcal{M}} \xrightarrow{u'}_{\mathcal{M}} s$ with $\pi(u', \Sigma) = u$. In other words, all states $s \in S$ can be reached from $s^0_{\mathcal{M}}$ by accepting words with the same projection (i.e $u$). Therefore, since $\Sigma$ is not ambiguous w.r.t $\mathcal{L}_{\mathcal{M}}$, $\forall s \in S$ such that $e \in out_{\mathcal{M}}(s)$, we have $\lambda_{\mathcal{M}}(s, e) = \mathcal{A}$. It follows by construction that $\forall s' \in S', \exists s'' \in S', \exists s \in S$ such that $s \xrightarrow{e/\mathcal{A}}_{\mathcal{M}} s''$ and such that $s' \in \Sigma\text{-}closure(s'')$, that is $\exists v \in ((\Sigma_{\mathcal{M}} \setminus \Sigma) \times 2^{A_{\mathcal{M}}})^*$ such that $s' \xrightarrow{v}_{\mathcal{M}} s''$. We can conclude that $\forall s'' \in S', \exists s'' \in S', \exists s \in S$ such that $s^0_{\mathcal{M}} \xrightarrow{u'}_{\mathcal{M}} s \xrightarrow{e/\mathcal{A}}_{\mathcal{M}} s'' \xrightarrow{v}_{\mathcal{M}} s'$ with $\pi(u' \cdot (e, \mathcal{A}) \cdot v, \Sigma) = \pi(u', \Sigma) \cdot \pi((e, \mathcal{A}), \Sigma) \cdot \pi(v, \Sigma) = u \cdot (e, \mathcal{A}) = w$.

Then, we prove that if $\Sigma$ is not ambiguous w.r.t $\mathcal{L}_{\mathcal{M}}$ then $\pi(\mathcal{L}_{\mathcal{M}}, \Sigma) \subseteq \mathcal{L}_{\pi(\mathcal{M},\Sigma)}$, or in other words that $\forall w \in \mathcal{L}_{\mathcal{M}}, \pi(w, \Sigma) \in \mathcal{L}_{\pi(\mathcal{M},\Sigma)}$. We prove by induction on $|w|$, that if $s^0_{\mathcal{M}} \xrightarrow{w}_{\mathcal{M}} s$ then $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{\pi(w,\Sigma)}_{\pi(\mathcal{M},\Sigma)} S$ with $s \in S$.

- For the base case, $|w| = 0$, and $w = \epsilon$. We have $\pi(w, \Sigma) = \epsilon$ accepted in $s^0_{\mathcal{M}}$. By construction $\epsilon$ is accepted in $S^0_{\pi(\mathcal{M},\Sigma)}$ and $s^0_{\mathcal{M}} \in S^0_{\pi(\mathcal{M},\Sigma)} = \Sigma\text{-}closure(s^0_{\mathcal{M}})$.
- For the induction step, we assume as inductive hypothesis that with $|w| = n$, if $s^0_{\mathcal{M}} \xrightarrow{w}_{\mathcal{M}} s$, $\exists S \in S_{\pi(\mathcal{M},\Sigma)}$ such that $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{\pi(w,\Sigma)} S$ and $s \in S$. Now, if $|w| = n + 1$, we can write $w = u \cdot (e, \mathcal{A})$. So we have $s^0_{\mathcal{M}} \xrightarrow{u}_{\mathcal{M}} s \xrightarrow{e/\mathcal{A}}_{\mathcal{M}} s'$. By inductive hypothesis, $\exists S \in S_{\pi(\mathcal{M},\Sigma)}$ such that $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{\pi(u,\Sigma)}_{\pi(\mathcal{M},\Sigma)} S$ with $s \in S$. If $e \notin \Sigma$, we have $\pi(u \cdot (e, \mathcal{A}), \Sigma) = \pi(u, \Sigma)$ and $s' \in \Sigma\text{-}closure(s) \subseteq S$. If $e \in \Sigma$, similarly to the first part of the proof, since $\Sigma$ is not ambiguous w.r.t $\mathcal{L}_{\mathcal{M}}$, we have that $\forall s \in S$ such that $e \in out_{\mathcal{M}}(s)$, $\lambda_{\mathcal{M}}(s, e) = \mathcal{A}$. It follows by construction that $S^0_{\pi(\mathcal{M},\Sigma)} \xrightarrow{\pi(u,\Sigma)}_{\pi(\mathcal{M},\Sigma)} S \xrightarrow{e/\mathcal{A}}_{\pi(\mathcal{M},\Sigma)} S'$ with $\pi(u \cdot (e, \mathcal{A}), \Sigma) = \pi(u, \Sigma) \cdot (e, \mathcal{A})$ and by construction $s' \in S'$. ■

## 5.2 Combining restriction and partial event projection

First, in chapter 3, we proved that restriction gave us a solution to the action driven distribution problem. Then, in chapter 4, we proved that if an AMM was action free, the partial event projection gave us a solution to the event driven distribution problem. In the previous section, we have explained how to extend partial event projection to AMM's with action. This is because we believe that, if we combine these two constructions, it would allows us to solve the general distribution problem. Given an AMM $\mathcal{M}$, a partition $\{A_1, A_2\}$ of $A_{\mathcal{M}}$ and two subsets $\Sigma_1$, $\Sigma_2$ of $\Sigma_{\mathcal{M}}$ such that $\Sigma_1 \cup \Sigma_2 = \Sigma_{\mathcal{M}}$, it is our intuition that by (1) duplicating $\mathcal{M}$ on each site, (2) restricting the actions to respectively $A_1$ and $A_2$ and (3) projecting the restricted duplicates to respectively $\Sigma_1$ and $\Sigma_2$, we would obtain a solution to the problem if one exists. However, before we can formally prove this intuition, we need some preliminary result.

**Lemma 5.1**
Given two AMM's $\mathcal{M}_1$, $\mathcal{M}_2$ such that $A_{\mathcal{M}_1} \cap A_{\mathcal{M}_2} = \emptyset$, $\mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2} = \{\ w \mid \pi(\rho(w, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1}) \in \mathcal{L}_{\mathcal{M}_1} \wedge \pi(\rho(w, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2}) \in \mathcal{L}_{\mathcal{M}_2}\}$.

**Proof**
This proof is divided in two parts. First we prove that if $w \in \mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2}$ then $\pi(\rho(w, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1}) \in \mathcal{L}_{\mathcal{M}_1}$ and $\pi(\rho(w, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2})$. We prove by induction on $|w|$ that if $(S^0_{\mathcal{M}_1}, S^0_{\mathcal{M}_2}) \xrightarrow{w}_{\mathcal{M}_1 \times \mathcal{M}_2} (S_1, S_2)$, then $S^0_{\mathcal{M}_1} \xrightarrow{\pi(\rho(w,A_{\mathcal{M}_1}),\Sigma_{\mathcal{M}_2})}_{\mathcal{M}_1} S_1$ and $S^0_{\mathcal{M}_1} \xrightarrow{\pi(\rho(w,A_{\mathcal{M}_2}),\Sigma_{\mathcal{M}_2})}_{\mathcal{M}_2} S_2$.

- For the base case, we have $|w| = 0$ and $w = \epsilon$. We have that $\epsilon$ is accepted from $(S^0_{\mathcal{M}_1}, S^0_{\mathcal{M}_2})$ in $\mathcal{M}_1 \times \mathcal{M}_2$. We also have that $\pi(\rho(w, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1}) = \epsilon$ is accepted from $S^0_{\mathcal{M}_1}$ in $\mathcal{M}_1$ and that $\pi(\rho(w, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2}) = \epsilon$ is accepted from $S^0_{\mathcal{M}_2}$ in $\mathcal{M}_2$.

- For the induction step, we assume as inductive hypothesis that with $|w| = n$, the implication above holds. Now if $|w| = n + 1$, we can write $w = u \cdot (e, \mathcal{A})$. Since $w \in \mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2}$, we have $(S^0_{\mathcal{M}_1}, S^0_{\mathcal{M}_2}) \xrightarrow{u}_{\mathcal{M}_1 \times \mathcal{M}_2} (S_1, S_2) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1 \times \mathcal{M}_2} (S'_1, S'_2)$ and by inductive hypothesis, $S^0_{\mathcal{M}_1} \xrightarrow{\pi(\rho(u, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1})}_{\mathcal{M}_1} S_1$ and $S^0_{\mathcal{M}_2} \xrightarrow{\pi(\rho(u, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2})}_{\mathcal{M}_2} S_2$. Then, depending on $e$ there are three possibilities:

  (i) if $e \in \Sigma_{\mathcal{M}_1} \cap \Sigma_{\mathcal{M}_2}$, by construction, we have $S_1 \xrightarrow{e/\mathcal{A}_1}_{\mathcal{M}_1} S'_1$ and $S_2 \xrightarrow{e/\mathcal{A}_2}_{\mathcal{M}_2} S'_2$ with $\mathcal{A}_1 \cup \mathcal{A}_2 = \mathcal{A}$. Since $A_{\mathcal{M}_1}$ and $A_{\mathcal{M}_2}$ are disjoint, we can deduce that $\mathcal{A}_1 = \mathcal{A} \cap A_{\mathcal{M}_1}$ and $\mathcal{A}_2 = \mathcal{A} \cap A_{\mathcal{M}_2}$. Therefore, we have $\pi(\rho(u, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1}) \cdot (e, \mathcal{A} \cap A_{\mathcal{M}_1}) = \pi(\rho(w, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1}) \in \mathcal{L}_{\mathcal{M}_1}$ and $\pi(\rho(u, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2}) \cdot (e, \mathcal{A} \cap A_{\mathcal{M}_2}) = \pi(\rho(w, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2}) \in \mathcal{L}_{\mathcal{M}_2}$.

  (ii) if $e \in \Sigma_{\mathcal{M}_1} \setminus \Sigma_{\mathcal{M}_2}$, by construction, we have $S_1 \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1} S'_1$ and $S_2 = S'_2$. Therefore, we have $\pi(\rho(u, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1}) \cdot (e, \mathcal{A}) = \pi(\rho(w, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1}) \in \mathcal{L}_{\mathcal{M}_1}$ and $\pi(\rho(u, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2}) = \pi(\rho(w, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2}) \in \mathcal{L}_{\mathcal{M}_2}$.

  (iii) if $e \in \Sigma_{\mathcal{M}_2} \setminus \Sigma_{\mathcal{M}_1}$, the proof is symmetrical to the previous case.

Then, we prove that if $\pi(\rho(w, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1}) \in \mathcal{L}_{\mathcal{M}_1}$ and $\pi(\rho(w, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2}) \in \mathcal{L}_{\mathcal{M}_2}$ then $w \in \mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2}$. We prove that if $S^0_{\mathcal{M}_1} \xrightarrow{\pi(\rho(w, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_2})}_{\mathcal{M}_1} S_1$ and $S^0_{\mathcal{M}_1} \xrightarrow{\pi(\rho(w, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2})}_{\mathcal{M}_2} S_2$ then $(S^0_{\mathcal{M}_1}, S^0_{\mathcal{M}_2}) \xrightarrow{w}_{\mathcal{M}_1 \times \mathcal{M}_2} (S_1, S_2)$ by induction on $|w|$.

- For the base case, we have $|w| = 0$ and $w = \epsilon$. We have that $\pi(\rho(w, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1}) = \epsilon$ is accepted from $S^0_{\mathcal{M}_1}$ in $\mathcal{M}_1$ and that $\pi(\rho(w, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2}) = \epsilon$ is accepted from $S^0_{\mathcal{M}_2}$ in $\mathcal{M}_2$. We also have that $\epsilon$ is accepted from $(S^0_{\mathcal{M}_1}, S^0_{\mathcal{M}_2})$ in $\mathcal{M}_1 \times \mathcal{M}_2$.

- For the induction step, we assume as inductive hypothesis that with $|w| = n$ the implication above holds. Now if $|w| = n + 1$, we can write $w = u \cdot (e, \mathcal{A})$. Depending on $e$, there are three possibilities:

  (i) if $e \in \Sigma_{\mathcal{M}_1} \cap \Sigma_{\mathcal{M}_2}$, then $\pi(\rho(w, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1}) = \pi(\rho(u, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1}) \cdot (e, \mathcal{A} \cap A_{\mathcal{M}_1}) \in \mathcal{L}_{\mathcal{M}_1}$ and $\pi(\rho(w, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2}) = \pi(\rho(u, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2}) \cdot (e, \mathcal{A} \cap A_{\mathcal{M}_2}) \in \mathcal{L}_{\mathcal{M}_2}$. It follows directly that $S^0_{\mathcal{M}_1} \xrightarrow{\pi(\rho(u, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1})}_{\mathcal{M}_1} S_1 \xrightarrow{e/\mathcal{A} \cap A_{\mathcal{M}_1}}_{\mathcal{M}_1} S'_1$ and $S^0_{\mathcal{M}_2} \xrightarrow{\pi(\rho(u, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2})}_{\mathcal{M}_2} S_2 \xrightarrow{e/\mathcal{A} \cap A_{\mathcal{M}_2}}_{\mathcal{M}_2} S'_2$. By inductive hypothesis, we have that $(S^0_{\mathcal{M}_1}, S^0_{\mathcal{M}_2}) \xrightarrow{u}_{\mathcal{M}_1 \times \mathcal{M}_2} (S_1, S_2)$, and by definition of $\times$, since $(\mathcal{A} \cap A_{\mathcal{M}_1}) \cup (\mathcal{A} \cap A_{\mathcal{M}_2}) = \mathcal{A}$, we have that $(S_1, S_2) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1 \times \mathcal{M}_2} (S'_1, S'_2)$.

  (ii) if $e \in \Sigma_{\mathcal{M}_1} \setminus \Sigma_{\mathcal{M}_2}$, then $\pi(\rho(w, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1}) = \pi(\rho(u, A_{\mathcal{M}_1}), \Sigma_{\mathcal{M}_1}) \cdot (e, \mathcal{A} \cap A_{\mathcal{M}_1}) \in \mathcal{L}_{\mathcal{M}_1}$ and $\pi(\rho(w, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2}) = \pi(\rho(u, A_{\mathcal{M}_2}), \Sigma_{\mathcal{M}_2}) \in \mathcal{L}_{\mathcal{M}_2}$. It follows directly that

$S^0_{\mathcal{M}_1} \xrightarrow{\pi(\rho(u,A_{\mathcal{M}_1}),\Sigma_{\mathcal{M}_1})}_{\mathcal{M}_1} S_1 \xrightarrow{e/\mathcal{A}\cap A_{\mathcal{M}_1}}_{\mathcal{M}_1} S'_1$ and $S^0_{\mathcal{M}_2} \xrightarrow{\pi(\rho(u,A_{\mathcal{M}_2}),\Sigma_{\mathcal{M}_2})}_{\mathcal{M}_2} S_2$. By inductive hypothesis, we have that $(S^0_{\mathcal{M}_1}, S^0_{\mathcal{M}_2}) \xrightarrow{u}_{\mathcal{M}_1\times\mathcal{M}_2} (S_1, S_2)$, and by definition of $\times$, we have that $(S_1, S_2) \xrightarrow{e/\mathcal{A}}_{\mathcal{M}_1\times\mathcal{M}_2} (S'_1, S'_2)$ with $S'_2 = S_2$.

(iii) if $e \in \Sigma_{\mathcal{M}_2} \setminus \Sigma_{\mathcal{M}_1}$, the proof is symmetrical to the previous case. ■

This lemma is the equivalent of lemma 4.1 of chapter 4. It allows us to characterize the language of a synchronized product of AMM's with disjoint sets of actions. The hypothesis of disjoint sets of actions are necessary for the proof's purpose. But this is not a problem since $A_1$ and $A_2$ are disjoint, in the general distribution problem. We can now prove that our construction is correct, and most importantly, complete.

**Theorem 5.3**

Given an AMM $\mathcal{M}$, a partition $\{A_1, A_2\}$ of $A_\mathcal{M}$ and two subsets $\Sigma_1$, $\Sigma_2$ of $\Sigma_\mathcal{M}$ such that $\Sigma_1 \cup \Sigma_2 = \Sigma_\mathcal{M}$, let $\mathcal{M}_1 = \pi(\rho(\mathcal{M}, A_1), \Sigma_1)$ and $\mathcal{M}_2 = \pi(\rho(\mathcal{M}, A_2), \Sigma_2)$. If there exists $\widehat{\mathcal{M}_1}$, $\widehat{\mathcal{M}_2}$ such that $\Sigma_{\widehat{\mathcal{M}_1}} = \Sigma_1$, $\Sigma_{\widehat{\mathcal{M}_2}} = \Sigma_2$, $A_{\widehat{\mathcal{M}_1}} = A_1$, $A_{\widehat{\mathcal{M}_2}} = A_2$ and such that $\widehat{\mathcal{M}_1} \times \widehat{\mathcal{M}_2} \equiv_B \mathcal{M}$, then $\mathcal{M}_1 \times \mathcal{M}_2 \equiv_B \mathcal{M}$

**Proof**

We will prove equivalently that $\mathcal{L}_{\mathcal{M}_1\times\mathcal{M}_2} = \mathcal{L}_\mathcal{M}$. First since $A_1$ and $A_2$ are disjoint, by lemma 5.1, we have that $\mathcal{L}_{\mathcal{M}_1\times\mathcal{M}_2} = \{\ w \mid \pi(\rho(w, A_1), \Sigma_1) \in \mathcal{L}_{\widehat{\mathcal{M}_1}} \wedge \pi(\rho(w, A_2), \Sigma_2) \in \mathcal{L}_{\widehat{\mathcal{M}_2}}\}$. But, since $\widehat{\mathcal{M}_1} \times \widehat{\mathcal{M}_2} \equiv_B \mathcal{M}$, we also have that $\mathcal{L}_{\widehat{\mathcal{M}_1}\times\widehat{\mathcal{M}_2}} = \mathcal{L}_\mathcal{M}$. We can deduce that $\forall w \in \mathcal{L}_\mathcal{M}$, $\pi(\rho(w, A_1), \Sigma_1) \in \mathcal{L}_{\widehat{\mathcal{M}_1}}$ and $\pi(\rho(w, A_2), \Sigma_2) \in \mathcal{L}_{\widehat{\mathcal{M}_2}}$. It follows directly that $\pi(\rho(\mathcal{L}_\mathcal{M}, A_1), \Sigma_1) \subseteq \mathcal{L}_{\widehat{\mathcal{M}_1}}$ and that $\pi(\rho(\mathcal{L}_\mathcal{M}, A_2), \Sigma_2) \subseteq \mathcal{L}_{\widehat{\mathcal{M}_2}}$. By contraposition of corollary 5.1, it follows that $\Sigma_1$ is not ambiguous w.r.t $\rho(\mathcal{L}_\mathcal{M}, A_1)$ and that $\Sigma_2$ is not ambiguous w.r.t $\rho(\mathcal{L}_\mathcal{M}, A_2)$. It follows, by theorem 5.2, that $\mathcal{L}_{\mathcal{M}_1} = \mathcal{L}_{\pi(\rho(\mathcal{M},A_1),\Sigma_1)} = \pi(\rho(\mathcal{L}_\mathcal{M}, A_1), \Sigma_1)$ and that $\mathcal{L}_{\mathcal{M}_2} = \mathcal{L}_{\pi(\rho(\mathcal{M},A_2),\Sigma_2)} = \pi(\rho(\mathcal{L}_\mathcal{M}, A_2), \Sigma_2)$. We can deduce that $\forall w \in \mathcal{L}_\mathcal{M}$, we have $\pi(\rho(\mathcal{L}_\mathcal{M}, A_1), \Sigma_1) \in \mathcal{L}_{\mathcal{M}_1}$ and $\pi(\rho(\mathcal{L}_\mathcal{M}, A_2), \Sigma_2) \in \mathcal{L}_{\mathcal{M}_2}$. As a consequence, we have that $\mathcal{L}_\mathcal{M} \subseteq \{\ w \mid \pi(\rho(w, A_1), \Sigma_1) \in \mathcal{L}_{\mathcal{M}_1} \wedge \pi(\rho(w, A_2), \Sigma_2) \in \mathcal{L}_{\mathcal{M}_2}\}$ and by theorem 5.1, $\mathcal{L}_\mathcal{M} \subseteq \mathcal{L}_{\mathcal{M}_1\times\mathcal{M}_2}$. It is left to prove that $\mathcal{L}_{\mathcal{M}_1\times\mathcal{M}_2} \subseteq \mathcal{L}_\mathcal{M}$. For that, we have proven earlier that $\mathcal{L}_{\mathcal{M}_1} = \pi(\rho(\mathcal{L}_\mathcal{M}, A_1), \Sigma_1) \subseteq \mathcal{L}_{\widehat{\mathcal{M}_1}}$ and that $\mathcal{L}_{\mathcal{M}_2} = \pi(\rho(\mathcal{L}_\mathcal{M}, A_2), \Sigma_2) \subseteq \mathcal{L}_{\widehat{\mathcal{M}_2}}$. Then, we have:

$$\begin{aligned}
\mathcal{L}_{\mathcal{M}_1\times\mathcal{M}_2} &= \{\ w \mid \pi(\rho(w, A_1), \Sigma_1) \in \mathcal{L}_{\mathcal{M}_1} \wedge \pi(\rho(w, A_2), \Sigma_2) \in \mathcal{L}_{\mathcal{M}_2}\} \\
&\quad \text{(by lemma 5.1)} \\
&\subseteq \{\ w \mid \pi(\rho(w, A_1), \Sigma_1) \in \mathcal{L}_{\widehat{\mathcal{M}_1}} \wedge \pi(\rho(w, A_1), \Sigma_1) \in \mathcal{L}_{\widehat{\mathcal{M}_1}}\} \\
&\quad \text{(because } \mathcal{L}_{\mathcal{M}_1} \subseteq \mathcal{L}_{\widehat{\mathcal{M}_1}} \text{ and } \mathcal{L}_{\mathcal{M}_2} \subseteq \mathcal{L}_{\widehat{\mathcal{M}_2}}\text{)} \\
&\subseteq \mathcal{L}_{\widehat{\mathcal{M}_1}\times\widehat{\mathcal{M}_2}} \\
&\quad \text{(by lemma 5.1)} \\
&\subseteq \mathcal{L}_\mathcal{M} \\
&\quad \text{(because } \widehat{\mathcal{M}_1} \times \widehat{\mathcal{M}_2} \equiv_B \mathcal{M}\text{)}
\end{aligned}$$

Finally, $\mathcal{L}_\mathcal{M} \subseteq \mathcal{L}_{\mathcal{M}_1\times\mathcal{M}_2}$ and $\mathcal{L}_{\mathcal{M}_1\times\mathcal{M}_2} \subseteq \mathcal{L}_\mathcal{M}$ implies that $\mathcal{L}_\mathcal{M} = \mathcal{L}_{\mathcal{M}_1\times\mathcal{M}_2}$. ■

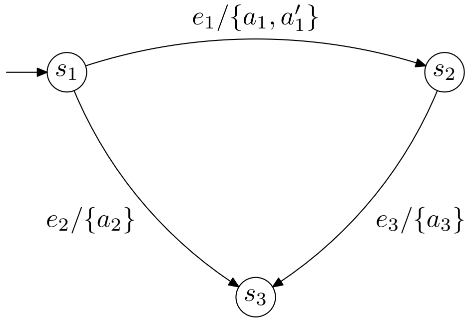



Figure 5.2: Finer grain distribution

This theorem leads to an effective procedure to solve the general distribution problem. Given an AMM $\mathcal{M}$, a partition $\{A_1, A_2\}$ of $A_\mathcal{M}$ and two subsets $\Sigma_1$, $\Sigma_2$ of $\Sigma_\mathcal{M}$ such that $\Sigma_1 \cup \Sigma_2 = \Sigma_\mathcal{M}$, we must first compute $\mathcal{M}_1 = \pi(\rho(\mathcal{M}, A_1), \Sigma_1)$ and $\mathcal{M}_2 = \pi(\rho(\mathcal{M}, A_2), \Sigma_2)$, and then compute their synchronized product $\mathcal{M}_1 \times \mathcal{M}_2$. Finally, theorem 5.3 assures us that $\mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2} \neq \mathcal{L}_\mathcal{M}$ then there exists no solution and if it is not the case, $\mathcal{M}_1$, $\mathcal{M}_2$ is a solution. Actually, instead of computing $\mathcal{M}_1 = \pi(\rho(\mathcal{M}, A_1), \Sigma_1)$ and $\mathcal{M}_2 = \pi(\rho(\mathcal{M}, A_2), \Sigma_2)$ separately and then check that $\mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2} \neq \mathcal{L}_\mathcal{M}$, we can directly compute $\pi(\rho(\mathcal{M}, A_1), \Sigma_1) \times \pi(\rho(\mathcal{M}, A_2), \Sigma_2)$ and check the bisimilarity along the way .

## 5.3 Discussion

In chapter 4, we have seen that an action free AMM $\mathcal{M}$ can be viewed as deterministic transition system over $\Sigma_\mathcal{M}$. In a certain way, an arbitrary AMM (possibly with actions) $\mathcal{M}'$ can also be viewed as a deterministic transition system, but this time over $\Sigma_{\mathcal{M}'} \times 2^{A_{\mathcal{M}'}}$. Therefore, one might ask why not simply use the results of [CMT99] on distributing transition systems, for our model of AMM, instead of the method we have described in the previous section. One main advantage of our method over [CMT99]'s method is that we allow a finer grain distribution. Indeed, the distribution, in our framework, allows to set the locations of each action independently from the locations of the events. As illustrated in example 5.2, if $\mathcal{M}$ contains a transition $s \xrightarrow{e/\{a,a'\}}_\mathcal{M} s'$ then, using [CMT99]'s method one would not be allowed to separate $a$ and $a'$. However, in our framework, it is possible.

**Example 5.2**
Let $\mathcal{M}$ be the AMM presented in figure 5.2. This AMM can be viewed as a transition system over $\{(e_1, \{a_1, a'_1\}), (e_2, \{a_2\}), (e_3, \{a_3\})\}$. Therefore, using [CMT99]'s method, we could not distribute this AMM is $a_1$ and $a_2$ are located on different execution site, whereas, in our framework, we can.

# Conclusion

In this work, we have studied the problem of distributing reactive systems. We have introduced the model of Action Mealy Machine which is very well adapted to describe those reactive systems. After formalizing the problem in chapter 2 in its most general form, we have studied and solved two subclasses of this problem in chapter 3 and 4. This study allowed us to better understand the problem inherent to our model and lead us to a solution for the general problem. In chapter 5, we have presented this solution and proved its correctness and its completeness. There remains however several open issues:

- In chapter 1, we stated that the model of Action Mealy Machine was inspired from the model of FIFFO-AUTOMATON. It would be interesting to adapt the distribution techniques we presented in this paper to this model.
- In chapter 3, the worst case complexity of the problem 3.2 of state minimal partitioning remains unknown.
- In chapter 4 and 5, we have seen that some AMM's are not distributable w.r.t the locations of the events and the actions. Could something be done to work around this situation? We could try to introduce some internal communication events in order to make those AMM's distributable. Given an AMM $\mathcal{M}$, it would therefore be interesting to study the problem of finding two AMM's $\mathcal{M}_1$ and $\mathcal{M}_2$ with $\Sigma_{\mathcal{M}} \subset \Sigma_{\mathcal{M}_1} \cup \Sigma_{\mathcal{M}_2}$ such that $\pi(\mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2}, \Sigma_{\mathcal{M}}) = \mathcal{L}_{\mathcal{M}}$.
- In chapter 5, we have presented a solution to the general distribution problem. However, it would be interesting to examine if and how this solution can imporved, maybe by merging states together as we did in chapter 3, for the action driven distribution problem.
- In our framework, the distributed AMM's communicate through synchronization. It would be interesting to investigate other forms of communication such as asynchronous communication. Events could for example be communicated by the means of FIFO channels[1].

[1]First In First Out

# Bibliography


[CMT99] Ilaria Castellani, Madhavan Mukund, and P. S. Thiagarajan. Synthesizing Distributed Transition Systems from Global Specification. In *Foundations of Software Technology and Theoretical Computer Science*, pages 219–231, 1999.

[Meu02] Cédric Meuter. Compilation et Répartition de programmes Electre pour Lego Mindstorm(tm). Master's thesis, Université Libre de Bruxelles, 2002.

[Mor99] René Morin. Decompositions of Asynchronous Systems. In *Proc. CONCUR'98, Springer Lect. Notes in Comp. Sci. 1466*, pages 549–565. Springer, 1999.

[RW92] Karen Rudie and W. Murray Wonham. Think Globally, Act Locally: Decentralized Supervisory Control. *IEEE Transactions on Automatic Control, Vol. 37, No. 11*, 1992.

[RW95] Karen Rudie and Jan C. Willem. The Computational Complexity of Decentralized Discrete-Event Control Problems. *IEEE Transaction on Automatic Control, Vol. 40, No. 7*, 1995.

[SEM03] Alin Stefănescu, Javier Esparza, and Anca Muscholl. Syntesis of Distributed Algorithm. In *To appear at 14th international conference on concurrency theory (CONCUR 2003)*, 2003.

[Tri01a] Stavros Tripakis. Problems and Examples of Decentralized Observation and Control for Discrete Event systems. In *Symposium on the Supervisory Control of Discrete Event Systems, 2001*, 2001.

[Tri01b] Stavros Tripakis. Undecidable Problems in Decentralized Observation and Control. *IEEE Transaction Conference on Decision and Control*, 2001.

# Index