UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION

A computational study of Algebraic Chemistry

Tesis para optar al grado de Magíster en ciencias mención Computación:

Tomás Veloz González.

Profesor Guía: Claudio Gutierrez Gallardo.

Miembros de la comisión:
Gonzalo Navarro
Alejandro Hevia
Marcelo Arenas

Santiago de Chile
Noviembre 2010.

Resumen

Las Químicas Algebraicas son una modelo abstracto para la bioquímica. Una Química Algebráica se compone de una red de reacciones moleculares, prescindiendo de una dinámica que permita estudiar su evolucion, pues el modelo se enfoca en como las moleculas pueden ser producidas o consumidas por las reacciones. Se ha probado que un tipo especial de subredes, llamadas organisaciones, son las unicas subredes que pueden tener estabilidad dinamica, una vez que es incorporada una dinamica en el modelo. Este hecho permite simplificar la comprension de la dinamica de los sistemas bioquímicos, dado que permite explicar la dinamica del sistema como movimientos entre organisaciones en el espacio de fase. De aquí que el computo del conjunto de organizaciones de una Química Algebráica es una tarea central en la teoría.

Al momento no han hay suficientemente buenos algorítmos para computar organisaciones, ni hay una comprensión de la estructura que subyace en la definición de organizacion (tal vez es esto es la razon de lo anterior). Esta tesis es un intento por formalizar el trabajo algorítmico en Químicas Algebráicas. Dicha formalizacion busca una fertilizacion cruzada entre modelos de Ciencias de la computacón y Químicas Alegebráicas.

Es posible enmarcar las Químicas Algebráicas en algunos conocidos formalismos de la Ciencia de la computación como Sistemas de Adición de Vectores y Redes de Petri. Se investiga la equivalencia entre los formalismos mencionados y las Químicas Algebráicas. Luego algunos conocidos problemas de los Sistemas de Adición de Vectores y Redes de Petri tales como reachability, liveness, etc., son estudiados desde la perspectiva de las Químicas Algebrícas, enfocando el análisis a la relacion de dichos problemas con el problema de computar organizaciones. De las ideas que surgen del anterior analisis, se hace posible el desarrollo de varios resultados sobre el computo de organizaciones, asi como sobre la estructura del conjunto total de organizaciones de una Química Algebríca. Un teorema de descomponer una organizacion, en subsistemas mas simples, y sus implicaciones son derivados (Teorema 14, Corolario 25, Corollary 26) como los resultados mas importantes de esta tesis. Los resultados de este trabajo hacen posible el desarrollo de nuevos y mas eficientes algoritmos para el cómputo de organizaciones y permite separar diferentes clases de Químicas Algebráicas en terminos de la dificultad de computar su conjunto de organizaciones (ver Figura 9.1).

Summary

Algebraic Chemistry (AC) is an abstract bio-chemical model. An AC consists of a reaction network, but neglects the dynamical evolution, by only focusing on how the reactions produce and consume molecules. It has been proved that some special subnetworks, the so-called organisations, are the only possible subnetworks which can be dynamically stable, once dynamical evolution is considered. This fact provides a simplification in the understanding of the dynamics of bio-chemical systems, because it is possible to explain the evolution of a bio-chemical system as movement between organisations in the phase space. Then computing the set of organisations is a central task in AC.

There are not good enough algorithms to compute organisations, perhaps because there is not good enough understanding of the underlying structure of organisations. This thesis is an attempt to formalize the algorithmic work in Algebraic chemistry. This formalization pursues the cross-fertilization between computer science models and Algebraic chemistry.

It is possible to frame the AC in well-known concurrent processing formalisms such as Vector Addition Systems and Petri Nets. We state the equivalence between these formalisms and AC. We explore some well-known problems of Vector Addition Systems and Petri Nets, such as reachability, liveness, etc., from the perspective of AC, by explaining the relation between these properties and organisations. From the insights which have arose from those exploration, we developed several results about the computation of organisations, as well as about the structure of the set of organisations of an AC. A decomposition theorem for organizations and its implications (Theorem 14, Corollary 25, Corollary 26) are the most important results of this work. The results of this work makes possible the development of new and more efficient algorithms to compute the organisations of an AC and also makes possible to separate different classes of AC in terms of how complicated is compute their set of organisations (see Figure 9.1).

Publications

- Peter S., Veloz T., Dittrich P.: Feasibility of Organizations -A Refinement of Chemical Organization Theory, Eleventh International Conference on Membrane Computing, Accepted paper (In press).

Other results

- Research visits: I had the chance to visit the Jena Centre of bioinformatics, Germany, where Peter Dittrich (the developer of AC) works. I explained him my desires to research in the computational aspects of AC and they invite me to visit their center and share experiences. An the same time, I also contacted Marie-France Sagot, the head of the BAMBOO team of University Claude Bernard I, Lyon, France. They were also interested in researching about computational aspects of systems biological models, thus they invited me too. I was since February to April 2010, six weeks in Germany and two in France. Most of the results of this thesis were developed on that period.

Contents

List of Figures

List of Definitions

List of Theorems

List of Lemmas

List of Corollaries

List of Algorithms

List of Procedures

Chapter 1

Introduction

One of the main driving forces of science is the quest for understanding the origin and nature of life.

Artificial Life is developing models to simulate and predict the behaviour of living systems, such as organisms, systems of organisms (colonies) and societies (systems of colonies) from an abstract point of view. Abstract models that could explain the origin of evolutionary systems would allow to investigate the theoretical conditions of the origin and evolution of life. One of those abstract models is called Reaction Network System (RNS), developed to deal with chemical reaction systems. A RNS consists of objects, and interaction rules among objects that lead to the appearance or disappearance of other objects.

RNS are not used only to model chemical phenomena. Their applications range from ecology, proto-biology, systems biology, bioinformatics to computer science and have reached even the study of language and social systems (see [14]). All these refers to the same concept: objects, that through the reaction laws generate other objects.

There is an interesting approach to RNS called Algebraic Chemistry (AC), developed by Peter Dittrich and Speroni di Fenizio [14]. An AC consists of a chemical reaction network. Quite differently to classical approaches to RNS, AC does not include explicitely the *chemical dynamics*, *i.e* deterministic or probabilistic evolution rules following a discrete or continuous schema. Instead, AC focuses on the possible ways to produce and consume the molecules in the network. Hence the study of AC pursues the understanding of properties that subnetworks, *i.e* subsets of molecules and the subset of rules in which the considered subset of molecules participate, can have.

Dittrich *et al.* in their seminal paper [14] define an organization as a special kind of subnetwork which is able to maintain completely its structure in the reaction network. They also shows how to decompose the AC in a *hierarchy of organizations*. Furthermore, they show that if a chemical dynamics is incorporated to the system (to perform simulations), the hierarchy of organizations represents *all feasible system states* including *all steady states* of the network. Then, the relevancy of AC is that it represents the required input data structure, from where to derive the long-term behaviour of the RNS.

AC gives an interesting insight, from the computational point of view, to the search of stable molecules of bio-chemical systems, such as immunity systems or metabolism of the first procaryotes cells (beginning of life), because provides a tool which would avoid the combinatorial explosion of possible stable molecules in a RNS [10].

Experiments are performed in AC by computer simulation, but still there are not enough advances in this field. In fact, Diettrich and Di Fenizio [14] write: *Identifying the computational complexity of algorithms able to find all organizations would be highly desirable. Especially it would be interesting to know, which kind of networks can be analyzed with contemporary computers. Developing efficient algorithms for computing or approximating the set of organizations and estimating the computational complexity of these algorithms is a separate research topic for the future.*

This work lays its foundation based on this idea. It is interesting to note that properties defined in AC can be reformulated in several other formalisms, for example Vector Addition Systems and Petri Nets. Vector Addition System and Petri Nets (in short VAS and PN respectively) are models to study parallel and concurrent processing. It is proved that VAS are equivalent to Petri Nets and to other formalisms to study concurrent computation [13]. Algorithmic work and complexity results in concurrent computation have been widely studied since the later sixties [24, 37, 21, 27, 19, 1, 16, 15].
Considering these developments, this thesis arises from the natural idea of making a cross-fertilization between AC's, VAS and PN: First, by framing the AC's in a well known computer science formalism, and second, giving new insights to VAS and PN from the questions which arise from AC's. This work leads to a better understanding of the AC's from the computational point of view, and would provide a new conceptual framework for studying the complexity of some important problems of theoretical computing which are postulated in VAS and PN framework such as reachability, boundedness, liveness, etc. The usage of Petri Nets in bio-chemical formalisms has been proposed in the past [3, 4, 43, 44], but the AC formalism has never been approached from the point of view of concurrent computation. Thus we focus in what is more distinctive on AC theory with respect to the other bio-chemical approaches, this is, the study of organisations. This work deals with algorithmic and structural results about the process of compute the organisations of a reaction netowrk. This work is organized in two parts: In the first part it is presented the preliminaries of Algebraic chemistry are presented as well as an explanation of the most interesting properties with illustrative examples, to provide the linkage of AC properties with respect to the dynamics of the system. Finally in the first part is reviewed the current work in organisation computation. The second part deals with a discretization of AC. From this discretization, the AC formalism is framed in VAS and Petri Nets. Then, a detailed analysis on the relation between Petri Nets properties and organisations in AC is done. The insights of these analysis are used to develop several new algorithmic and structural results of organisations: It is introduced the notion of *role* of a molecule inside the reaction network, from this notion it is possible to decompose an organization into several subnetworks where some molecules act as *borders* of those subnetworks. Then the verification of the organization property can be decomposed into several smaller verifications (see Theorem 14 and Corollary 26).

Part I

Algebraic chemistries formalism

Introduction

Algebraic Chemistry (AC) arise from the necessity to develop a formal framework of organizations of living systems and chemical reactions. Its motivation is encountered in the understanding of the evolution of organisms, from the metabolic process they are (and they are not) able to accomplish. From the thermodynamics it is well-known that the long-term behaviour of a system[1] is explained as a movement between steady states triggered by external perturbations in the phase space. From the point of view of computation, contrary to dynamical approaches such as the evolution of the system in the time leaded by a set of rules (discrete, probabilistic or continuous), Algebraic chemistry pursue to capture the essence of the dynamics understanding which states would compose the steady states, and how should be the perturbations to move from one steady state to the other. We are going to prove later that the set of molecules corresponding to any fixed point of the dynamics of a system has to fulfill some special properties: First, closure, which means that no novel species will be produced by the set, and self-maintainability, which means that the set is capable to self-maintain its structure by firing the reactions. We call such sets organisations, they are the only candidates to be fixed point in the dynamics. So, instead of simulating *blindly* the dynamics of the system, it is possible to evolve only the organisations to understand which organisations correspond to fixed points and which perturbations make evolve one organisation to the other. Chemical organsations theory has been applied succesfully in several biochemical networks [11, 34, 35].

Algebraic chemistry have been applied to several other areas. For example, Dittrich *et al.* have shown in [32, 33] how the chemical organization theory can help in designing and understanding chemical computing systems. The result of a computation appears as an emergent global behavior based on local reaction rules. After providing a recipe for mapping logic circuits to chemical reaction rules, they discussed reaction networks implementing various logic circuits (an XOR, a flip-flop, and a controllable oscillator). In the field of P-systems it has been found by Stephan Peter, Peter Dittrich and the author that organisations and membranes act as complementary principles to explain the gainance or breakdown of stability in chemical systems [42].

In this part of the thesis we introduce the Algebraic Chemistry formalism. In Chapter 2, following [14], we present the basic definitions of AC. In Chapter 3 we present the current state of algorithms for study the AC. Several examples are presented to illustrate the concepts and help the understanding of the reader.

---

[1] A dissipative system.

Chapter 2

Algebraic Chemistry: Preliminaries

2.1 Basic definitions

Let $\mathcal{M}$ be a finite set of molecules and $\mathcal{R} \subseteq \mathcal{P}_M(\mathcal{M}) \times \mathcal{P}_M(\mathcal{M})$ a set of *reactions* occurring among the molecular species in $\mathcal{M}$ (we also call it the *rules* set[1]), where $\mathcal{P}_M(X)$ is the set of all multisets formed by elements of the set X. We will denote the reaction $R = (A, B)$ in $\mathcal{R}$ as $R = A \to B$.

Definition **1 (Algebraic Chemistry)** *An Algebraic Chemistry is a pair* $\langle \mathcal{M}, \mathcal{R} \rangle$.

A multiset $H \in \mathcal{P}_M(\mathcal{M})$ will be denoted by $H = h_1 m_1 + \cdots + h_n m_{|\mathcal{M}|}$, where h_j denotes the multiplicity of each molecular specie $m_j,\ j = 1, ..., |\mathcal{M}|$. We say that the *multiplicity* (or *amount*) of a given molecule $m_i \in \mathcal{M}$ in the multiset H (which corresponds to h_i) is denoted by $\mathcal{A}(H, m_i)$. We say a molecule $m_i \in \mathcal{M}$ is *present* in the multiset H if and only if its multiplicity $\mathcal{A}(H, m_i) > 0$. Given a reaction $R = A \to B$ we say that molecules $m \in \mathcal{M}$ which are present in A are the *reactants* and molecules $m \in \mathcal{M}$ which are present in B are the *products* of the reaction R.

We say that a reaction $R = (A \to B) \in \mathcal{R}$ *can be fired* by a set $X \subseteq \mathcal{M}$ iff the molecules present in A are in X. A molecule $m \in \mathcal{M}$ is *produced* within a set $X \subseteq \mathcal{M}$, if there exists a reaction $(A \to B) \in \mathcal{R}$ which can be fired by X and $\mathcal{A}(A, m) < \mathcal{A}(B, m)$. Conversely, a molecule $m \in \mathcal{M}$ is *consumed* within a set $X \subseteq \mathcal{M}$, if there exists a reaction $(A \to B) \in \mathcal{R}$ which can be fired by X and $\mathcal{A}(A, m) > \mathcal{A}(B, m)$. From now on, let $\mathcal{R} = \{R_1, ..., R_k\}$, where $R_i = A_i \to B_i, i = 1, ..., k$ and $\mathcal{M} = \{m_1, \ldots, m_n\}$. We define the *stoichiometric matrix* $\mathbf{S} = (s_{ij})$ as the $n \times k$ matrix associated to the algebraic chemistry $\langle \mathcal{M}, \mathcal{R} \rangle$, where $s_{ij} = \mathcal{A}(B_j, m_i) - \mathcal{A}(A_j, m_i)$. s_{ij} represents the amount of molecular species of the type m_i produced by reaction R_j (s_{ij} is negative if molecule m_i is consumed by reaction R_j) within the set $\mathcal{M}$.

We now need to define how the reactions take place in the AC.

We define the *flux vector* $\mathbf{v} = (v_1, ..., v_k) \in \mathbb{R}_+^k$, where $\mathbb{R}_+$ denotes the set of non negative real

---

[1] They are called reactions in a chemical context and rules in computational context.

numbers. The flux vector $\mathbf{v}$ represents the rate of the reaction R_i in the network. Then, the application of the stoichiometric matrix $\mathbf{S}$ over the flux vector $\mathbf{v}$, represent a reaction process in the network such that the rate of the reaction R_i is v_i for $i = 1, ..., k$.
Finally, we define the *production rate* $f_i = \sum_{j=1}^{k} s_{ij} v_j$ of every molecule $m_i, i = 1, \ldots, n$, which represents the increase ($f_i > 0$), decrease ($f_i < 0$) or conservation ($f_i = 0$) of the total amount of the molecular species m_i in the reaction process specified by $\mathbf{v}$ and $\mathbf{S}$. The *production rate vector* is defined by $\mathbf{f} = (f_1, \ldots, f_n)$.

Summary of definitions:

1. $\mathcal{M} = \{m_1, ..., m_n\}$ set of molecules.
2. $\mathcal{R} = \{A_1 \to B_1, ..., A_k \to B_k\}$ set of reactions.
3. $\mathcal{A}(H, m_j)$ the amount of molecular species m_j in the multiset H.
4. Present: $m_i \in \mathcal{M}$ is present in the multiset H iff $\mathcal{A}(H, m) > 0$.
5. Reactant and Product: The reactants (respectively products) of a reaction $A \to B$ are the molecules present in A (respectively B).
6. Fireable: A reaction $R = A \to B$ can be fired by a set X iff every molecule present in A is in X.
7. Produced and Consumed: A molecule m_i is produced (respectively consumed) by a set $X \subseteq \mathcal{M}$ iff there exist a reaction $A \to B \in \mathcal{R}$ which can be fired by X and $\mathcal{A}(A, m_i) < \mathcal{A}(B, m_i)$ (respectively $\mathcal{A}(A, m_i) > \mathcal{A}(B, m_i)$).
8. Stoichiometric matrix: $\mathbf{S} = (s_{ij}) = \mathcal{A}(B_j, m_i) - \mathcal{A}(A_j, m_i),\ i = 1, ...n,\ j = 1, ..., k.$
9. Flux vector: The flux vector $\mathbf{v} \in \mathbb{R}_{+}^{k}$ represents the rate of the reaction R_i in the network.
10. Production rate vector: $\mathbf{f} = \mathbf{S}\mathbf{v}$.

We are going to show the concepts defined in this section in an example

Example 1. Let $\mathcal{M} = \{s_1, s_2, s_3\}$, $\mathcal{R} = \{R_i \ s.t \ R_i = A_i \to B_i,\ 1 \leq i \leq 4\}$ where

$$R_1 = 2s_1 \to s_2,\ R_2 = 2s_2 \to s_2 + s_3,\ R_3 = s_3 \to \emptyset,\ R_4 = s_2 + 2s_3 \to 4s_1.$$

From the algebraic chemistry stated above, the following statements are given to illustrate the concepts defined in this section.

- $\mathcal{A}(A_1, s_1) = 2$, $\mathcal{A}(A_1, s_2) = 0$, $\mathcal{A}(A_1, s_3) = 0$.

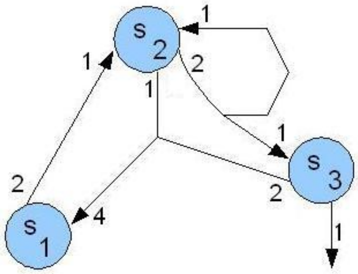

Figure 2.1: The algebraic chemistry of Example 1

- The only molecule present in A_1 is s_1. Molecules s_2 and s_3 are present in B_2.
- The reactants of R_4 are s_2 and s_3, the only reactant of R_2 is s_2. The products of R_2 are s_2 and s_3 and reaction R_3 has no products.
- Reaction R_4 can be fired neither by set $\{s_1, s_2\}$ nor by $\{s_1, s_3\}$, but R_4 can be fired by $\{s_2, s_3\}$.
- s_2 is produced by R_1, consumed by R_4 and is not consumed nor produced by R_3.
- The stoichiometric matrix associated of $\langle \mathcal{M}, \mathcal{R} \rangle$ is

$$\mathbf{S} = \begin{pmatrix} -2 & 0 & 0 & 4 \\ 1 & -1 & 0 & -1 \\ 0 & 1 & -1 & -2 \end{pmatrix}$$

- Applying the flux vector $\mathbf{v}_1 = (2, 1, 1, 1)$ and $\mathbf{v}_2 = (1, 2, 1, 1)$to the stoichiometric matrix of $\langle \mathcal{M}, \mathcal{R} \rangle$ we obtain the production rate $\mathbf{S}\mathbf{v}_1 = \mathbf{f}_1 = (0, 0, -2)$ and $\mathbf{S}\mathbf{v}_2 = \mathbf{f}_2 = (2, -2, -1)$ respectively.

2.2 Properties of subsets of $\mathcal{M}$

Given the AC $\langle \mathcal{M}, \mathcal{R} \rangle$, there are some interesting properties which the subsets of $\mathcal{M}$ can hold. We are going to define those properties in this section.

Definition **2 (Firable reaction set)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $\mathcal{Y} \subseteq \mathcal{M}$. Let $R_{\mathcal{Y}}$ be composed by the reactions $R_i = A_i \to B_i \in \mathcal{R}$ such that, if a molecule $m \in \mathcal{M}$ is present in A_i, then $m \in \mathcal{Y}$. We call $\mathcal{R}_{\mathcal{Y}}$ the possible (or firable) reactions set of $\mathcal{Y}$.*

Remark: The application of a flux vector $\mathbf{v} = (v_1, ..., v_k)$ such that $v_i > 0$ only if the reaction $R_i \in \mathcal{R}_{\mathcal{Y}}$, otherwise $v_i = 0$, to the stoichiometric matrix $\mathbf{S}$, lead to a production rate vector

$\mathbf{f} = \mathbf{S}\mathbf{v}$ constrained to the algebraic chemistry $\langle \mathcal{Y}, \mathcal{R}_{\mathcal{Y}} \rangle$. This is equivalent to apply a reduced flux vector $\bar{\mathbf{v}} \in \mathbb{R}^l$ where $l = |\mathcal{Y}|$ to a reduced stoichiometric matrix $\mathbf{S}_{\mathcal{Y}}$ implied only by the firable reaction set of $\mathcal{Y}$. In order to avoid introducing extra notation we define:

Definition 3 **(Flux vector constrained to a set)** *Let* $\mathcal{Y} \subseteq \mathcal{M}$. $\mathbf{v} = (v_1, ..., v_k)$ *is constrained to* $\mathcal{Y}$ *if and only if* $R_i \in \mathcal{R}_{\mathcal{Y}}$ *implies* $v_i > 0$ *and* $R_i \notin \mathcal{R}_{\mathcal{Y}}$ *implies* $v_i = 0$.

We now introduce the notions of production and consumption of molecules in the reaction process specified by a flux vector in a given stoichiometric matrix.

Definition 4 **(Produced-consumed set)** *Let* $\mathcal{Y} \subseteq \mathcal{M}$ *and* $\mathbf{v} = (v_1, ..., v_k)$ *constrained to* $\mathcal{Y}$. *Let*

$$P_{\mathcal{Y}}^{\mathbf{v}} = \{m_i \in \mathcal{M} \textit{ such that } \mathbf{f}_i > 0\},$$

$$C_{\mathcal{Y}}^{\mathbf{v}} = \{m_i \in \mathcal{Y} \textit{ such that } \mathbf{f}_i < 0\}.$$

$P_{\mathcal{Y}}^{\mathbf{v}}$ *is called the produced set of* $\mathbf{v}$, $C_{\mathcal{Y}}^{\mathbf{v}}$ *is called the consumed set of* $\mathbf{v}$. *Furthermore we define the hold set as* $H_{\mathcal{Y}}^{\mathbf{v}} = \mathcal{Y} - (C_{\mathcal{Y}}^{\mathbf{v}} \cup P_{\mathcal{Y}}^{\mathbf{v}})$.

Remark: In the reactions process new molecules can be added, existing molecules can be eliminated and the rest of the existing molecules are hold in the network. Therefore, the reaction process specified by a flux vector $\mathbf{v}$ potentially lead to a new set of molecules.
Remark: Note that molecules can be reactant and product simultaneously in the reactions, for example the reaction $a + b \to 2a$ the molecule a is reactant and product, but $a \in P_{\{a,b\}}^{(1)}$ because the total production of a is positive. Then for every flux vector we have $C_{\mathcal{Y}}^{\mathbf{v}} \cap P_{\mathcal{Y}}^{\mathbf{v}} = \emptyset$.

Definition 5 **(Steady state set)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC,* $\mathcal{Y} \subseteq \mathcal{M}$ *and* $\mathbf{v}$ *be a flux vector constraied to* $\mathcal{Y}$. *We define the steady state set of* $\mathcal{Y}$ *induced from* $\mathbf{v}$ *by* $\mathcal{Y}^{\mathbf{v}} = (\mathcal{Y} - C_{\mathcal{Y}}^{\mathbf{v}}) \cup P_{\mathcal{Y}}^{\mathbf{v}}$.

The steady state set is composed by the molecular species which are conserved or produced in reaction process of the network specified by the flux vector $\mathbf{v}$. We now introduce the most important properties in AC, these notions refers to the stability that a set of molecules can have in the reaction process.

Definition 6 **(Closed set)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC and* $\mathcal{Y} \subseteq \mathcal{M}$. *We say* $\mathcal{Y}$ *is closed if and only if for every* $R_i = (A_i \to B_i) \in \mathcal{R}_{\mathcal{Y}}$, *all the products of* R_i *belong to* $\mathcal{Y}$.

Definition 7 **(Semi-self-maintaining set)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC and* $\mathcal{Y} \subseteq \mathcal{M}$. *We say* $\mathcal{Y}$ *is semi-self-maintaining if and only if every molecule* $m \in \mathcal{Y}$ *which is consumed within* $\mathcal{Y}$ *is also produced within* $\mathcal{Y}$.

Definition 8 **(Self-maintainaning set)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC and* $\mathcal{Y} \subseteq \mathcal{M}$. *Let* $\mathbf{S} = (s_{i,j})$ *be the* $(n \times k)$ *stoichiometric matrix implied by the set of reaction rules* $\mathcal{R}$. *A set of molecules* $\mathcal{Y} \subseteq \mathcal{M}$ *is self-maintaining, if and only if there exists a flux vector* $\mathbf{v} = (v_1, ..., v_k) \in \mathbb{R}_+^k$ *constrained to* $\mathcal{Y}$ *such that*

$$\mathbf{f} = \mathbf{S}\mathbf{v} \geq 0, \textit{ or equivalently } \mathcal{Y} \subseteq \mathcal{Y}^{\mathbf{v}} \textit{ (or } C_{\mathcal{Y}}^{\mathbf{v}} = \emptyset\textit{)}.$$

Definition 9 **((Semi-)organisation)** *A set of molecules $\mathcal{Y} \subseteq \mathcal{M}$ is a (semi-)organization if and only if is closed and (semi-)self-maintaining.*

Example 2. Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an algebraic chemistry where $\mathcal{M} = \{a, b, c, d\}$ and $\mathcal{R} = \{a \to b, 2b + c \to a, \emptyset \to c, d \to abcd\}$. The stoichiometric matrix $\mathbf{S}$ in this case is

$$\mathbf{S} = \begin{pmatrix} -1 & 1 & 0 & 1 \\ 1 & -2 & 0 & 1 \\ 0 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

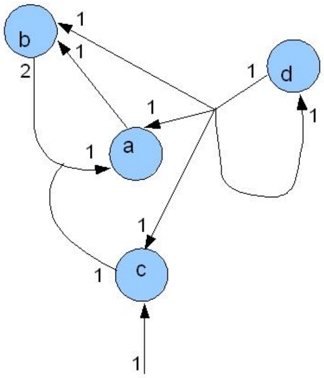

Figure 2.2: The algebraic chemistry of Example 2

- The set $\{a\}$ is not closed, nor (semi-)self-maintaining.
- The set $\{d\}$ is self-maintaining, but not closed. The sets $\{b\}$, $\{c\}$ are organization.
- The set $\{a, b\}$ is closed, but is not (semi-)self-maintaining.
- The set $\{a, b, c\}$ is a semi-organization, but is not self-maintaining[2].
- The set $\{a, b, c, d\}$ is an organization. The self-mantainance of $\{a, b, c, d\}$ is verified by the flux vector $\mathbf{v} = (1, 1, 1, 1)$, which gives the production rate $\mathbf{f} = (1, 0, 1, 0)$.

2.3 Dynamic analysis

Sections 2.1 and 2.2 deals with molecules $\mathcal{M}$ and their reaction rules $\mathcal{R}$, but not with the evolution of the system in time. We are going to define the dynamics of an AC: In general

---

[2] To verify this, it is necessary to construct the system of inequations $\mathbf{Sv} \geq 0$, with $\mathbf{v} = (v_1, v_2, v_3, v_4)$, where $v_i > 0,\ i = 1, 2, 3$ and $v_4 = 0$. This leads to an unsolvable system of inequations.

terms, the *dynamics* is given by a *state space* X and a formal definition (mathematical or algorithmic) that describes all possible movements in X only. Given an initial state $\mathbf{x}_0 \in X$, the formal definition describes how the state changes over time. We assume a deterministic dynamical process, which can be formalized by a phase flow $(X, (T_t)_{t\in\mathbb{R}})$ where $(T_t)_{t\in\mathbb{R}}$ is a one-parametric group of transformations from X. $T_t(\mathbf{x}_0)$ denotes the state at time t of a system that has been in state $\mathbf{x}_0$ at $t = 0$.

DEFINITION **10 (Reaction vessel)** *The state x_0 of the AC at some time t is called* $\mathbf{t}-$**reaction vessel**, *or simply the reaction vessel.*

From now on, we will use the term reaction vessel when we refer to the state of the network without concerning the time. This will simplify the explanations. In Sections 2.1 and 2.2 we considered just the presence or absence of molecular species (by choosing a subset $\mathcal{Y} \subseteq \mathcal{M}$) in the reaction vessel, but we did not considered another possible (and highly desirable to consider for a dynamic analysis) properties of the molecules in the reaction vessel, such as their concentrations, spatial distributions, velocities, and so on.
We assume in this section the same algebraic chemistry $\langle \mathcal{M}, \mathcal{R} \rangle$ that in previous sections, the state space $X = \mathbb{R}^n$, $\mathbf{x} = (x_1, ..., x_n)$ is a concentration vector, where x_i is a non-negative function of time and $x_i(t)$ denotes the concentration of molecular species m_i, $i = 1, ..., n$, in the reaction vessel (at time t). The dynamics is given by a system of ordinary differential equations of the form $\dot{\mathbf{x}} = \mathbf{S}\mathbf{v}(\mathbf{x})$ where $\mathbf{S}$ is the stoichiometric matrix implied by $\langle \mathcal{M}, \mathcal{R} \rangle$ and $\mathbf{v}(\mathbf{x}) = (\mathbf{v}_1(\mathbf{x}), \ldots, \mathbf{v}_k(\mathbf{x})) \in \mathbb{R}^k$ is a vector of non-negative functions[3]. For $j = 1, \ldots, k$, the j-th component $\mathbf{v}_j(\mathbf{x})$ of $\mathbf{v}(\mathbf{x})$ describes (as in Section 2.1) the rate of reaction of the reaction $R_j \in \mathcal{R}$, but in this section the flux vector depends on the current concentration $\mathbf{x}$. For the function $\mathbf{v}_j$ we require that $\mathbf{v}_j(\mathbf{x})$ is positive if and only if for every molecule $m_i \in \mathcal{M}$ is hold that if m_i is a reactant of the reaction R_j, then m_i has a concentration $x_i > 0$ in the state $\mathbf{x}$. Otherwise $\mathbf{v}_j(\mathbf{x}) = 0$. We call $\mathbf{v}(x)$ the *flux vector function.*

Example 3. A common way to study the dynamics of real systems is using the law of mass action kinetics [18, 25]. Suppose we are going to study a real system in which $\mathcal{M} = \{a, b\}$ and $\mathcal{R} = \{a \to b, a + b \to 2a\}$. In mass action kinetics the coordinates of the flux vector are composed by the product of the concentration of all the reactants of the reaction, each concentration to the power of the amount of times which is required in the reaction. In this case, let x_a and x_b the concentrations of molecule a and b respectively. The dynamics is ruled by

$$x'_a(t) = -k_1 x_a(t) + k_2 x_a(t) x_b(t),\ \ x'_b(t) = k_1 x_a(t) - k_2 x_a(t) x_b(t),$$

where k_1 and k_2 are constant reaction rates determined experimentally.

In order to link the dynamic analysis of this section with the non-dynamic (static) analysis introduced in Section 2.1, given the state $\mathbf{x}$ of the reaction vessel, we need a function that maps uniquely this state to the set of molecules $m_i \in \mathcal{M}$ such that $x_i > 0$. Analogously, given a set of molecules $\mathcal{Y} \subseteq \mathcal{M}$, we need to know which states from X correspond to this set of molecules. For this reason we introduce the concepts of *abstraction* and *instance.*

---

[3] In Sections 2.1 and 2.2 we were not considering the dynamics, therefore the rate of reaction $\mathbf{v}_j(\mathbf{x})$ was represented by some non-negative amount of reactions v_j in the network. Then, the flux vector was represented by a non-negative integer vector $\mathbf{v}$ instead of a vector of non-negative functions $\mathbf{v}(\mathbf{x})$.

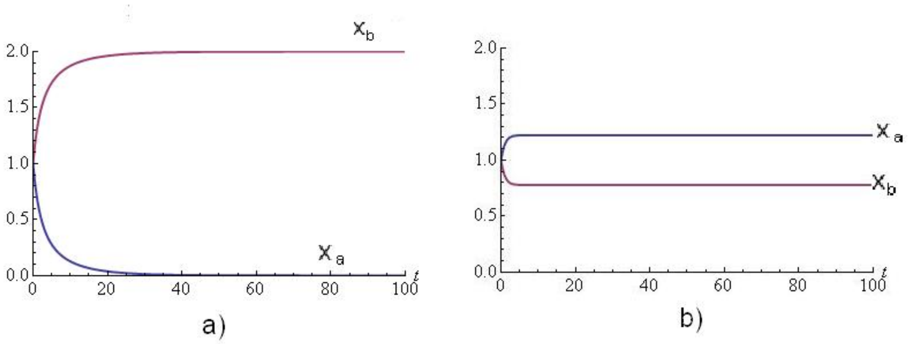

Figure 2.3: Evolution in time of the algebraic chemistry of Example 3 with initial conditions $x_a = x_b = 1$. In the plot **a)** the constant reaction rates are $k_1 = 0.7$, $k_2 = 0.3$. In the plot **b)** the constant reaction rates are $k_1 = 0.7$, $k_2 = 0.9$. Note that in plot **a)** the asymptotic behaviour of the system tends to $x_a \equiv 0$ and $x_b \equiv \bar{x} > 0$, this means that in stationary regime the only molecular specie present in the reaction vessel is b. In plot **b)** both molecules are present in the reaction vessel in asymptotic regime.

Definition **11 (Abstraction)** *Let* $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ *be a dynamical system and* $\mathbf{x}$ *be a state in* X. *An abstraction* ϕ *is defined by:*

$$\phi : X \mapsto \mathcal{P}(\mathcal{M}), \qquad \Theta > 0, \qquad \phi(\mathbf{x}) = \{m_i | x_i > \Theta, i \in \mathcal{M}\}, \tag{2.1}$$

where x_i *is the concentration of molecular species* m_i *in the state* $\mathbf{x}$, *and* Θ *is a threshold chosen such that it is smaller than any positive coordinate of any fixed point of* $\dot{\mathbf{x}} = f(\mathbf{x})$.

Definition **12 (Instance)** *Given an abstraction* $\phi : X \mapsto \mathcal{P}(\mathcal{M})$ *we say that a state* $\mathbf{x} \in X$ *is an instance of* $\mathcal{Y} \subseteq \mathcal{M}$ *if and only if* $\phi(\mathbf{x}) = \mathcal{Y}$.

Remark: The value of Θ is not important in this anlysis, it is related with the measurability of molecules in reaction vessels.

The importance of algebraic chemistry theory is based on the following theorem, the proof is in [14]:

Theorem **1 (Fixed point is instance of an organisation)** *Consider a general reaction system whose reaction network is given by the algebraic chemistry* $\langle \mathcal{M}, \mathcal{R} \rangle$ *and whose dynamics is given by* $\dot{\mathbf{x}} = \mathbf{S}\mathbf{v}(\mathbf{x}) = \mathbf{f}(\mathbf{x})$. *Let* $\mathbf{x}' \in X$ *be a fixed point, that is,* $\mathbf{f}(\mathbf{x}') = 0$, *and let* $\phi : X \mapsto \mathcal{P}(\mathcal{M})$ *be an abstraction. Then* $\phi(\mathbf{x}')$ *is an organization.*

Remark: Usually the set of organisations is smaller than the power set of $\mathcal{M}$ (much more smaller in several cases). Thus the set of all organisations is of interest in the understanding of the long-term behaviour of the systems dynamics.
Remark: The other possible stationary regimes of the reaction vessel, such as oscillations or higher dimension limit sets, in several networks is formed by organisations only. But this matter is not well understood yet. See [41] for a preliminary analysis of this topic.
Remark: Theorem 1 states that every fixed point is an instance of an organisation, the

inverse property is not true. In fact, let the algebraic chemistry $\langle\{s\},\{s \rightarrow 2s\}\rangle$, it is easy to see that the only organisations are the empty set $\{\emptyset\}$, which represents no molecules in the reaction vessel, and the set $\{s\}$, which represent exponential growth, but none of them can be fixed points with concentration $\mathbf{x}_s > 0$. This work does not concern in the relation of organisations and stablility of solutions of the dynamical system derived from the algebraic chemistry. For works regarding this matter see [49, 41, 42].

Theorem 1 lead to an important conclusion: We do not need to simulate the whole dynamics[4] of an AC, because it is known that it evolves as movements between steady states through unstable transients, triggered by external perturbations. Then, it is sufficient to find first the organizations and then calculate the possible evolutions of the system studying how are the organisations connected.

Remark: In this work we are going to focus on the problem of obtain the organisational-structure. The problem of study the evolution of the network in time will not be explored in this work.

2.4 Organisational structure - Hasse diagram

In this section we are going to define concepts which will be useful to obtain the set of organisations, which in turn by Theorem 1 corresponds to a set of candidate to be fixed points in the dynamics of the system.

Definition **13 (x-structure)** *Given an AC $\langle\mathcal{M},\mathcal{R}\rangle$, we define the set of all sets which has the property x in $\langle\mathcal{M},\mathcal{R}\rangle$, where*

$$x = \textit{Closed, (Semi-)self-maintaining, (Semi-)organisation,}$$

as the x-structure of $\langle\mathcal{M},\mathcal{R}\rangle$.

From now on, we will call the organisational structure to the organisation-structure. We now are going to introduce some notions from Order theory [52]. We are going to define the notion of partially ordered set

Definition **14 (Partially ordered set)** *A* **partial order** *is a binary relation $\leq$ over a set P which is reflexive, antisymmetric, and transitive, i.e., for all a, b, and c in P, we have that:*

- *$a \leq a$ (reflexivity);*
- *if $a \leq b$ and $b \leq a$ then $a = b$ (antisymmetry);*
- *if $a \leq b$ and $b \leq c$ then $a \leq c$ (transitivity). A partially ordered set (poset) is a pair $(P, \leq)$*

---

[4] Obtain differential equations of system, give initial and boundary conditions and *run* the molecular dynamics.

Remark: It is easy to see that for every $X \subseteq \mathcal{P}(\mathcal{M})$, we have the pair $(X, \subseteq)$ is a poset.

LEMMA **1** *Let X be an x-structure of the AC $\langle \mathcal{M}, \mathcal{R} \rangle$. Then $(X, \subseteq)$ is a poset.*

DEFINITION **15 (Comparability of sets)** *For a, b distinct elements of a partially ordered set P, if $a \leq b$ or $b \leq a$, then a and b are* **comparable**. *Otherwise they are* **incomparable**.

DEFINITION **16 (Chain and antichain)** *If any two elements of a poset are comparable, the poset is called a* **chain** *and an* **antichain** *if any two elements are incomparable.*

Now we are going to define the notion of lattice, which is a special type of poset.

DEFINITION **17 (Lattice)** *Let the poset $(L, \leq)$ satisfying the following axioms:*

Ax1. Existence of binary joins: For any two elements a and b of L, the set $\{a, b\}$ has a **join** *(also known as the least upper bound or supremum).*

Ax2. Existence of binary meets: For any two elements a and b of L, the set $\{a, b\}$ has a **meet** *(also known as the greatest lower bound or infimum).*

The join and the meet will be denoted by $a \sqcup b$ and $a \sqcap b$ respectively. We say $(L, \leq, \sqcup, \sqcap)$ is a lattice if the poset $(L, \leq)$ satisfies the axioms Ax1 and Ax2.

It will be shown later that we can classify different kinds of AC in terms of the properies which its $x-$structures has. We are going to introduce the notion of Hasse diagrams, which are a useful tool to diagram posets.
The Hasse diagram is forming a drawing of the transitive reduction of the partial order. Concretely, for a partially ordered set $(S, \leq)$ one represents each x element of S as a vertex on the page and draws a line segment or curve that goes upward from x to y if $x < y$, and there is no z such that $x < z < y$ (here, $<$ is obtained from the relation $\leq$ by removing elements (x, x) for all x). See an example of a Hasse diagram in Figure 2.4. The Hasse diagram allow us to draw all x-structures that an AC can have. The Hasse diagram suggest a contructive way to develop an algorithm to find "x and y"-structures where x and y are properties, by checking upwards in the Hasse diagram if each set holds the property x (or y). Then, once obtained the x-structure (the y-structure), to obtain the "xy"-structure, we just need to check in the elements of the Hasse diagram of the obtained x-structure (y-structure) if the property y (or x) is hold. And thus we would obtain the "x and y"-structure.

2.5 Structural properties of AC

In this section we are going to show some ideas and results about the x-structure found in literature. We are going to review first the structural properties related to closed sets. Next, following [14], we will gradually build the definition of a consistent system, by presenting some useful intermediate concepts.

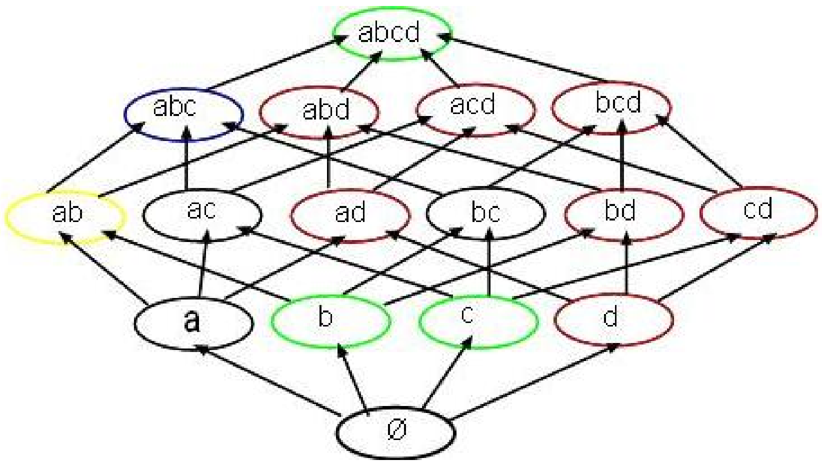

Figure 2.4: The Hasse diagram of Example 2.
Nodes are labelled by the set they represent and colored by the property the hold. If the color is black the set has not hold any property. Yellow means closed, orange for semi-self maintaining (no one in Example 2), red for self-maintaining, blue for semi-organisations and green for organisations.

Definition **18 (Generated closure)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC. Given $S \subseteq \mathcal{M}$, $G_{CL}(S)$ is the smallest closed set containing S. We say S generates the closed set $\mathcal{Y} = G_{CL}(S)$. Furthermore, we define the generated closure of union and intersection of sets of molecules $S_1, S_2 \subseteq \mathcal{M}$:*
$S_1 \sqcup_{CL} S_2 = G_{CL}(S_1 \cup S_2)$,
$S_1 \sqcap_{CL} S_2 = G_{CL}(S_1 \cap S_2)$.

Lemma **2** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $X, Y \subseteq \mathcal{M}$. Then $G_{CL}(G_{CL}(X)) = G_{CL}(X)$ and $X \subseteq Y$ implies $G_{CL}(X) \subseteq G_{CL}(Y)$.*

Lemma **3** *Given the AC $\langle \mathcal{M}, \mathcal{R} \rangle$, let $\mathcal{O} = \{\mathcal{Y}_1, ..., \mathcal{Y}_{|\mathcal{O}|}\}$ the set of all closed sets in the network (the closed-structure). We have $(\mathcal{O}, \sqcup_{CL}, \sqcap_{CL})$ is a lattice, where $\sqcup_{CL}$ and $\sqcap_{CL}$ are defined in Definition 18.*

Definition **19 (Semi-consistent AC)** *An algebraic chemistry $\langle \mathcal{M}, \mathcal{R} \rangle$ is* **semi-consistent** *iff given any pair of sets $\mathcal{X}$ and $\mathcal{Y}$ in $\mathcal{M}$ both (semi-)self-maintaining, their union is still (semi-)self-maintaining.*

For semi-consistent algebraic chemistries, is clear that for any couple $\mathcal{X}, \mathcal{Y}$ of (semi-)self-maintaining sets, we can obtain a (semi-)self-maintaining set $C = \mathcal{X} \cup \mathcal{Y}$. We now define the (unique in semi-consistent systems) generated (semi-)self-maintaining set:

Definition **20 (Generated semi-self-maintaining set)** *Given a semi-consistent algebraic chemistry $\langle \mathcal{M}, \mathcal{R} \rangle$, and given a set of molecules $\mathcal{C} \subseteq \mathcal{M}$, we define $G_{SSM}(\mathcal{C})$ as the biggest semi-self-maintaining set $\mathcal{S}$ contained in $\mathcal{C}$. We say that $\mathcal{C}$ generates the semi-self-maintaining set $\mathcal{S} = G_{SSM}(\mathcal{C})$. Furthermore, we define the generated semi-self-maintaining set of union and intersection of sets molecules:*
$\mathcal{X} \sqcup_{SSM} \mathcal{Y} = G_{SSM}(\mathcal{X} \cup \mathcal{Y})$,
$\mathcal{X} \sqcap_{SSM} \mathcal{Y} = G_{SSM}(\mathcal{X} \cap \mathcal{Y})$.

Definition **21 (Genreated self-maintaining set)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be a semi-consistent algebraic chemistry. Given a set of molecules $\mathcal{C} \subseteq \mathcal{M}$, we define $G_{SM}(\mathcal{C})$ as the biggest self-maintaining set $\mathcal{S}$ contained in $\mathcal{C}$. We say that $\mathcal{C}$ generates the self-maintaining set $\mathcal{S} = G_{SM}(\mathcal{C})$. Furthermore, we define the generated self-maintaining set of union and intersection of sets molecules:*
$\mathcal{X} \sqcup_{SM} \mathcal{Y} = G_{SM}(\mathcal{X} \cup \mathcal{Y})$,
$\mathcal{X} \sqcap_{SM} \mathcal{Y} = G_{SM}(\mathcal{X} \cap \mathcal{Y})$.

Remark: In semi-consistent AC, for every $\mathcal{C} \subseteq \mathcal{M}$ there always exist a unique biggest semi-self-maintaining set $\mathcal{C}^{ssm}$ and a unique a unique biggest self-maintaining set $\mathcal{C}^{sm}$. Note that $\mathcal{C}^{ssm}$ and $\mathcal{C}^{sm}$ are not necesarilly the same set.
Remark: In a semi-consistent algebraic chemistry, for any set of semi-self-maintaining sets $\mathcal{O} = \{C_1, ..., C_{|\mathcal{O}|}\} \subseteq \mathcal{P}(\mathcal{M})$ with the above defined operators, we have $(\mathcal{O}, \sqcup_{\mathcal{SSM}}, \sqcap_{\mathcal{SSM}})$ is a lattice. Analogously, if $C_1, ..., C_{|\mathcal{O}|}$ are self-maintaining, then $(\mathcal{O}, \sqcup_{\mathcal{SM}}, \sqcap_{\mathcal{SM}})$ is a lattice.

Lemma **4** *The following lemma is extracted from [14]. In semi-consistent systems the following statements are equivalent:*
The closure of a (semi-)self-maintaining set is (semi-)self-maintaining.
The (semi-)self-maintaining set generated by a closed set is closed.

Proof Let us suppose that the first statement is true, we shall prove the second: Let C be a closed set. Let $S = G_{SM}(C)$ be the self-maintaining set generated by C. Let D be the closure of S. Then $D \subseteq C$, because the closure of S cannot produce any molecule not contained in the closed set C. So $S \subseteq D \subseteq C$. But D is self-maintaining (because the closure of a self-maintaining set is closed). Yet S by construction is the biggest self-maintaining set in C. Thus $S = D$. So S is closed. Let us now suppose that the second statement is true, we shall prove the first: Let S be a self-maintaining set. Let $C = G_{CL}(S)$ be the closure of S. Let D be the self-maintaining set generated by C. Then $S \subseteq D$, because D is the biggest self-maintaining set, and if not we could take the union between S and D, which would still be a self-maintaining set, bigger than D. Since the self-maintaining set gener ated by a closed set is closed, then D is closed. So D is closed and self-maintaining. But since C is the closure of S, then C has to be the smallest closed set containing S. Thus $C \subseteq D$. So $C = D$. Thus C is closed and then the closure of S must return the smallest closed set that contains S. So C must be self-maintaining.

Lemma 4 states that in semi-consistent algebraic chemistries it will be generated the same (semi-)organisation by applying the closure generator and then the (semi-)self-maintaining generator to a set, than by applying the (semi-)self-maintaining generator and then the closure generator. From that useful property we can derive the notion of consistent algebraic chemistry.

Definition **22 (Consistent AC)** *A semi-consistent AC is called* **consistent** *if the closure of a (semi-)self-maintaining set is (semi-)self-maintaining.*

Definition **23 (Generated (semi-)organisation)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be a semi-consistent AC. Given* $C \subseteq \mathcal{M}$*, we define* $G_{SO}(\mathcal{C})$ *and* $G_O(\mathcal{C})$ *as* $G_{SSM}(G_{CL}(\mathcal{C}))$ *and* $G_{SM}(G_{CL}(\mathcal{C}))$ *respectively. We define the generated (semi-)organisation of union and intersection of sets molecules:*
$\mathcal{X} \sqcup_{SO} \mathcal{Y} = G_{SO}(\mathcal{X} \cup \mathcal{Y})$,
$\mathcal{X} \sqcap_{SM} \mathcal{Y} = G_{SO}(\mathcal{X} \cap \mathcal{Y})$.
$\mathcal{X} \sqcup_{O} \mathcal{Y} = G_{O}(\mathcal{X} \cup \mathcal{Y})$,
$\mathcal{X} \sqcap_{O} \mathcal{Y} = G_{O}(\mathcal{X} \cap \mathcal{Y})$.

Now we summarize the virtues of consistent reaction systems in the following theorem extracted from [14].

Theorem **2 (Organisations in consistent networks form a lattice)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be a consistent AC, given a set* $C \subseteq \mathcal{M}$ *of molecules, we can always uniquely generate a closure* $G_{CL}(C)$*, a semi-self-maintaining set* $G_{SSM}(C)$*, a semi-organisation* $G_{SO}(C) = G_{SSM}(G_{CL}(C))$ *, a self-maintaining set* $G_{SM}(C)$ *and an organisation* $G_O(C) = G_{SM}(G_{CL}(C))$*.*
Furthermore, the set of all organsations $\mathcal{O}$ *contained in* $\langle \mathcal{M}, \mathcal{R} \rangle$*, forms a lattice* $(\mathcal{O}, \sqcup_{\mathcal{O}}, \sqcap_{\mathcal{O}})$ *where* $\sqcup_O$ *and* $\sqcap_O$ *are defined in Definition 23.*

Theorem 2 state that in consistent reaction systems the organisational structure is a lattice w.r.t $\sqcup_O$ and $\sqcap_O$. Then, in consistent systems, the task of compute the organisational structure is simplified because it is not necessary to verify the organisation property to sets that are the join (or meet) of two or more organisations. One of the aims of this work is find other kind of structural properties which could simplify the computation of the organisational structure.

Chapter 3

A first algorithmic and complexity view to AC

The computational interest of Algebraic Chemistries is derived from Theorem 1. This theorem implies that in order to simulate the dynamics[1] of an AC, it is sufficient to find first the organisations, and then calculate the possible evolutions of the system by studying how are the organisations connected in the phase space. This is based on the fact that the system evolution can be understood as movements between steady states in the concentration space, through unstable transients, triggered by external perturbations. Then, to find good algorithms to obtain the set of all organisations (the organisational structure) is a central problem in Algebraic chemistries research.
Furthermore, Theorem 2 shows that in consistent reaction systems, the task of obtaining the organisational structure can be simplified by the use of some special union (join) and intersection (meet) operators. Then, the algorithms to verify and build organisations can be improved,for certain classes of networks, by structural results such as Theorem 2. In this chapter we will present the current known results on algorithms and computability of the properties of sets of molecules of a given AC defined in Chapter 2. Most of the algorithmic work in AC presented in this chapter is done in [10].

Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $X \subseteq \mathcal{M}$ be a set of molecules. We split the chapter in three main sections:

1. Verification: Verifying if X holds closure or (semi-)self-maintainance.
2. Generation: Deals with the generation of a closed set from X, the generation of semi-organisations above X when X is a semi-organisations, and generation of elementary organisations (see Section 3.3.2).
3. Organisational structure: Based in 1) and 2) we present the known methods to compute the organisational structure.

---

[1] Give the initial and boundary conditions to the the system of differential equations and *run* the molecular dynamics.

For this section let $|\mathcal{M}| = n$, $|\mathcal{R}| = k$. We also assume that the comparison of two integers has unitary cost.

Remark: The problem of studying the time-evolution of the network, *i.e* study the movement of organisations and transients in the phase space, will not be explored in this work. For a detailed analysis of these matters see [49].

3.1 Verification

Recalling Chapter 2 we have that a set $X \subseteq \mathcal{M}$ is a (semi-)organisation iff X is closed and (semi-)self-maintainang. Thus, verify both closure and (semi-)self-maintainance in X is the current known procedure to verify if X is an organisation. In this section we are going to study how to verify those properties.

3.1.1 Closure

The central question of this section is:

Problem **1** *Given a set $X \subseteq \mathcal{M}$, is X a closed set in $\langle \mathcal{M}, \mathcal{R} \rangle$?.*

To verify if X is closed we have to check that none of the reactions in $\mathcal{R}_X$ produces a molecule $m \notin X$. This means that for all reactions $R = A \to B \in \mathcal{R}_X$ it holds that m is present in B implies $m \in X$.

Lemma **5** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $X \subseteq \mathcal{M}$. To verify if X is closed set can be done in $|\mathcal{R}_X||X|$ steps.*

Proof It is enough to check that m is present in B implies $m \in X$ for every reaction $R = A \to B \in \mathcal{R}_X$.

$\square$

3.1.2 Semi-self-maintainance

Problem **2** *Given a set $X \subseteq \mathcal{M}$, is X a semi-self-maintaining set in $\langle \mathcal{M}, \mathcal{R} \rangle$?.*

To verify if X is semi-self-maintaining we have to check if every consumed molecule in $\mathcal{R}_X$ is produced in $\mathcal{R}_X$.

Lemma **6** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $X \subseteq \mathcal{M}$. To verify if X is semi-self-maintainaing can be done in $2|\mathcal{R}_X||X|$ steps.*

Proof We first check every reaction of $\mathcal{R}_X$ to build the set of consumed molecules in X; let us call such a set C. Set C is built in $|\mathcal{R}_X||X|$ steps. Then we check every reaction again to eliminate from C the produced molecules in X. This elimination also can be done in $|\mathcal{R}_X||X|$ time. At the end of this process, C is empty iff X is semi-self-maintaining.

□

3.1.3 Self-maintainance

Problem **3** *Given a set $X \subseteq \mathcal{M}$, is X a self-maintaining set in $\langle \mathcal{M}, \mathcal{R} \rangle$?.*

We are going to prove that self-maintainance verification corresponds to a Linear programming problem. We suppose that the reader is familiar with Linear programming definition. Definition and algorithms to solve Linear programming problems can be found in [53].

Lemma **7** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $X \subseteq \mathcal{M}$, verifying if X is self-maintaining is a Linear programming problem of $|\mathcal{R}_X|$ variables and $|\mathcal{R}_X| + |X|$ equations.*

Proof Let $\mathbf{S}_X$ be the stoichiometric matrix of the AC $\langle X, R_X \rangle$. We have X is self-maintainaing iff there exists a flux vector $\mathbf{v}$ constrained to X (this means that $\mathbf{v}_i > 0$ if i is s.t $R_i \in \mathcal{R}_X$ and else $\mathbf{v}_i = 0$) which holds:

$$\mathbf{S}\mathbf{v} \geq \mathbf{0}. \tag{3.1}$$

This is the first part (feasibility) of a Linear programming problem (no optimization of the solution is required). Furthermore, there are $|\mathcal{R}_X|$ variables and $|\mathcal{R}_X| + |X|$ equations.

□

Corollary **1** *To verify if a set $X \subseteq \mathcal{M}$ is an organisation is a Linear Programming problem of $|\mathcal{R}_X|$ variables and $|\mathcal{R}_X| + |X|$ equations.*

Remark: From the known LP bounds [53] we have that the complexity of the LP problem $\mathbf{A}\mathbf{x} \geq 0$ is $n^{3.5}L$ where n is the number of variables in $\mathbf{x}$ and L is the size of the representation of $\mathbf{A}$ in bits. Let l the binary representation of the largest mentioned number in $\mathcal{R}$. Then, for a given set X, the representation of $\mathbf{S}_X\mathbf{v} \geq 0$ under the condition $\mathbf{v} > 0$ is bounded by $|R_X| \cdot (|\mathcal{R}_X| + |X|)l$. Thus the complexity of verify the self-maintaining property for a set X is bounded by $l|R_X|^{4.5} \cdot \max(|X|, |\mathcal{R}_X|)$.

3.1.4 Generated closure

Lemma **8** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $X \subseteq \mathcal{M}$. The minimal cardinality set Z s.t $X \subseteq Z$ which is closed is unique.*

Proof Follows strightforward from Lemma 2.

□

In [10] it is explained how to build up a closed set Z starting from a set $X \subseteq Z$. It is based on add produced molecules until no new molecules can be produced.

Algorithm 1 Generated closure

```
Require: The initial set of molecules is $X$ and $\mathcal{R}_X$ is its firable reaction set;
Ensure: $G_{CL}(X)$
Let $Z = \emptyset, R_Z = \emptyset$
while $Z \neq X$ do
    $Y = \{$ produced molecules by $X\}$ and $Z = X \cup Y$
    $X = Z$
end while
return $(Z, \mathcal{R}_Z)$
```

Definition **24 (recursion loop)** *We are going to refer the execution of steps 2, 3, 4 in Algorithm 1 as a recursion loop.*

Remark: Note that in step 2 of Algorithm 1, in order to obtain Y, it is necessary to check only the reactions which were not checked in previous recursion loops.

Definition **25 (Amount of recursion loops)** *Let* n_{CL}^X *the amount of recursion loops required in Algorithm 1 to terminate with input* (X, R_X).

Remark: Note that $n_{CL}^X \leq |\mathcal{M}| - |X|$, because in each recursion step is required to add at least one molecule to the set Z. Furthermore, we have in step 3 of Algorithm 1 that $X \subset Z$ ($Y \neq \emptyset$), otherwise, the algorithm terminates.

Definition **26 (Symbols to bound the recursion loop)** *At the end of step 3 of the* i*-th recursion loop of Algorithm 1 with input* (X, R_X) *we define*

$$n_X^i = |Y|, \quad n_R^i = |\mathcal{R}_Z| - |\mathcal{R}_X|.$$

Furthermore we define

$$n_X = \max_{1 \leq i \leq n_{CL}^X} (n_X^i), \quad n_R = \max_{1 \leq i \leq n_{CL}^X} (n_R^i),$$

the maximum amount of molecules and reactions incorporated to Z *and to* R_Z *respectively in any recursion loop, where* n_{CL}^X *is defined in Definition 25.*

By Lemma 6 we have that obtaining the produced set of molecules requires linear time w.r.t $|\mathcal{R}_Z||Z|$. Then, in order to estimate the time required by Algorithm 1 to terminate is necessary to consider the way in which X is is growing until reach its closure.

Theorem **3 (Time to compute generated closure)** *Let* $i = 1, ..., n_{CL}^X$ *identifying the recursion loop of Algorithm 1. Let* n_X^i *and* n_R^i *defined in Definition 26. Then Algorithm 1 can be done in* $2|X||\mathcal{R}_X| + 2\sum_{i=1}^{n_{CL}^X} |n_X^i||n_R^i|$ *steps.*

Proof In the first recursion loop of the procedure, this means for $i = 1$, by Lemma 6 we have that obtain the produced set of molecules in X requires $|\mathcal{R}_X||X|$ steps. For $i > 1$, by Lemma 6 we have that obtain the produced set of molecules in X requires $|n_R^i||n_X^i|$ steps.

$\square$

Corollary **2** *Algorithm 1 can be done in* $2(|X||\mathcal{R}_X| + n_{CL}^X n_X n_R)$ *steps.*

3.2 Generating semi-organisations

The only result which is known for AC on generating semi-self-maintaing sets is that, in semi-consistent systems, for every closed set X there exist a set $X' \subseteq X$ which is semi-self-maintaining. In general systems we have no such result. In [10] it is presented an algorithm to generate, given a set X, all the semi-organisations which are *directly above*[2] a semiorganisation X. We are going to show that this algorithm has two defects:

1. It can be very expensive even returning an empty set.
2. It can return semi-organisations which are not directly above X.

We are going to explain the algorithm and then present the defects.

Algorithm 2 Generating semi-organisations above a set

```
Require: An AC $\langle \mathcal{M}, \mathcal{R} \rangle$and the semiorganisation $X$.
Ensure: The set $A_X$ of semi-organisations directly above $X$.
$A_X = \emptyset$
for all $s \in \mathcal{M} - X$ do
    Let $A_X = A_X \cup \text{SosDirAb}(X, \{s\}, \langle \mathcal{M}, \mathcal{R} \rangle)$
end for
```

If a set X is closed, but not semi-self-maintaining, then there is a set of molecules $X' \subseteq X$ which are consumed but not produced. The idea of the algorithm is based in that we can look at the AC trying to find all the sets of molecules $Y_1, ..., Y_{n_{X'}}$ which generates X' and then for each set Y_i, $i = 1, ..., n_{X'}$, we generate a new set $X_i = Y_i \cup X$ and check if the generated closure of Y_i is a semi-organisation. If $G_{CL}(X_i)$ is a semi-organisation, then X_i is added to

---

[2] In the Hasse diagram.

Algorithm 3 SosDirAb

```
Require: An AC $\langle \mathcal{M}, \mathcal{R} \rangle$, a set of molecules $X$ and a set of species species s.t $s \in$ species implies $s \notin X$.
Ensure: The set result formed by all the semi-organisations directly above $X$ that contain the set species in it.
result = $\emptyset$, closure = $G_{CL}(X \cup \textbf{species})$
if closure is semi-self-maintaining then
    result = result $\cup$ closure
else
    Let SptoProd = $\{s \in \textbf{closure}$ s.t $s$ is consumed but not produced$\}$
    Let Pset = $\{Y \subseteq \mathcal{M}$ s.t $Y$ produces SptoProd$\}$
    for all $Y \in$ Pset do
        result = result $\cup$ SosDirAb($X$, species $\cup\, Y$, $\langle \mathcal{M}, \mathcal{R} \rangle$)
    end for
end if
```

the set of semi-organisations directly above X, if it is not, then there is a set of molecules $X_i' \subseteq G_{CL}(X_i)$ which are consumed but not produced, and the process has to be repeated.

Now we are going to show that Algorithm 2 with input $\langle \mathcal{M}, \mathcal{R} \rangle$ and $X \subset \mathcal{M}$ can take exponential time w.r.t $|\langle \mathcal{M}, \mathcal{R} \rangle|$, and returns an empty set. Let the generic class of AC $Q(n,k) = \langle \mathcal{M}, \mathcal{R} \rangle$ defined by

$$\begin{aligned} \mathcal{M} &= \{d, a_1^1, ...a_n^1, a_1^2, ...a_n^2, ...a_1^k, ...a_n^k\}, \\ \mathcal{R} &= \{a_i^1 \to d, \text{ for } i = 1, ..., n\} \cup \{a_k^i \to a_l^j, \text{ for } i > j \text{ and } k, l = 1, ..., n\}. \end{aligned} \tag{3.2}$$

Let $X = \{d\}$. Algorithm 2 has to check all the subsets of $\mathcal{M}$ which contains $G_{CL}(X) = \{d\}$. Let $\bar{X} = X - \{d\}$. This means Algorithm 2 has to check all the subsets which are in $\bar{X}$. The amount of such subsets is exponential w.r.t $|\bar{X}| = |X| - 1$. The amount of reactions in $\langle \mathcal{M}, \mathcal{R} \rangle$ is bounded by the square of $|\mathcal{M}|$ because for every pair of molecules in $\bar{X}$ we have one reaction in the network and d is produced only by a_k^1, $k = 1, ..., n$. Then Algorithm 2 takes exponential time w.r.t $|\langle \mathcal{M}, \mathcal{R} \rangle|$, but there is no semi-organisation directly above $\{d\}$.

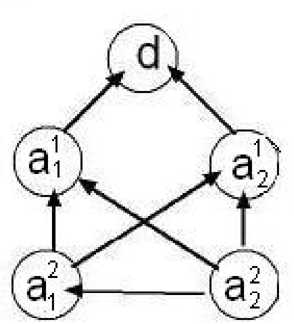

Figure 3.1: Network $Q(n,k)$ defined in equation (3.2) when $n = k = 2$. Algorithm 2 takes exponential time w.r.t the size of $Q(n,k)$ because it has to check each set which contains $\{d\}$, all those sets are closed, but none of them is semi-self-maintaining, thus there is no semi-organisation above $\{d\}$.

Furthermore considering the AC $\langle\{a,b\},\{a \to a+b\}\rangle$ and $X = \{\emptyset\}$, we obtain that Algorithm 2 returns $\{a\},\{a,b\}$, because in step 2 it has to compute the closure of $\{b\}$, which is $\{b\}$, and the closure of $\{a\}$ which is $\{a,b\}$. As both sets are semi-self-maintaining and closed, Algorithm 2 returns both sets as semi-organisations directly above $\{\emptyset\}$. But $\{a,b\}$ is not directly above to the empty set in the Hasse diagram.

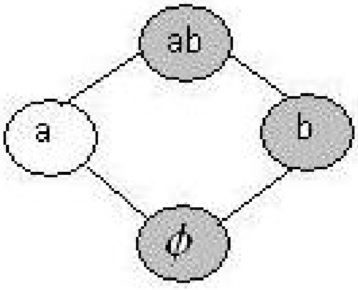

Figure 3.2: Algorithm 2, with the network $\langle\{a,b\},\{a \to a+b\}\rangle$ and the empty set as input returns the grey sets in the figure. The set $\{a,b\}$ is part of the output. But $\{a,b\}$ is not directly above to the empty set in the Hasse diagram.

3.3 Structural properties

In this section we are going to review some properties which allow us to understand better how the organisations are formed. This information is useful in the developing of algorithms to compute the organisational structure. We are going to introduce the fundamental concepts and by using them we are going to present the underlying algorithms of both approaches. Finally we are going to explain how to proceed with those algorithms in each case to build a procedure which computes the organisational structure.

3.3.1 Connectivity and reactivity

Definition **27 (Directly connected)** *Let $\langle\mathcal{M},\mathcal{R}\rangle$ be an AC. Two species s_o and s_p in $\mathcal{O} \subseteq \mathcal{M}$ are directly connected in $\langle\mathcal{O},\mathcal{R}_\mathcal{O}\rangle$ iff there exist a reaction $R = A \to B \in \mathcal{R}_\mathcal{O}$ such that $\{s_o,s_p\} \subseteq A \cup B$.*

Definition **28 (Connected molecules)** *Let $\langle\mathcal{M},\mathcal{R}\rangle$ be an AC. Two molecular species s_i and s_j in $\mathcal{O} \subseteq \mathcal{M}$ are connected in $\langle\mathcal{O},\mathcal{R}_\mathcal{O}\rangle$ if there exist a sequence of species $s_0,...,s_p \in \mathcal{O}$ such that $s_i = s_0$, s_k and s_{k+1} are directly connected in $\langle\mathcal{O},\mathcal{R}_\mathcal{O}\rangle$ for all $k = 0,...,p-1$ and $s_p = s_j$.*

Definition **29 (Connected set)** *Let $\langle\mathcal{M},\mathcal{R}\rangle$ be an AC. A set $\mathcal{O} \subseteq \mathcal{M}$ is connected if every pair of molecules is connected in $\mathcal{O}$.*

Definition **30 (Reactive set)** *Let $\langle\mathcal{M},\mathcal{R}\rangle$ be an AC. A set $C \subseteq \mathcal{M}$ is reactive iff for all $s \in C$ there exists at least one reaction $R = A \to B \in \mathcal{R}_C$ such that $s \in A \cup B$.*

Note that the set $G_{CL}(C) - C$ has only reactive molecules which are connected to C.

Lemma **9** *The generated closure of a connected set is connected, and the generated closure of a reactive set is reactive.*

Remark: Note that every connected set is a reactive set, but the inverse is not true, because a reactive set may contain several connected subnetworks that are disjoint.

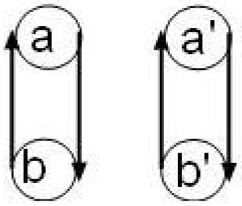

Figure 3.3: The organisation $\{a, b, a', b'\}$ is reactive but not connected, because is splitted in two connected organisations $\{a, b\}$ and $\{a', b'\}$, which are not connected among themselves.

Now we are going to show that from the set of connected organisations it is possible to obtain the organisational structure. The proof is extracted from [10].

Theorem **4 (Connected organisations and organisational structure)** *The organisational structure of a network $\langle \mathcal{M}, \mathcal{R} \rangle$ can be obtained from the connected organisational structure, i.e from the set of all connected organisations contained in $\mathcal{M}$.*

Proof We are going to prove that the following four steps precedure is sufficient to create all organisations.

1. Compute the set of connected organisations O_{init} contained in $\mathcal{M}$.
2. Let $\mathcal{R}' \subseteq \mathcal{R}$ be the set of all inflow reactions. Compute the set of connected organisations for the modified network $O_{\text{withoutinput}}$.
3. Let the set of basis organisations $O_{\text{basis}} = O_{\text{init}} \cup O_{\text{withoutinput}}$.
4. Make set unions of all possible combinations of organisations from O_{basis} such that exactly one organisation from O_{init} is contained in every combination. (If a combination of organisations from O_{init} is already an organisation, it is already an element of O_{init}). Test the species set of each combination for the closure and self-maintenance property. With $|O_{\text{init}}| = m$ and $|O_{\text{withoutinput}}| = n$, there are

$$m \sum_{i=0}^{n} \binom{n}{i} = m \cdot 2^n \tag{3.3}$$

species sets to be tested.

We are going to prove that each organisation is a combination of the basis organisations defined in step 3. For this purpose, networks with and without input species will be discussed separately:

Networks without input: If the network has no input species, the basis organisations are exactly the connected organisations. Taking an organisation O, we find that it is either connected or not. In the former case, it is a basis organisation. In the later case, it consists of two or more parts that are connected, but not connected to each other part. When inspecting each isolated part separately, we find that each part is closed and self-maintaining. In other words, each part is an organisation. Even more, each part is a connected organisation and hence a basis organisation. Therefore, the unconnected organisation O is equal to the set union of these basis organisations.

Networks with input: Again, taking any organisation O of the network, we find that it is either connected or not. If it is connected, it is already a basis organisation. If it is not connected, we again inspect the isolated parts of the organisation. Like in the case without input species, all parts are closed, self-maintaining, and connected. Some parts contain input species and some others not. Recall that in the presence of input species, all input species are present in all organisations. Hence, the union of all isolated parts that contain at least one input species will be an organisation (and contained in O_{init}). Parts without input species are only organisations in the absence of input species, and hence contained in $O_{\text{withoutinput}}$. We find that all isolated parts of organisation O can be associated to basis organisations in O_{init} and $O_{\text{withoutinput}}$. Consequently, O is equal the set union of these basis organisations. We conclude that all organisations are created using the described procedure.

□

Corollary **3** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC. Obtaining the connected organisational structure of $\langle \mathcal{M}, \mathcal{R} \rangle$ is output sensitive. Furthermore, the Algorithm is fixed parameter tractable with respect to the amount of organisations in the modified network $\langle \mathcal{M}, \mathcal{R}' \rangle$ where $\mathcal{R}' \subseteq \mathcal{R}$ is the result from the substraction of the inflow reactions to $\mathcal{R}$.*

Proof Follows directly from equation (3.3).

□

Corollary 3 state that the time to computate the connected organisational structure depends on the amount of organisations that will be obtained as output. The fixed parameter tractability means that taking the class of networks in which the amount of connected organisations of the modified AC $\langle \mathcal{M}, \mathcal{R}' \rangle$ is a constant number (n in equation (3.3)), the connected organisational structure of this class of networks can be obtained in polynomial time w.r.t to the amount of connected organisations (m in equation (3.3)).

Remark: It must be noted that the basis organisations O_{basis} do not form a basis for all organisations that are minimal. Consider the reaction network consisting of two species and one reaction $\langle \{a, b\}, \{a + b \to 2a + 2b\} \rangle$. The organisational structure in this case is $\{\{\emptyset\}, \{a\}, \{b\}, \{a, b\}\}$. All four organisations are connected and therefore basis organisations. However, $\{a, b\} = \{a\} \cup \{b\}$ and hence would not be required in a minimal basis.

3.3.2 Elementary organisations/flux modes

When viewing organisations as sets of reactions, the elementary organisations are those which cannot be represented as the union of other organisations (sets of reactions).

Definition **31 (Elementary organisation)** *Given a set of organisations $\mathcal{O}$, an organisation $O \in \mathcal{O}$ with its corresponding set of reactions $\mathcal{R}_O = \{A \to B \in \mathcal{R} \text{ s.t } A, B \in \mathcal{P}_M(O)\}$ is an elementary organisation, if there exists no subset of organisations $T \subseteq \mathcal{O} - \{O\}$ such that $\mathcal{R}_O = \bigcup_{P \in T} \mathcal{R}_P$.*

We now are going to present a well known concept in the biochemical network literature.

Definition **32 (Elementary mode)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $\mathbf{S}$ the stoichiometric matrix of this reaction network. Let $\mathbf{v} \geq \mathbf{0}$ such that $\mathbf{Sv} = \mathbf{0}$. We say that $\mathbf{v}$ is a mode of $\langle \mathcal{M}, \mathcal{R} \rangle$. We say that $\mathbf{v}$ is an elementary mode if it is a mode and it is not possible to decompose it as the sum of two non empty modes.*

Definition **33 (Extreme rays)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ and $Q' = \langle \mathcal{M}, \mathcal{R} \cup \{m_1 \to \emptyset, ..., m_n \to \emptyset\} \rangle$ be two AC. We say the set of elementary modes of Q' is the set of extreme rays of $\langle \mathcal{M}, \mathcal{R} \rangle$.*

The set of extreme rays is composed by all the flux vectors which fulfills the self-maintainance condition and cannot be decomposed into other extreme rays. Note that in Definition 33 if we take the projection of the first $k = |\mathcal{R}|$ coordinates of the elementary modes of Q', we will obtain all the elementary modes of $\langle \mathcal{M}, \mathcal{R} \rangle$, but also a basis of the flux vectors which fulfills $\mathbf{Sv} > \mathbf{0}$, where $\mathbf{S}$ is the stoichiomteric matrix associated to $\langle \mathcal{M}, \mathcal{R} \rangle$.

Remark: In [20] it is proved that the maximal amount of elementary modes grows exponentially with the size of the network. Research in computation of elementary modes is out of the scope of this work. For a more detailed treatment on computation of elementary modes see [17, 47, 2].

We now are going to present an algorithm for obtaining all the elementary organisations from the set of extreme rays (Algorithm 4). First, the closure of the reaction set v^{set}, respectively $M(v^{set})$, is computed. This is done by taking the species set $M(v^{set})$ and iteratively adding all species to the set that can be created by reactions of the network from the species set. The reaction set v^{set}_{cl} contains all reactions that can take place in the generated closed set of species. If this reaction set is identical to v^{set}, the species set $M(v^{set})$ is an organisation. The reaction set v^{set}_{cl} contains more reactions than v^{set} if either species were added, or $M(v^{set})$ is closed but v^{set} does not contain all reactions that are possible in this set. One of such reactions is taken and all the reaction sets $v^{set}_B \in V^{set}_B$ that contain this reaction are consecutively combined with the original reaction set v^{set} and the function is called again recursively. As the initial reaction set v^{set} and the extreme ray reaction sets v^{set}_B correspond to flux vectors fulfilling the self-maintenance condition, also a flux vector $\mathbf{v_u}$ fulfilling the self-maintenance property exists for the union $v^{set}_u = v^{set} \cup v^{set}_B$. Hence, all reaction sets that are considered in the recursive function calls are associated with self-maintaining flux vectors.

Algorithm 4 OrgsAb

```
Require: Reaction network $\langle \mathcal{M}, \mathcal{R} \rangle$, the set of reaction sets $V_B^{set}$, each set in $V_B^{set}$ corresponds to the reactions with positive coefficients in each extreme ray of $\langle \mathcal{M}, \mathcal{R} \rangle$and a set $v^{set} \subseteq V_B^{set}$.
Ensure: Set result of elementary organisations s.t $O \in$ result implies $v^{set} \subseteq \mathcal{R}_O$.
Let result $= \emptyset$,
$M(v^{set}) = \{m \in \mathcal{M}$ s.t $m$ is consumed or produced by some $R \in v^{set}\}$,
$v_{cl}^{set} = \{R \in \mathcal{R}$ s.t $R$ is firable by $G_{CL}(M(v^{set}))\}$
if $v_{cl}^{set} = v^{set}$ then
    result $=$ result $\cup M(v^{set})$
else
    Select a reaction $r \in v_{cl}^{set} - v^{set}$
    for all $v_B^{set} \in V_B^{set}$ s.t $r \in v_B^{set}$ do
        result $=$ result $\cup$ OrgsAb$(v^{set} \cup v_B^{set})$.
    end for
end if
```

3.4 Some useful examples

In this section there will be presented five examples showing some exemplar (semi-)organisational structures. Each example reflect different features that the (semi-)organisational structure can exhibit, as well as different relations among the semi-organisational structure and the organisational structure.

Example 4. Consider the AC $\langle \mathcal{M} = \{a_1, ..., a_n\}, \{\emptyset\} \rangle$. In this case, the (semi-)organisational structure is $\mathcal{P}_M(\mathcal{M})$. This means the amount of organisations can be exponential w.r.t the amount of molecules in the network. But all of them are non-reactive.

Example 5. Consider the AC $\langle \mathcal{M} = \{a_1, ..., a_n\}, \{R_1, ..., R_{n-1}\} \rangle$ where $R_i = 2a_i \rightarrow a_i$, for $i = 1, ..., n$. We have that the semi-organisational structure is equal to $\mathcal{P}_M(\mathcal{M})$, but the organisational structure is only the empty set. Note that no molecule is connected to each other. But all molecules are reactive.

Example 6. Consider the AC $\langle \mathcal{M} = \{a_1, ..., a_n\}, \{R_{ij}, i = 1, ..., n \text{ and } j \geq i\} \rangle$ where $R_{ij} = a_i + a_j \rightarrow 2a_i + 2a_j$. The semi-organisational structure and the organisational structure are both $\mathcal{P}_M(\mathcal{M})$. But in this case every organisation is connected.

Example 7. Consider the AC $\langle \mathcal{M} = \{a, b\}, \{a + b \rightarrow \{\emptyset\}\} \rangle$. The organisational structure is $\{\{\emptyset\}, \{a\}, \{b\}\}$. Every organisation is elementary, but no union of elementary organisations generates an organisation.

Example 8. Consider the AC $\langle \mathcal{M}, \mathcal{R} \rangle$where

$$\mathcal{M} = \{a, b, c, d\},$$
$$\mathcal{R} = \{a + b \to 2a + 2b, d \to c, c \to d, a + b + c \to \emptyset\}.$$

In this example the semi-organisational structure is

$$S = \{\{\emptyset\}, \{a\}, \{b\}, \{a, b\}, \{c, d\}, \{a, c, d\}, \{b, c, d\}, \{a, b, c, d\}\},$$

the organisational structure is $O = S - \{a, b, c, d\}$ and the elementary organisations are $\{\{\emptyset\}, \{a\}, \{b\}, \{c, d\}\}$. This means that in this case the generated closure of union of elementary organisations is an organisation, but the generated closure of union of an organisations in general does not generate an organisation.

Remark: The examples above show that the properties defined in previous sections do not give much information **in general** about the total size or shape of the (semi-)organisational structure, but as we are going to see in the next section, those concepts will be useful to develop algorithms to calculate the organisational strucure.

3.5 Building the organisational structure

Using definitions of previous sections we are going to explain the known algorithms to compute the organisational structure. In this section there are presented two algorithms to compute the organisational structure for a given reaction network: The constructive and the flux-based approach. Both algorithms compute the organisational structure in an exhaustive manner, this means they compute the whole organisational structure. However, due to the exponential nature of the problem, *i.e* the existence of networks in which any species subset is an organisation, these algorithms do not always finish in reasonable time [10]. In such cases, an heuristic approach can be used to compute at least a significant subset of the organisational structure (see the remark at the end of the chapter). While the constructive approach compute the organisations in a bottom-up manner starting from the smallest organisation, the flux-based approach combines flux vectors which verifies the self-maintenance until organisations are found. Testing both algorithms on several network models shows that neither of them is superior in all cases [10].

3.5.1 Constructive approach

The constructive approach builds first the semi-organisational structure from bottom to top in the Hasse diagram (this means starting from the empty set) and once having the semi-organisational structure the algorithm verifies whose of those semi-organisations are self-maintaining and thus organisations. In order to build the semi-organisational structure Algorithm 2 is used. Recalling Theorem 4 it is enough to build the connected semi-organisational structure. Thus we are going to change line 2 of Algorithm 2, now it looks:

Modification of Algorithm 2

```
...
2: For all $s \in \mathcal{M} - X$ connected to $X$ do
...
```

With these slight variation Algorithm 2 obtain only the connected semi-organisational structure. As we saw in Section 3.1.3, verification of self-maintainance is a Linear Programming problem, then for every set of the semi-organisational structure it has to be solved a Linear Programming problem to verify its self-maintainance. The set of self-maintaining semi-organisations verified in step *3* of Procedure 1 is the organisational structure.

PROCEDURE **1 (Constructive approach)**

1. **Beginning:** $\langle \mathcal{M}, \mathcal{R} \rangle$ *and* $\mathcal{O}$ *the organisational structure are intialized empty.*
2. *Let* $\mathcal{O}$ *the connected semi-organisational structure. This is obtained by using Algorithm 2.*
3. *Let* $\mathcal{O}'$ *the set of all self-maintainaing sets contained in* $\mathcal{O}$*. This set is obtained by solving the associated Linear programming problem (see Lemma 7).*
4. *Using* $\mathcal{O}'$ *obtain the organisational structure by the method stated in Theorem 4.*

As Procedure 1 is based in Algorithm 2, then it has the same defects than Algorithm 2. This means the organisational structure can take exponential time w.r.t $|\langle \mathcal{M}, \mathcal{R} \rangle|$ even returning an empty organisational structure.

3.5.2 Flux based approach

The flux based approach takes a somehow opposite strategy as the constructive approach of Section 3.5.1 by combining self-maintaining flux vectors until closed sets are discovered. This algorithm is useful for small networks, because it is proved that in general the amount of self-maintaining flux vectors grows exponentially with the size of the network [20]. While the first approach operates on species, the flux-based approach operates on reactions. First, the flux based approach computes elementary organisations (step *1*), which are combined in step *2* to obtain reactive organisations, that is, organisations where each species participates at least in one reaction within the organisation. Organisations containing species that do not react are determined in step *3*. Starting with the condition of self-maintenance, methods from convex analysis can be employed. Given a reaction network $\langle \mathcal{M}, \mathcal{R} \rangle$ and its $n \times k$ stoichiometric matrix $\mathbf{S}$, a flux vector $\mathbf{v} \in R^k_+$ fulfilling the self-maintenance condition must be found to show that a species set is self-maintaining. All such flux vectors lie in a convex polyhedral cone P in the k-dimensional flux space R^k_+, originated in the point of origin.

The cone is defined by the $k + n$ inequalities $\mathbf{v} \geq \mathbf{0}$ and $\mathbf{Sv} \geq \mathbf{0}$. These constraints can be transformed into a matrix $\mathbf{A}$ representing the set of spanning vectors V_B or extreme rays of P [17]. These extreme rays are the elementary modes defined in Definition 32. Each point within P can be written as a linear combination of these elementary modes. Thus, we can compute organisations by searching for combinations of elementary modes whose corresponding set of species fulfills the closure condition.

PROCEDURE **2 (Flux based approach)**

1. **Beginning:** *The AC* $\langle \mathcal{M}, \mathcal{R} \rangle$*, and the set of extreme rays* V_B^{set} *correspondig to* $\langle \mathcal{M}, \mathcal{R} \rangle$*.*
2. *Obtain the elementary organisations: This is done by calling Algorithm 4 for each reaction set* $v_B^{set} \in V_B^{set}$ *(see Algorithm 4).*
3. *Computing reactive organisations: Elementary organisations are combined to determine all reactive organisations. This is done by taking all possible combinations of two elementary organisations and calling Algorithm 4 for the union of their reaction sets. Every newly discovered organisation, has to be combined again with each of the elementary organisations, i.e Algorithm 4 must be called again for the reaction set unions, until this process computes all the reactive organisations.*
4. *Computing all organisations: The organisations we have obtained so far all possess a different set of reactions. Consequently, the final step consists in searching the organisations which have the same set of reactions as already discovered ones, but containing different species sets. Hence, we need to determine for all discovered organisations all species sets, that can be added to the organisation without changing its set of reactions. This can be done by simply inspecting the reaction list.*

Remark: There is also an heuristic algorithm which is able to calculate a subset of the organisational structure. We are not going to study that algorithm because the size of the set of organisations which this algorithm calculates, compared with the whole organisational structure is not well understood. This implies there is no notion of how good is this algorithm. Furthermore in biochemistry it is not too useful to obtain a subset of the organisational structure because it is not known a priori which organisations are more important in the dynamics, so we have to have all the candidates to be fixed point (organisations) to study the dynamics properly.

Part II

Study of AC from a discrete point of view

Introduction

To better present this second part, let us summarize the problems found in the first part of this thesis:

- **Obtain the organisational structure is the central task:** The organisational structure is the most important set to be obtained for a reaction network. Its calculation cannot be done by a brute force algorithm, because it should be verified the self-maintainance for every subset of species, and the power set grows exponentially with the amount of molecules.
- **Known algorithms to compute organisations are based in structural properties:** The constructive approach (see Section 3.5.1) is based essentially in the connectivity and reactivity of molecules in the network. The flux-based approach (see Section 3.5.2) is based essentially in the set of elementary organisations and in the extreme rays. All these properties intend to simplify the understanding of how the organisations are formed or how the organisations can be combined.
- **There are hard cases for all known approaches:** There are simple cases in which the organisational structure is extremely hard to compute for every algorithm. For some networks the constructive approach is better than the flux based approach and for some other networks is the opposite [10]. But there is not a good enough understanding about when one approach is better than the other.

The three points above reflect the inmature state in which the computational aspects of algebraic chemistries are. There seems to be a strong potential in Theorem 1, which links fixed points of a chemical system and the organisations in its underlying reaction network, but the current work does not give a deeper enough understanding of the underlying process of organisation formation. The topological properties of the networks are almost not considered in the analysis (only connection and reactivity), and the treatment of other properties such as elementary organisations have not been sistematically studied. For example, there is no definition of topological properties which could *a priori* suggest which approach to compute the organisational structure would be faster. AC theory was developed to deal with huge biological networks in which is hard to study the dynamics with the classic dynamical systems tools. Then the computational properties which can be defined to understand better the underlying structure of an organisation will be very useful to study the time evolution of the dynamical system.
The second part of this thesis develops the following topics:

- Frame the study of reaction networks in others well known computer science formalisms.
- Study the organisations in detail to reveal their underlying structure.
- Provide a framework to develop algorithms which takes in consideration as much as possible the structure of the network.

In Chapter 4, a discrete version of Algebraic chemistries is defined, and we discuss the relevant aspects of this approach. In Chapter 5 we present two discrete event formalisms (Petri Nets and Vector Addition Systems), and an equivalence between AC and those formalisms is stated. In Chapter 6 we overview the relation of some important properties defined in Petri Nets and Vector Addition Systems with respect to the organisations in Algebraic chemistries. In Chapter 7 we develop an algorithm to identify organisations and estimate its complexity. Based on the insights arosed from the discrete analysis, in Chapter 8 we present a new mathematical framework to understand the process of organisation formation, which permits to identify the difficulties when it is verified if a set of molecules is (or is not) an organisation. Finally, in Chapter 9 we present some structural results about the organisational structure, *i.e*, develop a separation of classes of networks in terms of the computational properties that the organisations of each class of network possesses.

Chapter 4

Discretizing AC

1 Sets versus Multisets in AC

The purpose of AC is to predict properties of (bio-)chemical compounds, given the reaction rules between molecules. In AC formalism, we study properties of sets (instead of multisets) of molecules. Hence these properties do not depend on the amount of molecules of each type in the reaction vessel, but on the *presence* of the molecular species in the reaction vessel and on the possible *firing rates* between the reactions. This is because in a chemical system it is implicit the fact that the amount of each type of molecules occurs in a higher order of magnitude (Avogadro number) than the amount of molecules required in the reactions. In practical terms, if some molecular specie m is present in the reaction vessel, then we have as much molecules of type m as we need to fire reactions.

We are going to explore the AC $\langle \mathcal{M}, \mathcal{R} \rangle$ of Section 2.1, with $\mathcal{M} = \{m_1, ..., m_n\}$ and $\mathcal{R} = \{R_1, ..., R_k\}$, from a discrete point of view, that is, considering a discrete amount of molecules in the reaction vessel. The goal is to capture the algorithmic essence of the AC formalism in a discrete schema. We need some definitions.
The reaction vessel, which is determined by a set $\mathcal{Y} \subseteq \mathcal{M}$ of molecules in the AC, in the discrete schema will be described by a multiset $y \in \mathcal{P}_M(\mathcal{Y})$, where $\mathcal{A}(y, m_i)$ will denote the amount of molecules of specie m_i in the multiset y.

Definition **34 (Discrete algebraic chemistry)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC. Given a multiset $y \in \mathcal{P}_M(\mathcal{M})$, the triple $\langle \mathcal{M}, \mathcal{R}, y \rangle$ is a Discrete Algebraic Chemistry (in short DAC). We say y is the discrete reaction vessel.*

In AC, the reaction process is determined by the application of the flux vector $\mathbf{v}$ on the stoichiometric matrix $\mathbf{S}$, leading to a production rate vector $\mathbf{f}$. Note that if we want to consider the reaction R_i in the reaction process of $\langle \mathcal{M}, \mathcal{R} \rangle$, we require that *if reactants of R_i are present in the reaction vessel, then R_i can be fulfilled*, formally:

Definition **35 (Causal reaction w.r.t a flux vector)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $\mathcal{Y} \subseteq \mathcal{M}$ and $\mathbf{v} = (v_1, ..., v_m)$ a flux vector, let $i \in \{1, ..., m\}$. The reaction $R_i = k_1^i m_1 + ... + k_n^i m_n \rightarrow$*

$j_1^i m_1 + ... + j_n^i m_n$ is causal w.r.t $\mathbf{v}$ and $\mathcal{Y}$ in $\langle \mathcal{M}, \mathcal{R} \rangle$, if and only if $v_i > 0$ and $k_l^i > 0$ implies $m_l \in \mathcal{Y}$ for all $l = 1, \ldots, n$. Furthermore $\mathbf{v}$ is **causal** *w.r.t $\mathcal{Y}$ and $\langle \mathcal{M}, \mathcal{R} \rangle$ if and only if for $i = 1, ..., m$ we haveR_i is causal w.r.t $\mathbf{v}$ and $\mathcal{Y}$ in $\langle \mathcal{M}, \mathcal{R} \rangle$.*

Note also that in the reaction process of the AC it is not considered the order in which reactions are specified by $\mathbf{v}$. This is because in chemical systems the reaction process is considered *per se* a parallel process, that is, every reaction consumes its own reactants and the reactions occur simultaneously. From a discrete point of view, the reactions occur in a serial process, that is, the reactions are fulfilled one by one in some order and the products of every reaction are incorporated as possible reactants to the next reactions. Then, we cannot (*a priori*) capture the reaction process by a flux vector as in AC, because in spite of all the reactants of R_i that could be present in $\mathcal{Y}_M$, it is possible that, for some molecular specie, there are not enough reactants as required to fire R_i. Then, the order of reactions plays an important role in the discrete reaction process. We are going to define the reactions as a serial process, but we are going to show that for some multisets of molecules the order in which reactions take place does not change the resulting multiset of molecules.
The flux vector of the AC formalism, is considered as a sequence of reactions in DAC, such that the reactions are applied to y one by one:

Definition **36 (Sequence of reactions)** *Let $\langle \mathcal{M}, \mathcal{R}, y \rangle$ be a DAC, $L_{\mathcal{R}} = (R_1 + ... + R_k)^*$ a regular language contructed with the elements of $\mathcal{R}$, $\mathbf{v} = (v_1, ..., v_k)$ a flux vector and $R_{\mathbf{v}} = R_1^{v_1} \cdots R_k^{v_k} \in L_{\mathcal{R}}$. P is a sequence of reactions of $\mathbf{v}$ if and only if P is a permutation of $R_{\mathbf{v}}$.*

Definition **37 (Apply a reaction)** *Let $Q^* = \langle \mathcal{M}, \mathcal{R}, y \rangle$ be a DAC and $R_i = k_1^i m_1 + ... + k_n^i m_n \to j_1^i m_1 + ... + j_n^i m_n \in \mathcal{R}$, $i = 1, ..., m$. R_i is applied to Q^* if there are enough reactants in y to consume the molecules in R_i. We say R_i is fulfilled leading a new reaction vessel y^{R_i} if $\mathcal{A}(y^{R_i}, m_t) = \mathcal{A}(y, m_t) - k_t^i + j_t^i$ for $t = 1, ..., n$. We denote the application of R_i over y as $y \xrightarrow{R_i} y^{R_i}$.*

Definition **38 (Apply a sequence of reactions)** *Let $P = R_{\alpha_1} \cdots R_{\alpha_s}$ a sequence of reactions of a flux vector $\mathbf{v}$. P is applied to the DAC $\langle \mathcal{M}, \mathcal{R}, y \rangle$ as follows: If R_{α_1} can be fulfilled, then R_{α_1} is fulfilled, leading to a new discrete reaction vessel $y^{R_{\alpha_1}}$. To the reaction R_{α_l} we have $y^{R_{\alpha_1} \cdots R_{\alpha_{l-1}}} \xrightarrow{R_{\alpha_l}} y^{R_{\alpha_1} \cdots R_{\alpha_l}}$ only if $y^{R_{\alpha_1} \cdots R_{\alpha_{l-1}}}$ has enough reactants to consume R_{α_l}, for $l = 2, ..., s$.*

Definition **39 (Terminal reaction vessel)** *Let $Q^* = \langle \mathcal{M}, \mathcal{R}, y \rangle$ be a DAC and $P = R_{\alpha_1} \cdots R_{\alpha_s}$ a sequence of reactions of a flux vector $\mathbf{v}$. The application of P over a $\langle \mathcal{M}, \mathcal{R}, y \rangle$ leads to a terminal discrete reaction vessel y^P such that $y \xrightarrow{R_{\alpha_1}} \cdots \xrightarrow{R_{\alpha_s}} y^P$, which is the discrete reaction vessel once all reactions specified in P are fulfilled.*

Example 9. $\mathcal{M} = \{a, b, c\}$, $\mathcal{R} = \{4a + 3b \to c, 2c \to 10a + 2b, a \to 2b\}$. This leads to the stoichiometric matrix $\mathbf{S} = \begin{pmatrix} -4 & 10 & -1 \\ -3 & 2 & 2 \\ 1 & -2 & 0 \end{pmatrix}$. We are going to focus in the self-maintaining

property of the AC, but similar analysis can be done for other properties.
Note that the self-maintaining sets are $\{\emptyset\}, \{b\}$ and $\{a, b, c\}$. But the only set in which the reactions can take place is $\{a, b, c\}$. It is easy to verify (by inspection) that the minimal flux vector which satisfies the condition of self-maintainance is $\mathbf{v}^* = (2, 1, 2)$. In fact $\mathbf{S}\mathbf{v}^* = (0, 0, 0)$, hence there is no destruction or creation of molecules in the reaction process specified by $\mathbf{v}^*$.
From a discrete point of view, to fulfill the sequence of reactions specified by $\mathbf{v}^*$ without taking care of the order in which reactions specified in $\mathbf{v}^*$ take place, it is easy to show that we require an initial multiset y such that $\mathcal{A}(y^*, m_i) \leq \mathcal{A}(y, m_i)$ where $y^* = 10a + 6b + 2c$.

On the other hand, consider the discrete reaction vessel $\bar{y} = 4a + 3b + c$. Note that for all $i = 1, ..., n$ we have $\mathcal{A}(\bar{y}, m_i) < \mathcal{A}(y^*, m_i)$. But applying the sequence of reactions $P^* = R_1R_2R_3R_3R_1$ of $\mathbf{v}^*$ to $\bar{y}$ we have

$$\bar{y} \xrightarrow{R_1} 2c \xrightarrow{R_2} 10a + 2b \xrightarrow{R_3} 9a + 4b \xrightarrow{R_3} 8a + 6b \xrightarrow{R_1} \bar{y}.$$

Then, we can *re-obtain* the multiset $\bar{y}$ with the sequence of reactions P^*. But we cannot re-obtain $\bar{y}$ from any sequence of reactions of $\mathbf{v}^*$ as did for y^*. In fact, the sequence of reactions P^* cannot begin with R_2 because $\bar{y}$ has not enough reactants to fulfill it, and if P^* would begin with R_3, $\bar{y}^{R_3}$ would not be able to fulfill R_2.

Theorem **5 (Equivalence between AC and DAC)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $\mathcal{Y} \subseteq \mathcal{M}$ a self-maintaining set of $\langle \mathcal{M}, \mathcal{R} \rangle$ and $\mathbf{v}$ a flux vector which verifies the self-maintenance of $\mathcal{Y}$. Then, there exist a DAC $\langle \mathcal{M}, \mathcal{R}, y \rangle$ where $y \in \mathcal{P}_M(\mathcal{Y})$, such that for every sequence of reactions P obtained from $\mathbf{v}$ we have that $\mathcal{A}(y, m_i) \leq \mathcal{A}(y^P, m_i)$ for $i = 1, ..., n$.*

Proof Let $\mathcal{R} = \{R_1 = A_1 \to B_1, ..., R_k = A_k \to B_k\}$, $\mathbf{v} = (v_1, ..., v_k)$. As the stoichiometric matrix involves only integer coefficients we can consider without loss of generality that $v_i \in \mathbb{N}$ for $i = 1, ..., k$. Let $\omega = \sum_{i=1}^{k} v_i$. If y is such that $\mathcal{A}(y, m_i) = \sum_{j=1}^{k} \mathcal{A}(A_j, m_i)$ we are going to have enough reactants for every reaction R_{α_i} in the application of the sequence of reactions $P = R_{\alpha_1} \cdots R_{\alpha_\omega}$ obtained from $\mathbf{v}$, no matter the order in which reactions are applied. In fact, note that for every $u \leq \omega$ we have

$$\mathcal{A}(y^{\alpha_1 \cdots \alpha_u}, m_i) = \mathcal{A}(y, m_i) + \sum_{l=1}^{u} (-\mathcal{A}(A_{\alpha_l}, m_i) + \mathcal{A}(B_{\alpha_l}, m_i)) \geq 0.$$

Then we reach the terminal discrete reaction vessel y^P. Furthermore note that: $\mathcal{A}(y^P, m_i) = \mathcal{A}(y, m_i) + \sum_{l=1}^{\omega}(-k_i^{\alpha_l} + j_i^{\alpha_l}) = \mathcal{A}(y, m_i) + f_i$, with f_i the production rate of molecule m_i. Then $\mathcal{A}(y^P, m_i) \geq \mathcal{A}(y, m_i)$ if and only if $f_i \geq 0$, for all $i = 1, ..., n$.

□

Definition **40 (Discrete-self-maintaining)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $\mathcal{Y} \subseteq \mathcal{M}$ and $\mathcal{R}_{\mathcal{Y}} = \{R_{\alpha_1}, ..., R_{\alpha_l}\}$. The DAC $\langle \mathcal{Y}, \mathcal{R}, y \rangle$, where $y \in \mathcal{P}_M(\mathcal{Y})$, is discrete-self-maintaining if and only if there exists a sequence of reactions P such that $s = R_{\alpha_1} \cdots R_{\alpha_l}$ is a subsequence of P and $\mathcal{A}(y^P, m_i) \geq \mathcal{A}(y, m_i)$ for all $i = 1, ..., n$.*

Corollary **4** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC and* $\mathcal{Y} \subseteq \mathcal{M}$*.* $\mathcal{Y}$ *is self-maintaining iff for some* $y \in \mathcal{P}_M(\mathcal{Y})$ *the DAC* $\langle \mathcal{Y}, \mathcal{R}_{\mathcal{Y}}, y \rangle$ *is discrete-self-maintaining.*

Proof Follows from Theorem 5 and Definition 40.

$\square$

Chapter 5

Discrete formalisms and DAC

We are going to introduce two equivalent formalisms to frame an algorithmic study of AC.

5.1 Petri Nets

Petri nets are one of the most popular formal models for the representation and analysis of parallel processes. It is due to C.A. Petri, who introduced it in his doctoral dissertation in 1962.

Definition **41 (A Net)** *A net N is a triple (S, T, F), where S and T are two disjoint, finite sets, and F is a relation on $S \cup T$ such that $F \subseteq (S \times T) \cup (T \times S)$.*

The elements of S and T are called *places* and *transitions*, respectively, and the elements of F are called *arcs*. A *marking* of a net $N = (S, T, F)$ is a mapping $M : S \to \mathbb{N}$. A marking M *enables* a transition t if it marks all its input places. If t is enabled at M, then it can *occur*, and its occurrence leads to the successor marking M' , which is defined for every place s as follows: a token is removed from each input place of t and a token is added to each output place of t (if a place is both input and output place of a transition, then its number of tokens does not change). This is denoted by $M \xrightarrow{t} M'$.

Definition **42 (Petri net)** *A Petri net is a pair (N, M_0), where N is a net and M_0 a marking of N, called initial marking.*

A sequence $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} ... \xrightarrow{t_n} M_n$ *is a finite occurrence sequence* leading from M_0 to M_n and we write $M_0 \xrightarrow{t_1 \cdots t_n} M_n$. A sequence $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} ...$ is an infinite occurrence sequence. An occurrence sequence is *maximal* if it is infinite, or it leads to a marking which does not enable any transition. A marking M of N is *reachable* if $M_0 \xrightarrow{\sigma} M$ for some sequence σ.

Definition **43 (Language of a Petri net)** *The language of* (N, M_0) *with respect to* M^f *is*

$$L(N, M_0, M^f) = \{\sigma | M_0 \overset{\sigma}{\to} M^f\},$$

and the trace set of (N, M_0) *is*

$$T(N, M_0) = \{\sigma | M_0 \overset{\sigma}{\to} M \text{ for some marking } M\}.$$

5.2 Vector Addition Systems

VAS arose from the necessity to develop a formal framework of parallel programming [29]. A vector in this context is usually an element of $\mathbb{Z}^n$ or $\mathbb{N}^n$ for a fixed n. $a[i]$ denotes the i-th coordinate of the vector a. For $a, b \in \mathbb{Z}^n$, $a + b$ is defined as usual and $a \geq b \Leftrightarrow a[i] \geq b[i]$ for $i = 1, \ldots, n$. 0 is the zero vector. For a finite set V of vectors, V^* denotes the set of all finite strings in V, including the empty string λ. If $w = v_1 v_2 \ldots v_k$ $(v_1, \ldots, v_k \in V)$, then $a + w$ will denote the vector $a + v_1 + \ldots + v_k$. For example $a + xy = a + x + y$. Also $a + \lambda = a$. If w is obtained by the concatenation of two strings x and y from V^*, i.e $w = xy$, x is said to be a prefix of w. So the empty string λ and w itself are prefixes of w.

Definition **44 (Vector Addition system)** *A Vector Addition System (in short VAS) is an ordered pair* $A = (V, a_0)$ *where* $V \subset \mathbb{Z}^n$ *is a finite set of vectors and* $a_0 \in \mathbb{N}^n$.

Definition **45 (Legal string)** *Given a VAS* $A = (V, a_0)$ *a string* $w \in V^*$ *is said to be legal in* A *if and only if for every prefix* x *of* w, $a_0 + x \in \mathbb{N}^n$. *The set of all legal strings of* A *is the language of* A *and will be denoted by* $L(A)$.

5.3 Equivalence between VAS and PN

There are several previous works in which VAS and PN are considered as equivalent formalisms to study algorithmic and complexity issues in the field of concurrent computation, see for example [36, 38, 16, 21]. There are also works in which Petri Nets are used to model chemical reaction networks [23, 30, 46, 27], but there is no work linking these formalisms to the computational aspects of organisation computation in AC. We are going to show the equivalence of PN and VAS by an example.

Example 10. Let $N = (\{s_1, s_2, s_3\}, \{t_0, t_1, t_2, t_3\}, \{f_1, ..., f_{10}\})$, with

$$f_1 = (s_1, t_3),\ f_2 = (t_3, s_2),\ f_3 = (s_2, t_2),\ f_4 = (t_3, s_1),$$
$$f_4 = (t_2, s_3),\ f_5 = (s_3, t_2),\ f_6 = (s_3, t_0),$$
$$f_7 = (s_1, t_1),\ f_8 = (t_1, s_3),\ f_9 = (s_1, t_0).$$

Let M be the initial marking s.t $M(s_1) = 1,\ M(s_2) = 0,\ M(s_3) = 0.$

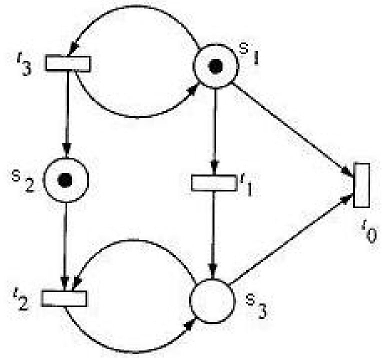

Figure 5.1: The Petri Net (N, M) showed as a bipartite graph. The round nodes represents the places, the rectangular nodes are the transitions, the black circles represent the available tokens in the places of M.

Note that transitions represent how the tokens in the places are consumed, produced or exchanged. In this case the transition t_0 its a token consumer and the rest of transitions are exchange transitions. There is no production transition in this example.

Now consider the VAS $A = (V = \{\mathbf{v_0}, \mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}\}, \mathbf{a_0} = (1, 1, 0))$ with

$$\begin{aligned} \mathbf{v_0} &= (-1, 0, -1), \\ \mathbf{v_1} &= (-1, 0, 1), \\ \mathbf{v_2} &= (0, -1, 0), \\ \mathbf{v_3} &= (0, 1, 0). \end{aligned} \tag{5.1}$$

The amount of tokens of M in the place s_i is represented by the i-th coordinate of a_0. Note that $\mathbf{v_0}$ represents the consumption transition t_0, in the same way $\mathbf{v_1}$ represents t_1. Vectors $\mathbf{v_2}, \mathbf{v_3}$ represents the transitions t_2, t_3. But there is a subtle detail: note that t_2 consumes one token from s_2 and one token from s_3 to generate a token in s_3. But this consumtion-production of token s_3 is not seen in $\mathbf{v_3}$ because it has only zero in the coordinate which represents s_3. From here we deduce that the set of reachable markings of a given PN (N, M) is equal to the legal states of its corresponding VAS (V, a_0) if and only if every place has at least enough tokens to fire all the transitions in which the amount of consumed tokens is equal to the produced tokens (and thus a_0 will be the vectorial representation of the amount of tokens of M).

Remark: The equivalence stated above between PN and VAS is useful enough to study the structural properties of these systems. Results about structural properties of PN and VAS are the source of insights and ideas of this work.

5.4 VAS induced by a DAC

The language of a given VAS is composed by multisets[1] instead of sets. We are going to show an equivalence (w.r.t the discrete-self-maintaining property) between DAC and VAS, and due to equivalence of VAS and PN [36], the properties of AC can also be studied in PN. To show the relation between DAC and VAS, we need first the notion of vector addition system with states, which is a VAS with a state transition control. Formally:

DEFINITION **46 (Vector Addition system with States)** *Let* $V \subset \mathbb{Z}^n$ *be a finite set of vectors,* T *a finite set of states and* $Q \subseteq T \times V \times T$ *a transition relation. We say that the four-tuple* $A_T = (V, T, \alpha, p_0, Q)$ *with* $\alpha \in \mathbb{N}_0^n$*,* $p_0 \in T$ *is a Vector addition system with states, in short VASS.*

A VASS represents a VAS in which each state has its own set of vectors, and the application of any vector leads to a new state. Thus any single state VASS is exactly a VAS. The application of a vector and the corresponding state change will be called *transition*. The transition $(p, q, v) \in Q$ which will be noted by $p \to (q, v)$, can be applied at point x at state p and yields the point $x + v$ in state q. We say the transition is *legal* iff $x + v \geq 0$. We denote the concatenation (or word) of the transitions $t_1, ..., t_k$ as $w = t_1 \cdots t_k$. We define that w is *reachable* from a point x_0 and a state p_0 if and only if every prefix $\bar{w} = t_1 \cdots t_l$, $1 \leq l \leq k$ of w is legal. We denote the application of a reachable word of transitions w from a point x_0 and a state p_0, leading to a state x_f and a state p_f as $(x_0, p_0) \xrightarrow{w} (x_f, p_f)$.
We are going to introduce a way to generate a VASS from a DAC (which is derived from an AC as well) in which the words of the VASS simulate sequences of reactions applied to the DAC (which in turn simulate flux vectors applied over the stoichiometric matrix derived from the AC). It is inspired in the similarity between the reaction process in DAC, and the transitions defined in the VASS formalism.

THEOREM **6 (Relating DAC, PN and VAS from organisations)** *Let* $\langle \mathcal{M}, \mathcal{R}, y \rangle$ *with* $y \in \mathcal{P}_M(\mathcal{M})$ *be a DAC. Then there exists a VASS* $A = (T, V_{\mathcal{R}}, \alpha, p_0, Q)$*, such that for every sequence of reactions* P *which can be fulfilled in* $\langle \mathcal{M}, \mathcal{R}, y \rangle$*, there exists a reachable word* w_P *in* A*. Furthermore if* P *verifies the discrete-self-maintainance of* $\langle \mathcal{M}, \mathcal{R}, y \rangle$*, then the reachable word* w_P *in* A *is such that* $(\alpha, p_0) \xrightarrow{w_P} (\bar{\alpha}, \bar{p})$ *with* $\bar{\alpha} \geq \alpha$*.*

Proof We are going to define A: α is the vectorial representation of y (we assume a given order in $\mathcal{M}$), the initial state p_0 belong to some state control T which we will derive later and $V_{\mathcal{R}}$ is derived from $\mathcal{R}$ as follows:
Each $R_i = A_i \to B_i$ is represented in $V_{\mathcal{R}}$ by a reactant vector $\mathbf{v_i^r}$ and production vector $\mathbf{v_i^p}$ defined by

$$\mathbf{v_i^r} = (-\mathcal{A}(A_i, m_1), ..., -\mathcal{A}(A_i, m_n)), \quad \mathbf{v_i^p} = (\mathcal{A}(B_i, m_1), ..., \mathcal{A}(A_i, m_n)).$$

The state control T and the transitions are constructed to keep causality of the reactions. T has to have one state r_i for every reaction $R_i \in \mathcal{R}$, which represents the beginning of the

---

[1] Where muiltiset are represented as vectors in $\mathbb{N}^n$.

reaction R_i (where the reactants are consumed), and one state p which represents the end of the reaction (where the products are created and the system is ready for a new reaction). Then $T = \{r_1, ..., r_m, p\}$. Finally, we need to define the relation transition Q: For each r_i we have transitions $p \rightarrow (r_i, \mathbf{v^r_i})$ and $r_i \rightarrow (p, \mathbf{v^p_i})$, and $p_0 = p$.
We can *simulate* the application of the reaction R_γ with the word $w_\gamma = \mathbf{v}^\mathbf{r}_\gamma \mathbf{v}^\mathbf{p}_\gamma$. If y has enough reactants to fulfill R_γ then the transition $p \rightarrow (\mathbf{v}^\mathbf{r}_\gamma, r_\gamma)$ is legal, then w_γ is reachable. In fact, $(\alpha, p) \xrightarrow{w_\gamma} (\alpha_\gamma, p)$ where α_γ is the vectorial representation of y^{R_γ}. Furthermore, if $P = R_{\gamma_1} \cdots R_{\gamma_f}$ is a sequence of reactions which reaches the terminal discrete reaction vessel y^P, then the word $w_P = \mathbf{v}^\mathbf{r}_{\gamma_1} \mathbf{v}^\mathbf{r}_{\gamma_1} \cdots \mathbf{v}^\mathbf{r}_{\gamma_f} \mathbf{v}^\mathbf{p}_{\gamma_f}$ hold $(\alpha, p_0) \xrightarrow{w_P} (\bar{\alpha}, \bar{p})$, where $\bar{\alpha}$ is the vectorial representation of y^P and $\bar{p} = p$. Finally if P verifies the discrete-self-maintainance of $\langle \mathcal{M}, \mathcal{R}, y \rangle$, it means we have $\mathcal{A}(y, m_i) \leq \mathcal{A}(y^P, m_i)$ for $i = 1, ..., n$, hence we have $\bar{\alpha} \geq \alpha$.

$\square$

DEFINITION **47 (Induced VASS from a DAC)** *The VASS A described in the proof of Theorem 6 is called the induced VASS from the DAC* $\langle \mathcal{M}, \mathcal{R}, y \rangle$.

COROLLARY **5** *The induced VASS A from* $\langle \mathcal{M}, \mathcal{R}, \mathcal{Y}_M \rangle$ *is unique and can be constructed in* $|\langle \mathcal{M}, \mathcal{R} \rangle|$ *TIME.*

LEMMA **10** *[24] Is posible to simulate a VASS of dimension n generating an equivalent VAS of dimension* $n + 3$ *with an* $O(n)$ *time procedure. Then, VASS and VAS have the same expressive power.*

Remark: It is important to note that the VASS generated in Theorem 6 must have two states because it has to consider the causality of the discrete approach of reactions (see Example 9). If the initial reaction vessel has unlimited[2] molecules, then it is not necessary to consider two states. This means it is required to consider one single state (and thus we have a VAS instead of a VASS) and define the reactions as the columns of the stoichiometric matrix instead of consider separately the consumed (left hand side of the reaction specified in AC) and produced (right hand side of reactions specified in AC) parts of the reaction.

---

[2] This will be defined precisely in Chapter 7.

Chapter 6

Discrete properties and AC

In this chapter we are going to interpret in DAC language some of the well known properties in Petri nets and Vector addition systems in order to show the strong relation between the properties defined for Algebraic chemistries and its discrete counterpart. We will let fix for this chapter an AC $Q = \langle \mathcal{M}, \mathcal{R} \rangle$, a DAC $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$, where $|\mathcal{M}| = n$, $|\mathcal{R}| = k$ and $\mathcal{A}(M, m_i) > 0$ for $i = 1, ..., n$. There are several structural properties defined for PN [15, 36, 40] as well as for VAS [29, 22, 38]. The application of PN to biochemical modelling has been considered in previous works [23, 30, 46], but this approach has never been considered for organisation computation. We are going to review some relevant properties which are strongly related to the concept of organisation, in particular with respect to the self-maintainance property. As DAC, PN and VAS are equivalent (see Chapter 5), we are going to define the properties using the PN or DAC notation. We will also discuss briefly the relation that each of these properties has w.r.t biochemical reaction networks and we will show examples (in Petri Net bipartite graph notation) which exhibits the mentioned properties (all the examples are extracted from [36]).

6.1 Deadlock-freedom

Definition 48 **(Deadlock-free Petri Net)** *A Petri net is deadlock-free if every reachable marking enables some transition.*

In DAC terminology, deadlock-freedom means that every firable sequence of reactions P applied to the initial reaction vessel M will lead to a resulting reaction vessel M^P which is able to fire some reaction. We will explore the consecuences of this fact.

Lemma **11** *Let $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ be a deadlock-free DAC. Then there exists an infinite firable sequence of reactions.*

Proof Follows strightforward by using inductively the Definition 48.

□

Now we can sate a result which will be useful in the rest of the chapter.

Lemma **12** *Let* $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ *be a DAC. If there exist an infinite sequence* P *of reactions in* $\mathcal{R}$ *which can be fired from* M*, then there exist a set* $Y \subseteq \mathcal{M}$ *which is self-maintaining.*

Proof Let $P = R_{p_1} R_{p_2} \cdots$. Hypotesis implies that there exist a set of reactions $R^* \subseteq \mathcal{R}$ which is fired infinitely often in P. If R^* has inflow reactions, then the products of the inflow reactions in R^* forms a self-maintaining set.
If R^* have not any inflow reaction, let $M_0 = M$ and $M_i = M_0^{R_{p_1} \cdots R_{p_i}}$. M_i represents the reaction vessel once the first i reactions of the sequence P have been applied to M. As the sequence $\{M_i\}_{i=0}^{\infty}$ is infinite, we can extract an infinite subsucession $\{M_i^1\}_{i=0}^{\infty}$ in which the first coordinate of each term is non decreasing with respect to i. Now we can extract from $\{M_i^1\}_{i=0}^{\infty}$ an infinite subsequence $\{M_i^2\}_{i=0}^{\infty}$ in which the second coordinate is no decreasing with respect to i. Following this process, $\{M_i^n\}_{i=0}^{\infty}$ is an infinite subsequence of reaction vessels in which each coordinate is non decreasing with respect to i. Let Y the set of molecules in which M_0^n has positive values and S such that $M_0^n \xrightarrow{S} M_1^n$. As $\mathcal{A}(M_0^n, m_i) \leq \mathcal{A}(M_1^n, m_i)$ for all $i = 1, ..., n$, we have the DAC $\langle \mathcal{Y}, \mathcal{R}_{\mathcal{Y}}, y \rangle$where $y = M_0^n$ is discrete-self-maintaining (verified by the sequence of reactions S). Then by Corollary 4 we have Y is self-maintaining.

□

Corollary **6** *Let* $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ *be a DAC. If there exists an infinite sequence* P *of reactions in* $\mathcal{R}$ *which can be fired from* M *and each reaction* R *of* $\mathcal{R}$ *is infinitely often in* P*, then* $\mathcal{M}$ *is self-maintaining.*

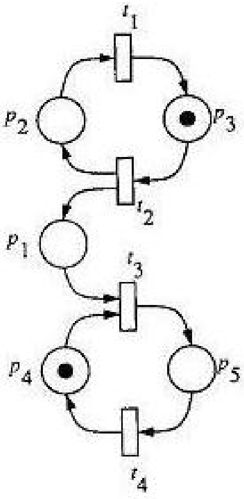

Figure 6.1: A deadlock-free Petri Net.

6.2 Boundedness

Definition **49 (Bounded Petri net)** *A Petri net is bounded if its set of reachable markings is finite.*

In DAC terminology boundedness means that the possible set of reaction vessels reachable from the initial reaction vessel is finite. Thus, if some sequence S of reactions verifies the discrete-self-maintainance of a bounded DAC $Q = \langle \mathcal{M}, \mathcal{R}, M \rangle$, then the terminal reaction vessel M^S has to fulfill $M^S = M$. Because if $M^S \geq M$ but for some coordinate i we have $M^S[i] > M[i]$, then the successive application of S to M would lead to infinite different possible reachable states from M. This means a bounded and self-maintaining AC cannot *overproduce* molecules. The overproduction in biology is important; there exist molecules called *growth factors* which are responsible for the growth of organisms [48, 51]. We are going to study in detail the overproduction in AC in Chapter 8.

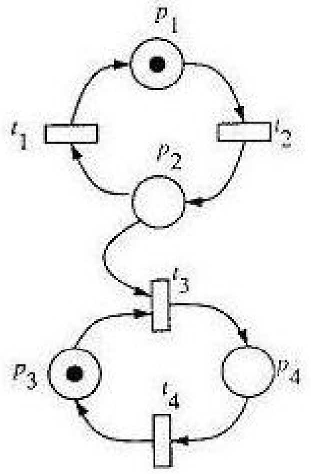

Figure 6.2: A bounded Petri Net.

6.3 Reachability

The reachability problem for Petri nets consists of deciding, given a Petri net (N, M_0) and a marking M of N, if M can be reached from M_0.

Definition **50 (Reachable marking)** *Let $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ be a DAC and $M^* \in \mathcal{P}_M(\mathcal{M})$ a discrete reaction vessel. M^* is reachable from M in Q^* iff there exists a sequence of reactions P built with the reactions in $\mathcal{R}$ s.t $M^* = M^P$.*

Remark: If there exist M^* s.t for some sequence of reactions P we have $\mathcal{A}(M^*, m_i) \leq \mathcal{A}(M^P, m_i)$ for all $i = 1, ..., n$, then Q^* is discrete-self maintaining, and thus Q is self-maintaining (by Corollary 4). This modified reachability problem is called the *Coverability problem*, we are going to explore this problem in detail in the next chapter.

6.4 Liveness

Liveness in Petri nets deals with the firability of transitions in a sequence of reactions, i.e how firable is a transition w.r.t the possible sequences of reactions.

DEFINITION **51 (Live reaction)** *Let $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ be a DAC and $R \in \mathcal{R}$ a reaction. R is*

- 0-*Live iff no sequence of reactions built from $\mathcal{R}$ and firable from M contains R in it.*
- 1-*Live iff there exists a sequence of reactions built from $\mathcal{R}$ and firable from M which contains at least once R in it.*
- 2-*Live iff for any $k > 0$ there exists a sequence of reactions built from $\mathcal{R}$ and firable from M which contains at least k times R in it.*
- 3-*Live iff R appears infinitely often in some sequence of reactions built from $\mathcal{R}$ firable from M.*
- 4-*Live iff R is* 1-*Live for any reachable reaction vessel from M.*

Remark: Note that if a given PN (N, M_0) is j-live, then (N, M_0) is i-live for any $1 \leq i \leq j \leq 4$.

LEMMA **13** *Let $Q = \langle \mathcal{M}, \mathcal{R} \rangle$ be an AC:*

1. *If for all $M \in \mathcal{P}_M(\mathcal{M})$ s.t $\mathcal{A}(M, m_i) > 0$ for all $i = 1, ..., n$, we have that the DAC $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ has a* 0-*live transition, then $\mathcal{M}$ is not self-maintaining.*
2. *If for some $M \in \mathcal{P}_M(\mathcal{M})$ we have the DAC $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ every transition is* 2-*live, then there is some set $Y \subseteq \mathcal{M}$ which is self-maintaining.*

Proof *1*) Hypothesis implies that we cannot build a sequence of reactions s.t all the reactions are fired at least once. This means that every flux vector which has all its coordinates greater that zero, would lead to a negative production rate for some molecule, and thus no flux vector would verify the condition of self-maintainance.
2) Hypothesis implies that for each $k \in \mathbb{N}$ we can build a sequence of reactions S_k which verifies 2-liveness of $\mathcal{R}$. Then by using Lemma 12 we complete the proof.

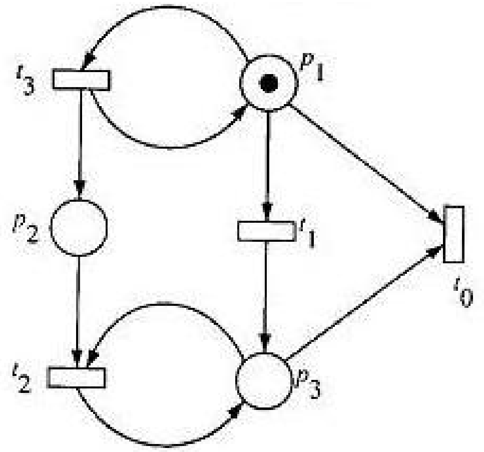

Figure 6.3: Transition t_1, t_2, t_3 are 1-live, 2-live and 3-live respectively.

□

There have been much effort in understanding liveness in PN under certain constrains on the sequences reactions; those constrains were called *fairness, justice, impartiality, etc.* These constrains are defined, in most cases, in terms of the relative amount of times which each reaction is fired in a live sequence of reactions. See [26, 9] for a detailed algorithmic work on those constrains. Liveness concept under *fairness-like* constrains has a deep link with biology. In biochemical systems the reactions can have different time-scales, this means that the chemical structure of the system constrains the relative amount of times that each reaction *happens* in the reaction vessel with respect to the rest of the reactions, these constraints come from the size of the molecular species involved in the reactions, their relative velocities, charges, and so on. We are not going to explore this aspect in this work.

6.5 Persistence

Definition **52 (Disabling reactions)** *Let $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ be a DAC and let $R_1, R_2 \in \mathcal{R}$. We say R_1 disable R_2 in Q^* if any sequence P of reactions containing R_2 is not firable from the DAC $\langle \mathcal{M}, \mathcal{R}, M^{R_1} \rangle$.*

Definition **53 (Persistent DAC)** *A DAC $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ is persistent if for any two different reactions R_1, R_2 of $\mathcal{R}$ and any reachable reaction vessel M^*, if R_1 and R_2 are enabled at M^*, then the occurrence of one cannot disable the other.*

Lemma **14** *Let $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ s.t every reaction in $\mathcal{R}$ is firable from M. If Q^* is persistent, then $\mathcal{M}$ is self-maintaining.*

Proof Let $\mathcal{R} = \{R_1, ..., R_k\}$. By hypothesis R_1 is firable from M, and Q^* is persistent. Hence there exist a sequence of reactions S_{12} such that R_2 is firable from $M^{S_{12}}$. As R_2 is firable from $M^{S_{12}}$, and Q^* is persistent, there exist a sequence of reactions S_{23} such that R_3 is firable from $M^{S_{12}S_{23}}$. Continuing this process, we will obtain a sequence $S_1 = S_{12}S_{23}\cdots S_{k-1k}S_{k1}$ in which all the reactions are fired at least once and every reaction in $\mathcal{R}$ is firable from M^{S_1}. Starting from M^{S_1}, we can repeat the process of build a sequence S_2 in which each reaction is fired at least once and every reaction is firable from $M^{S_1S_2}$. Then we can build an infinite sequence of reactions $S_1S_2\cdots$ in which every reaction is infinitely often in it. By Corollary 6 $\mathcal{M}$ is self-maintaining.

□

Corollary 7 *Let* $Q^* = \langle \mathcal{M}, \mathcal{R}, M\rangle$ *s.t every reaction in* $\mathcal{R}$ *is firable from* M*. If* Q^* *is persistent, then* $\mathcal{M}$ *is an organisation.*

Proof It follows from the the fact that $G_{CL}(\mathcal{M}) = \mathcal{M}$.

□

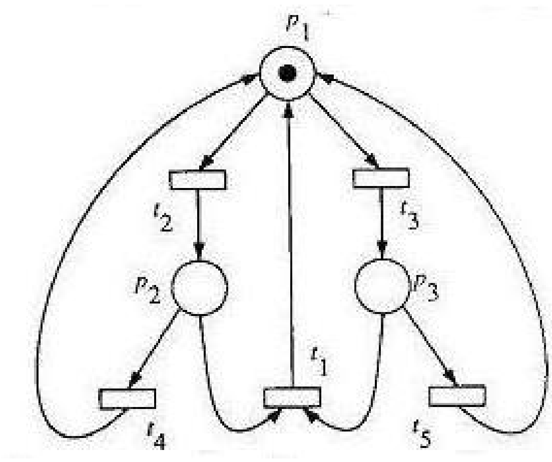

Figure 6.4: A Persistent Petri Net.

Remark: Persistence is strictly stronger property than self-maintainance. Let for example

$$Q^* = \langle \{a\}, \{R_1 = a \to \emptyset, R_2 = a \to 2a\}, a\rangle.$$

Clearly Q^* is discrete-self-maintainaing (and thus $Q = \langle\{a\}, \{R_1, R_2\}\rangle$ is self-maintaining). Note that R_1 and R_2 are reachable, but firing R_1 we disable R_2. Persistence is a very important property in the study of concurrence [12, 31, 5]. It also has arised as a new source of research in biochemical networks in recent years, principally by works of Angeli *et.al.* It has been observed that persistent networks are stable, this meaning that all molecules has concentration greater than zero in asymptotic regime (for a detailed work on this matters see [3, 4]).

6.6 Home states

Definition **54 (Home state of a DAC)** *Let* $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ *be a DAC. A discrete reaction vessel* $M^* \in \mathcal{P}_M(\mathcal{M})$ *is a home state if it is reachable from every reachable state from* M*.*

Definition **55 (Reversible DAC)** *A DAC* $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ *is reversible iff* M *is reachable from any reachable marking.*

Lemma **15** *Let* $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ *be a DAC. If* $M^* \in \mathcal{P}_M(\mathcal{M})$ *is a home state of* Q^**, then the DAC* $\langle \mathcal{M}, \mathcal{R}, M^* \rangle$ *is reversible.*

Corollary **8** *Let* $Q^* = \langle \mathcal{M}, \mathcal{R}, M \rangle$ *be a DAC. If* $M^* \in \mathcal{P}_M(\mathcal{M})$ *is a home space of* Q^**, then the set* $M^*_{set} = \{m \in \mathcal{M} \text{ s.t } \mathcal{A}(M^*, m) > 0\}$*, is self-maintaining.*

Proof Let $\mathcal{R}_{M^*_{set}} = \{R_1, ..., R_l\}$ and $s = R_1 \cdots R_l$. As M^* is a home state then there exists a sequence s' s.t $(M^*)^{ss'} = M^*$. Then $\langle M^*_{set}, \mathcal{R}_{M^*_{set}}, M^* \rangle$ is discrete-self-maintaining. Thus by Corollary 4 M^*_{set} is self-maintaining.

□

The home state definition is used as an extension of the reversibility concept in concurrent systems as it is stated in Lemma 15, see [36] for more details about reversibility and home states. The concept of home states has been used to understand the reversible process inside biochemical networks principally by Reddy *et.al* in [43, 44].

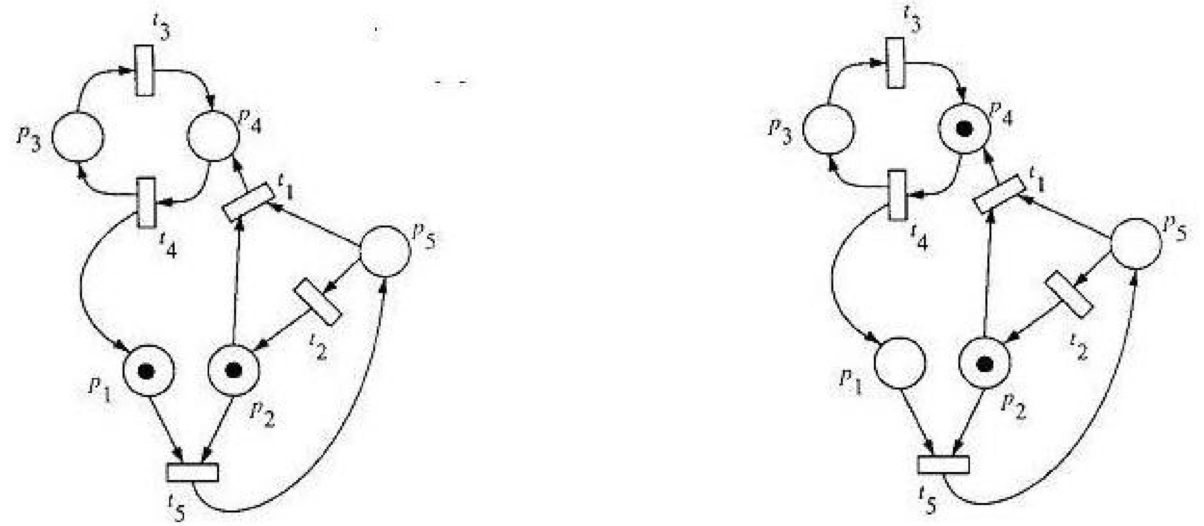

Figure 6.5: The marking in the left picture is not a home state, but the marking in the right picture is a home state Petri Net.

Chapter 7

Computing organisations from discrete point of view

This chapter analizes the problem of *verifying the self maintaining property of a given set of molecules.* This problem will be studied using concepts, algorithms and methods derived from PN, VAS and Integer Linear Programming (ILP) (see section 7.4). This means we are going to study the problem from a discrete point of view.

We will present the Karp and Miller tree [29] (see definition 57). Based on some modifications of the Karp and Miller tree we will develop an algorithm to build a tree useful to verify self maintainance. Finally we will to bound the size of this new tree using results from Linear Algebra and ILP.

In this chapter will fix a DAC and use the following notation:
$Q = \langle \mathcal{M} = \{m_1, ..., m_n\}, \mathcal{R} = \{R_1, ..., R_k\}\rangle$ be an AC, and $Q^* = \langle \mathcal{M}, \mathcal{R}, M_0 = m_1^{p_1} \cdots m_n^{p_n}\rangle$ be a DAC. Let μ (μ^* respectively) be the number of bits required to represent the largest value mentioned in Q (Q^* respectively).

7.1 The Karp & Miller Tree

The Karp & Miller tree was developed to decide the boundedness[1] problem in VAS, which is equivalent to the non-termination problem in PN. It was shown that it is also useful to decide the covering problem[2] of VAS and PN (via a PTIME reduction) [38].

Definition **56 (Coverability Problem)** *Given a discrete reaction vessel M_f, the coverability problem in DAC consists in deciding if there exists a sequence of reactions s in Q^* such that $M_0 \xrightarrow{s} M$ (M is the terminal reaction vessel[3] of the sequence of reactions s starting*

---

[1] See Section 6.2.

[2] See section 6.3.

[3] See definition 39.

from M_0) and for all $m \in \mathcal{M}$ it holds $\mathcal{A}(M,m) \geq \mathcal{A}(M_f,m)$. We say Q^* eventually covers M_f or simply Q^* covers M_f.

In this chapter we consider Q^* as a Vector addition system or a Discrete algebraic chemistry indifferently (the equivalence between both formalisms is explained in detail in Chapter 5). We will also use vectorial notation for multisets. This means that the multiset $m_1^{p_1} \cdots m_n^{p_n}$ will be represented by the vector $(p_1, ..., p_n)$.

In order to explain the algorithm to construct the Karp & Miller tree, we introduce the following terminology:

1. The relation $\leq$ between n-dimensional vectors is defined as follows: $y \leq z$ if and only if $y_i \leq z_i$, for all $i = 1, 2, ..., n$.
2. 0 denotes the zero vector or the integer 0 depending on the context.
3. ω is a symbol such that, if n is an integer, then $n < \omega$ and $n + \omega = \omega - n = \omega$. Most of results in this chapter considers the extended natural numbers set $\mathbb{N} \cup \{\omega\}$ which includes ω. The element ω was first introduced in [29] to deal with infinite growing sequences of vectors in VAS, we are going to use ω in the same aim.
4. A *rooted tree* is a directed graph such that one vertex (the root δ) has no edges directed into it, each other vertex has exactly one edge directed into it, and each vertex is reachable from the root.
5. If ξ and ν are distinct vertices of a rooted tree, and there is a path from ξ to ν, then we write $\xi \prec \nu$.
6. If there is an edge from ξ to ν we say ν is a successor of ξ. We equivalently say ξ is an ancestor of ν.
7. A vertex without successors is called a *leaf* or *end*.
8. $l : V \to (\mathbb{N} \cup \{\omega\})^n$ is a labelling function of the set V of vertices.
9. Following definition 37, $l(\xi)^R$ represents the multiset (in vectorial notation) once the reaction R is applied to the multiset $l(\xi)$.

We shall give an algorithm for constructing a labelled rooted tree to verify coverability. Every node of the tree will be labelled by a discrete reaction vessel (in vectorial notation). The root of the tree is labelled by M_0, and each node ξ has one son for each reaction that it can fire, labelled by the multiset resulting of applying the reaction to the label of ξ. To ensure that the tree is always finite, it is necessary to introduce the element ω to identify the infinite growing paths in the tree.

DEFINITION **57 (Karp & Miller tree)** *For any DAC Q^*, the rooted labelled tree $\mathcal{T}(Q^*)$ constructed by Algorithm 5 is called the Karp and Miller tree, or K&M tree.*

Algorithm 5 The Karp and Miller Tree

```
Require: $Q^* = \langle \mathcal{M}, \mathcal{R}, M_0 = m_1^{p_1} \cdots m_n^{p_n} \rangle$
Ensure: A labelled rooted tree $\mathcal{T}(Q^*) = (V, E, l : V \to \mathbb{N} \cup \{\omega\})$.
The root is labelled $M_0$, Let $\nu$ be a vertex
if For some vertex $\xi$, $\xi \prec \nu$ and $l(\xi) = l(\nu)$ then
  $\nu$ is an end
else
  The successors of $\nu$ are in one-to-one correspondence with the elements $R = A \to B \in \mathcal{R}$ such that $0 \leq l(\nu)^R$.
  Let the successor of $\nu$ corresponding to $R$ be denoted by $\nu_R$
  for each $i = 1, ..., n$ do
    The $i$-th coordinate of the label $l(\nu_R)$ denoted by $l(\nu_R)[i]$ is determined as follows:
    if There exists $\xi$ s.t $\xi \leq \nu$, $l(\xi) \leq l(\nu)^R$ and $\mathcal{A}(l(\xi), m_i) < \mathcal{A}(l(\nu)^R, m_i)$ then
      $l(\nu_R)[i] = \omega$.
    else
      $l(\nu_R)[i] = \mathcal{A}(l(\nu), m_i) + \mathcal{A}(B, m_i) - \mathcal{A}(A, m_i)$.
    end if
  end for
end if
```

Lemma **16** *For every DAC Q^* the K&M tree $\mathcal{T}(Q^*)$ is finite.*

Proof The DAC Q^* has an equivalent VAS V, the proof follows from [29].

□

Lemma **17** *The complexity of the coverablity problem with input Q^* is in $(\mu^* + log(2))2^{cn}$-SPACE, where c is some constant independent from Q^*.*

Proof The DAC Q^* has an equivalent VAS V, the proof follows from [45].

□

Remark: Algorithm 5 does not find the sequence of reactions which covers the initial reaction vessel (the root of the tree). Algorithm 5 only verifies the existence of such sequence, because the ω element induces a lost of information in how the covering sequences are.

For example, let

$$Q^* = \langle \{a, b\}, \{R_1 = a \to b^2, R_2 = b \to a, R_3 = b \to \emptyset\}, a \rangle.$$

The K&M algorithm returns the tree:

In the next section we will give a sligthly modified version of the Karp & Miller tree construction to decide self maintainance in AC.

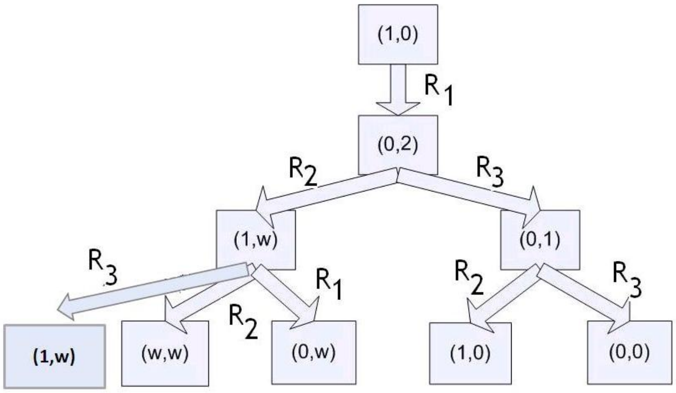

Figure 7.1: The node labelled by (ω, ω) corresponds to the sequence of reactions $R_1R_2R_2$ applied to $(1,0)$ in the tree. But $(1,0)^{R_1R_2R_2} = (2,0)$, which only covers $(1,0)$ in the first coordinate. This is because $(1,0)^{R_1R_2} = (1,1)$ which covers $(1,0)$ in the second coordinate, thus the second coordinate is labelled by ω in the tree. $(1,\omega)^{R_2} = (2,\omega)$ which covers $(1,\omega)$ in the first coordinate, and then (ω,ω) reflects that both coordinates can be increased infinitely. The element ω induces a lost of information about how the coordinates were covered, in order to simplify the verification of coverability.

7.2 Generalizing the algorithm by avoiding the initial marking

The K&M algorithm depends on the initial reaction vessel M_0 of Q^*. To verify self maintainance of Q is equivalent to verify the existence of some DAC Q^* (meaning some initial reaction vessel M_0) such that, $M_f = M_0^{R_1 \cdots R_k}$ is covered by Q^* (to see a formal proof check Theorem 5). Thus, to verify the self maintainance of Q, first we should *guess* an initial reaction vessel M_0 with enough molecules to derive the minimal sequence of reactions which would cover M_0 starting from $M_f = M_0^{R_1 \cdots R_k}$. Such M_0 should be then the root of the K&M tree.

To simplify the explanations we are going to assume the existence of an element α which represents the *unlimited* (see remark of lemma 10) amount of molecules requiered in the initial reaction vessel to verify self maintainance. This means, the minimal sequence of reactions s which verifies self maintainance of Q^* can be applied to the initial reaction vessel $M_0 = (\alpha, ..., \alpha)$ (leading to a terminal reaction vessel $M_0^s \geq M_0$). At the end of the chapter we will prove that for every network this α is a fixed natural number which is determined by the size of the minimal (in L_1 norm) flux vector which verifies the self maintainane of $\mathcal{M}$ and by the maximal element mentioned in Q represented by 2^μ (see Corollary 12 and Corollary 14). We introduce here this element α only to simplify the presentation.

Definition **58 (Unlimited element)** *Given an AC $\langle \mathcal{M}, \mathcal{R} \rangle$, let α be a natural number such that any sequence of some flux vector which verifies the self maintainance of $\mathcal{M}$ is firable by $(\alpha, ..., \alpha)$.*

In the rest of the chapter we will maintain the notations of Section 7.1.

7.2.1 A generalized tree

We are going to modify the Algorithm 5 to build the K&M tree for verifying self maintainance of Q.

1. We will change the input of the Algorithm 5 to prescind from the initial marking, now it will be

   Require Q.

2. As the sequence of reactions which verifies the discrete self maintainance of Q^* (and thus the self maintainance of Q) has to fire at least one time each reaction, we will change the labelling of the root of the tree

   1: The root ν_0 is labelled by $l(\nu_0) = M_0^s$, where $M_0 = (\alpha, ..., \alpha)$ and $s = R_1 \cdots R_k$.

3. As α represents an unlimited amount of molecules, we are able to fire all the reactions from each node (because there will be always enough molecules). Then every node would have $|\mathcal{R}|$ sons. We will change the condition in line 5 because it is not necessary to ask for a *firability* condition. Now it will be

   5: The successors of ν are in one-to-one correspondence with the elements
   $R = A \to B \in \mathcal{R}$.

Definition **59 (Intermediate K&M tree)** *The output of Algorithm 5 including the three modifications stated above is called the intermediate K&M tree.*

Theorem **7 (Intermediate K&M tree to verifiy self-maintainance)** *Let* $Q = \langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC,* $M_0 = (\alpha, ..., \alpha)$ *and* $s = R_1 \cdots R_k$*. Let* $\mathcal{T}_2(Q)$ *be the intermediate K & M tree. Q is self maintaining if and only if some node* ν *of* $\mathcal{T}_2(Q)$ *satisfies* $l(\nu) = M_0^{sP} \geq M_0$ *for some sequence of reactions P.*

Proof $\Rightarrow$: Let $\mathbf{v} = (v_1, ..., v_k)$ the flux vector which satisfies the self maintainance of Q. This means $v_i > 0$ for all $i = 1, ..., k$ and

$$\mathbf{S}\mathbf{v} \geq 0 \text{ , where } \mathbf{S} \text{ is the stoichiometric matrix implied by } Q. \tag{7.1}$$

We can assume without loss of generality that $\mathbf{v}$ has only natural coeffcients because $\mathbf{S}$ has only integer coefficients, (7.1) is an homogeneous equation and $v_i > 0$ for all $i = 1, ..., k$. This means $v_i \in \mathbb{N}$ for all $i = 1, ..., k$. Let t some sequence of reactions built from $\mathbf{v}$. Let $v'_i = v_i - 1$ for all $i = 1, ..., k$. Let $\mathbf{v}' = (v'_1, ..., v'_k)$ and P be a sequence of reactions built from $\mathbf{v}'$. By definition of self maintainance we have

$$M_0^{sP} = (M_0^{v_1 R_1 \cdots v_k R_k}) = (M_0^{R_1 \cdots R_k})^{v'_1 R_1 \cdots v'_k R_k} = M_0 + \mathbf{f},$$

where $\mathbf{f} = (f_1, ..., f_n)$ and $f_i \geq 0$ for all $i = 1, ..., n$. Then for some node ν of $\mathcal{T}_2(Q)$ we have $M_0^{sP} = l(\nu) \geq M_0$.

$\Leftarrow$: Let P be the sequence of reactions s.t $(M_0^s)^P \geq M_0$. This means $(M_0^s)^P = M_0 + \mathbf{f}$, where $\mathbf{f} = (f_1, ..., f_n)$ and $f_i \geq 0$ for all $i = 1, ..., n$. Let $v'_i = j$ if the reaction R_i is repeated j times in the sequence of reactions P. Then the flux vector $\mathbf{v} = (v_1 = v'_1 + 1, ..., v_k = v'_k + 1)$ is a solution of (7.1) s.t $v_i > 0$ for all $i = 1, ..., k$.

□

Corollary **9** *Let* $Q = \langle \mathcal{M}, \mathcal{R} \rangle$ *an AC,* $\mathcal{Y} \subseteq \mathcal{M}$, $|\mathcal{Y}| = p$ *and* $|\mathcal{R}_\mathcal{Y}| = k_\mathcal{Y}$. $\mathcal{Y}$ *is self-maintaining if and only if there exists a* $k_\mathcal{Y}$*-dimensional nonnegative integer flux vector* $\mathbf{v}$ *such that* $\mathbf{S}_\mathcal{Y}\mathbf{v} \geq -\mathbf{S}_\mathcal{Y}\mathbf{1}$, *where* $\mathbf{S}_\mathcal{Y}$ *is the* $p \times k_\mathcal{Y}$ *stoichiometric matrix associated to the AC* $\langle \mathcal{Y}, \mathcal{R}_\mathcal{Y} \rangle$ *and* $\mathbf{1}$ *is the* $k_\mathcal{Y}$*-dimensional vector with all its entries equal to* 1.

The intermediate K&M tree is an algorithm to verify self maintainance in AC. But as the initial reaction vessel is unlimited, when the AC is not self maintainaing the tree would be infinite. For example if $Q = \langle \{a, b\}, \{R = a \to b\} \rangle$, then the intermediate K&M tree will begin with the initial reaction vessel $M_0^R = (\alpha - 1, \alpha + 1)$, and it will not terminate.

To solve the problem of possible infiniteness of the depth of the tree, we are going to assume that there exist a bound in the *maximum amount of reactions* required to cover the intial reaction vessel. This implies that the depth of the intermediate K&M tree is bounded. As we are going to see later, there are results from ILP which allows us to ensure that.

We are going to introduce a depth tree cut $d_{max} \geq 1$. We will note with d the depth of the tree during its construction. We will also assume that if for some node ν we have $l(\nu) \geq (\alpha, ..., \alpha)$ then the algorithm terminates (see Theorem 7).

With those slight modifications we change line 1 and we add a step to the algorithm which builds the intermediate K&M tree.

```
1: The root is labelled $M_0$, $d = 0$. Let $\nu$ be a vertex.
13: **if** $l(\nu)^R \geq (\alpha, ..., \alpha)$ or $d > d_{max}$ **then**
terminate the Algorithm.
```

Definition **60 (Generalized K&M tree)** *The tree built by the intermediate K&M algorithm with the modifications stated in this section will be called the generalized K&M tree.*

Corollary **10** *The generalized K&M tree algorithm always terminates. Furthermore, the size of the generalized K&M tree to verify the self maintainance of* $\mathcal{Y} \subseteq \mathcal{M}$ *is bounded by* $\frac{k_\mathcal{Y}^{d_{max}+1} - 1}{k_\mathcal{Y} - 1}$ *where* $k_\mathcal{Y} = |\mathcal{R}_\mathcal{Y}|$.

Proof The termination of the algorithm follows from the finitness of the depth cut d_{max}. As the branching of the tree is $k_\mathcal{Y}$ we have the amount X_d of nodes at depth d in the tree follows the recurrence $X_{d+1} = k_{\mathcal{Y}} X_d$. As $X_0 = 1$ (the root node), we have the size $S(d)$ of the tree at depth d is

$$S(d) = \sum_{i=0}^{d} X_i = \sum_{i=0}^{d} k_{\mathcal{Y}}^i = \frac{k_{\mathcal{Y}}^{d+1}}{k_{\mathcal{Y}} - 1}.$$

Evaluating last expression for $d = d_{max}$ the proof is complete.

$\square$

Remark: The key in the complexity study of this algorithm will be given by the possible bounds that we can find for d_{max}. But there are another interesting improvements that we can do, by prunning some branches.

7.2.2 Reducing the size of the tree

In this section we are going to introduce some prunning rules into the generalized K&M tree to obtain a smaller tree.

Improvement 1: Eliminate decreasing pathways

Definition **61 (Decreasing node/sequence of reactions)** *Let ν be one of the nodes of the generalized K&M tree, ξ an ancestor of ν such that $l(\nu) \leq l(\xi)$ and for some coordinate $i = 1, ..., |l(\nu)|$ is verified that $\mathcal{A}(l(\nu), m_i) < \mathcal{A}(l(\xi), m_i)$. We say ν is a decreasing node, and the sequence of reactions t in the tree that fulfills $l(\xi)^t = l(\nu)$ is called decreasing sequence of reactions.*

Remark: Any decreasing node ν verifies $l(\nu) \leq M_0$ and for some $i = 1, ..., n$ is hold $\mathcal{A}(l(\nu), m_i) < \alpha$. In other case some ancestor ξ would cover M_0 (thus the algorithm to build the generalized K&M tree would have been finished already).

Lemma **18** *Let $\bar{\nu}$ be a decreasing node of the generalized K&M tree such that for some sequence t of reactions we have $l(\bar{\nu})^t \geq M_0$, and let q be a sequence of reactions such that $l(\nu_0)^q = l(\bar{\nu})$, where ν_0 is the initial node of the tree. Then there exist a subsequence q' of q such that $l(\nu_0)^{q't} \geq M_0$.*

Proof As $\bar{\nu}$ is a decreasing node, there exists an ancestor ξ such that $l(\nu) \leq l(\xi)$ and $\mathcal{A}(l(\xi), m_i) > \mathcal{A}(l(\nu), m_i)$ for some $i = 1, ..., n$. We have two cases:

1. $\xi = \nu_0$ (the root of the tree): By hypotesis we have

$$\mathcal{A}(l(\nu_0)^q, m_i) - \mathcal{A}(l(\nu_0), m_i) \leq 0, \text{ then } \mathcal{A}(M_0^q, m_i) - \mathcal{A}(M_0, m_i) \leq 0$$

for all $i = 1, ..., n$, and for some $i = i_0$ the inequality is strict. We also have that $\mathcal{A}(l(\nu_0)^{qt}, m_i) - \mathcal{A}(M_0, m_i) \geq 0$ for all $i = 1, ..., n$. Then we have

$$\mathcal{A}(l(\nu_0)^{t}, m_i) - \mathcal{A}(M_0, m_i) \geq \mathcal{A}(l(\nu_0)^{qt}, m_i) - \mathcal{A}(M_0, m_i) \geq 0,$$

for all $i = 1, ..., n$. This means in this case $q' = \emptyset$.

2. $\xi \neq \nu_0$: Let q' the sequence such that $l(\nu_0)^{q'} = l(\xi)$. Note that q' is a non-empty subsequence of q. Then applying the same argument as in the previous case, but from the node ξ instead of ν_0 we have that

$$\mathcal{A}(l(\nu_0)^{q't}, m_i) - \mathcal{A}(M_0, m_i) \geq \mathcal{A}(l(\nu_0)^{qt}, m_i) - \mathcal{A}(M_0, m_i) \geq 0,$$

for all $i = 1, ..., n$.

□

Remark: Lemma 18 indicates that decreasing pathways are not useful to verify the self maintainance.

Now the line 2 of the generalized K&M tree algorithm will be:
2: **if**, for some vertex ξ, $\xi \prec \nu$ and $l(\xi) \leq l(\nu)$ **then**
ν is an end.

This improvement allows to avoid decreasing pathways. Thus we reduce the amount of candidates to verify the discrete self maintainance of Q^* in the tree.

Improvement 2: Avoid permutations of the pathways

Let ν be a node of the generalized K&M tree. Note that given a sequence of reactions s we have $l(\nu)^s = l(\nu)^{p(s)}$, where $p(s)$ is any permutation of the sequence s. This means that both sequences applied to $l(\nu)$ lead to the same terminal reaction vessel. This is because we have an unlimited initial reaction vessel in the input, then we do not need to care about the order in which the reactions are fired (see Example 9 in Chapter 4). We are going to modify the construction of the generalized K&M tree in such a way that ensures that every sequence of reactions *modulo permutations* is present only once in the tree. This would lead to a smaller tree.

DEFINITION **62 (Succesor control function)** *For every node $\nu \neq \nu_0$ of the generalized K&M tree we define $b(\nu) = i$ if ν is the succesor of $\bar{\nu}$ and $l(\bar{\nu})^{R_i} = l(\nu)$, and $b(\nu_0) = 1$ for $i = 1, ..., k$.*

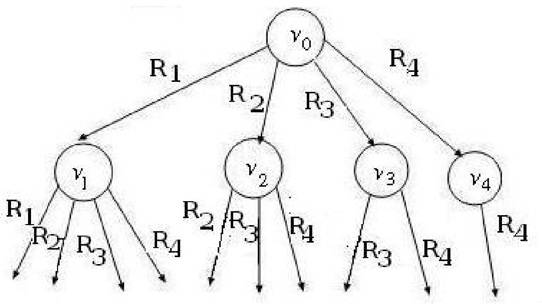

Figure 7.2: A picture of how will look the tree for $k = 4$. Note that $b(\nu_0) = 1, b(\nu_1) = 1, b(\nu_2) = 2, b(\nu_3) = 3, b(\nu_4) = 4$. All the possible sequences R_iR_j (modulo permutation) are resent once in the tree, and no sequence of reactions modulo permutations is present twice in the tree.

The function b applies to the nodes and it is used to control which reactions have to be fired below the nodes. Introducing this idea, line 5 of the generalized K&M tree looks:
5: The successors of ν *are in one-to-one correspondence with the elements* $R_j = A_j \to B_j \in \mathcal{R}$ *such that* $j \geq b(\nu)$.
With this improvements introduced, the generalized K&M tree algorithm look as follows:

Algorithm 6 The generalized K&M tree with prunning rules

```
Require: $Q = \langle \mathcal{M}, \mathcal{R} \rangle$ and $d_{max}$.
Ensure: verify the self maintainance of $\mathcal{M}$.
 1: The root is labelled $M_0^s$ and $d = 0$, where $M_0 = (\alpha, ..., \alpha)$ and $s = R_1 \cdots R_m$. Let $\nu$ be a vertex
 2: if For some vertex $\xi$, $\xi \prec \nu$ and $l(\xi) \leq l(\nu)$ then
 3:     $\nu$ is an end.
 4: else
 5:     The successors of $\nu$ are in one-to-one correspondence with the elements $R_j = A_j \to B_j \in \mathcal{R}$ such that $j \geq b(\nu)$.
 6:     Let the successor of $\nu$ corresponding to $R$ be denoted by $\nu_R$
 7:     for each $i = 1, ..., n$ do
 8:         The $i$-th coordinate of the label $l(\nu_R)$ denoted by $l(\nu_R)[i]$ is determined as follows:
 9:         if There exists $\xi \leq \nu$, $l(\xi) \leq l(\nu)^R$ and $\mathcal{A}(l(\xi), m_i) < \mathcal{A}(l(\nu)^R, m_i)$ then
10:             $l(\nu_R)[i] = \omega$.
11:         else
12:             $l(\nu_R)[i] = \mathcal{A}(l(\nu), m_i) + \mathcal{A}(B, m_i) - \mathcal{A}(A, m_i)$.
13:         end if
14:         if $l(\nu_R) \geq (\alpha, ..., \alpha)$ or $d \geq d_{max}$ then
15:             terminate the Algorithm
16:         else
17:             $d = d + 1$
18:         end if.
19:     end for
20: end if
```

Corollary **11** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ an AC, $\mathcal{Y} \subseteq \mathcal{M}$ and $|\mathcal{R}_{\mathcal{Y}}| = k_{\mathcal{Y}}$. Algorithm 6 verifies self-maintainance of $\mathcal{Y}$ iff there exist an $k_{\mathcal{Y}}$-dimensional nonnegative integer flux vector $\mathbf{v}$ such that $\mathbf{S}_{\mathcal{Y}}\mathbf{v} \geq -\mathbf{S}_{\mathcal{Y}}\mathbf{1}$ and $|\mathbf{v}|_1 = \sum_{i=1}^{k_{\mathcal{Y}}} |\mathbf{v}_i| \leq d_{max}$, where $\mathbf{1}$ is the $k_{\mathcal{Y}}$-dimensional vector with all its entries are equal to* 1.

Proof Follows from Theorem 7 and Corollary 9.

□

7.3 The size of the generalized K&M tree

In this section we are going to bound the size of the tree constructed by Algorithm 6.

Theorem 8 **(Sizing the generalized K&M tree)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $\mathcal{Y} \subseteq \mathcal{M}$, $|\mathcal{R}_{\mathcal{Y}}| = k_{\mathcal{Y}}$ and $|\mathcal{Y}| = n_{\mathcal{Y}}$. The size of the tree built by Algorithm 6 with input $(\langle \mathcal{Y}, \mathcal{R}_{\mathcal{Y}} \rangle, d_{max})$ is bounded by $\begin{pmatrix} d_{max} + k_{\mathcal{Y}} - 1 \\ k_{\mathcal{Y}} - 1 \end{pmatrix}$, where $\begin{pmatrix} \cdot \\ \cdot \end{pmatrix}$ is the biniomial coefficient.*

Proof By step 5 of Algorithm 6 every node ν such that $b(\nu) = j$ has $k_{\mathcal{Y}} - j + 1$ sons. We are going to call such a node of the type X_j and we denote by $X_j(d)$ the number of nodes of type X_j at depth d in the tree.
Note that

$$X_j(d+1) = \sum_{i=1}^{j} X_i(d) \text{ , } X_1(1) = 1 \text{ and } X_j(1) = 0 \text{ for } j > 1. \tag{7.2}$$

Let $G_i(z) = \sum_{i=0}^{\infty} X_i(n) z^n$ be the generating function of $X_i(n)$ for $i = 1, ..., k$.
As $X_1(n) = 1$ for all $n \in \mathbb{N}$, we have that $G_1(z) = \frac{1}{1-z}$.
Now we are going to prove by induction that for $i > 1$ $G_i(z) = \frac{z}{(1-z)^2}(\frac{1}{1-z})^{i-2}$.
From (7.2) we have that for $i > 1$

$$G_j(z) = z \sum_{i=1}^{j} G_i(z), \tag{7.3}$$

then for $i = 2$ we have

$$G_2(z) = \frac{z}{(1-z)^2}. \tag{7.4}$$

Now for the inductive case, supose that $G_j(z) = \frac{z}{(1-z)^2}(\frac{1}{1-z})^{j-2}$ for $j \leq k$. For $j = k+1$ we have

$$G_{k+1}(z) = z \sum_{i=1}^{k+1} G_i(z), \tag{7.5}$$

then

$$\begin{aligned}G_{k+1}(z) &= \frac{z}{(1-z)}\sum_{i=1}^{k} G_i(z) = \frac{z}{(1-z)^2} + \frac{z^2}{(1-z)^3}\sum_{i=0}^{k-2}\left(\frac{1}{1-z}\right)^i \\ &= \frac{z}{(1-z)^2}\left(1 + \frac{z}{1-z}\left(\frac{\left(\frac{1}{1-z}\right)^{k-1} - 1}{\frac{1}{1-z} - 1}\right)\right) = \frac{z}{(1-z)^2}\left(\frac{1}{1-z}\right)^{k-1}.\end{aligned} \tag{7.6}$$

Now, we have that the size of the tree constructed by Algorithm 6 is given by the d_{max}-th Taylor coefficient of $\sum\limits_{i=1}^{k} G_i(z)$. We have that

$$\begin{aligned}\sum_{i=1}^{k} G_i(z) =& \frac{1}{1-z} + \frac{z}{(1-z)^2}\sum_{i=0}^{k-2}\left(\frac{1}{1-z}\right)^i \\ =& \frac{1}{1-z} + \frac{z}{(1-z)^2}\left(\frac{\left(\frac{1}{1-z}\right)^{k-1} - 1}{\left(\frac{1}{1-z}\right) - 1}\right) = \left(\frac{1}{1-z}\right)^k.\end{aligned} \tag{7.7}$$

Then, the Taylor d-th coefficient of (7.7) is given by $\begin{pmatrix} d+k-1 \\ k-1 \end{pmatrix}$. Now, as the algorithm has to reach a depth of d_{max}, the size of the tree built by Algorithm 6 is bounded by $\begin{pmatrix} d_{max}+k-1 \\ k-1 \end{pmatrix}$.

□

Corollary **12** *If α is replaced by the value $2^{\mu} d_{max}$, then Algorithm 6 does not change its self maintainance verification property.*

7.4 d_{max} is bounded by Integer Linear Programming

Integer Linear Programming corresponds to solve the following problem:

$$\begin{gathered}\text{Maximize } \mathbf{c^T x}, \\ \text{Subject to } \mathbf{Ax} \leq \mathbf{b}.\end{gathered} \tag{7.8}$$

Where all the coordinates of $\mathbf{x}$ are integers. Thus, ILP consist in two problems, first verify the existence of a solution $\mathbf{x}$, and then find the $\mathbf{x}$ which maximizes $\mathbf{c^T x}$ in the space of solutions. If the coefficients of $\mathbf{A}$ and $\mathbf{b}$ are integers we say that $\mathbf{Ax} \leq \mathbf{b}$ is a Diophantine inequality. Borosh and Trevig [7, 8] developed criterias to bound the absolute value of the coefficients of the solutions of a Diophantine system of equations from the determinants of the submatrices (minors) of $\mathbf{A}$. Rackoff in [38] used the results of Borosh and Trevig to bound the complexity of the coverability and boundedness problem in VAS (see Section 5.2). Later Rosier and Chen in [45] developed a multiparameter analysis to give tight bounds to the boundedness problem. Based on those ideas we are going to bound d_{max}.

We are going to show that d_{max} is bounded by the minors of the stoichiometric matrix $\mathbf{S}$ and the minors of the augmented matrix $(\mathbf{S}, -\mathbf{S1})$, where $\mathbf{S}$ is the stoichiometric matrix assosiated to the AC Q and $\mathbf{1}$ is the $k-$dimensional vector with all the coordinates equal to one (see Corollary 9). Finally by using the results of Johnson and Newmann [28] we are going to bound the value of those minors in order to obtain a bound for the depth d_{max} of the tree builded by Algorithm 6.

From [8] we have the following theorem.

THEOREM **9** **(Bounding solution of Linear system of equations)** *Let* $\mathbf{Ax} = \mathbf{b}$ *a system of* $n \times m$ *diophantine equations. Assume the rows of* $\mathbf{A}$ *are linearly independent and denote by* X *(respectively* Y*) the maximum of the absolute values of the* $n \times n$ *minors of the matrix* $\mathbf{S}$ *(respectively the augmented matrix* $(\mathbf{A}, \mathbf{b})$*). If the system has a solution in nonnegative integers, then the system has a solution* $\mathbf{x} = (x_i)$ *in nonnegative integers with* $x_i \leq X$ *for* $m - n$ *variables and* $x_i \leq (n - m + 1)Y$ *for* n *variables.*

Then we have the following corollary:

COROLLARY **13** *Let* $\mathbf{Ax} \geq \mathbf{b}$*, a system of* $n \times m$ *diophantine inequations. Assume the rows of* $\mathbf{A}$ *are linearly independent and denote by* X *(respectively* Y*) the maximum of the absolute values of the* $n \times n$ *minors of the matrix* $\mathbf{A}$ *(the augmented matrix* $(\mathbf{A}, \mathbf{b})$*). If the system has a solution in nonnegative integers, then the system has a solution* $\mathbf{x} = (x_i)$ *in nonnegative integers with* $x_i \leq \max((m+1)\bar{Y}, \bar{X})$ *where* $\bar{X}$ *is the maximum between the* $n - i$ *minors of the matrix* $\mathbf{A}$ *(*$\bar{Y}$ *is the maximum of the* $n - i$ *minors of the augmented matrix* $(\mathbf{A}, \mathbf{b})$*)* $i = 1, ..., n-1$*, and one.*

Proof Note that

$$\mathbf{Ax} \geq \mathbf{b} \Leftrightarrow \mathbf{A}'\mathbf{t} = \mathbf{0},$$

where $\mathbf{A}'$ is the resulting matrix by catenate $\mathbf{A}$ with the $n \times n$ diagonal matrix $\mathbf{B}$ defined by

$$\mathbf{B}_{ij} = \begin{cases} -\frac{\mathbf{b}_i}{|\mathbf{b_i}|}, & \text{if } i = j \\ 0, & \text{else} \end{cases}$$

.

Now we can apply Theorem 9 to the system $\mathbf{A}'\mathbf{t} = \mathbf{0}$. We have that the rank of $\mathbf{A}'$ is n, then we have $\mathbf{t}_i \leq X'$ for $m + n - n = m$ variables and $\mathbf{t}_i \leq (m+1)Y'$ for n variables, where X' and Y' are the maximum of the determinants of the $n \times n$ minor associated to $\mathbf{A}'$. In this case $X' = Y'$ because the system $\mathbf{A}'\mathbf{t} = \mathbf{0}$ is homogeneous.

By the Laplace rule to calculate determinants the maximum of the $n \times n$ minors $X' \leq \bar{X}$ ($Y' \leq \bar{Y}$ respectively), with $\bar{X} = \max(X, X_1, ..., X_{n-1}, 1)$ ($\bar{Y} = \max(Y, Y_1, ..., Y_{n-1}, 1)$ respectively), where X_i (Y_i respectively) is the maximum of the $n - i \times n - i$ minors of $\mathbf{A}$ (the augmented matrix $(\mathbf{A}, \mathbf{b})$ respectively). This follows from the fact that catenate $\mathbf{b}$ to $\mathbf{A}$ can lead the case that the maximum absolute value of some $n - i$ minor X_i (Y_i respectively) of $\mathbf{A}$ ($(\mathbf{A}, \mathbf{b})$ respectively) has a bigger absolute value than X (Y respectively) for some $1 \leq i \leq n$, and thus the maximum of the values of the $n \times n$ minors of the catenate matrix would be X_i (Y_i respectively).

Then, as we do not know in principle which values of $\mathbf{t}$ are associated to the variables x_i or y_i, but we know that $X \leq \bar{X}$ and $Y \leq \bar{Y}$ we have that $\mathbf{t}_i \leq \max((m+1)\bar{Y}, \bar{X})$.

□

Now we are going to recall a theorem to bound the value of the determinant of a matrix, the result is from [28].

Theorem **10 (Bound on the determinant of a matrix)** *Let $A = [a_{ij}]$ be a $n \times n$ matrix. For each $1 \leq i \leq n$ set*

$$R_i^+(A) = \sum_{1 \leq j \leq n, a_{ij} > 0} a_{ij}, \quad R_i^-(A) = -\sum_{1 \leq j \leq n, a_{ij} \leq 0} a_{ij}$$

Then, $|det(A)| \leq \prod_{i=1}^{n} \max(R_i^+(A), R_i^-(A)) - \prod_{i=1}^{n} \min(R_i^+(A), R_i^-(A))$.

Now, combining results from Corollary 13 and Theorem 10 we have

Theorem **11 (Depth of the generalized Karp and Miller tree)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $\mathcal{Y} \subseteq \mathcal{M}$, $|\mathcal{Y}| = r$ and $|\mathcal{R}_{\mathcal{Y}}| = q$. For $R_i = A \to B \in \mathcal{R}_{\mathcal{Y}}$, with $i = 1, ..., q$. Let*

$$\max\left(\sum_{i=1}^{n} \mathcal{A}(A, m_i), \sum_{i=1}^{n} \mathcal{A}(B, m_i)\right) = p_i,$$

and s_i the decreasing ordered sequence of those p_i. Then the depth d_{max} of the tree defined by Algorithm 6 to verify the self-maintainance of $\mathcal{Y}$ is bounded by $q^2 \prod_{i=1}^{r} s_i$.

Proof By Corollary 11 it is sufficient prove that $d_{max} \geq |\mathbf{v}|_{\mathbf{1}}$ for some $\mathbf{v} = (v_1, ..., v_q)$ such that $\mathbf{S}_{\mathcal{Y}}\mathbf{v} \geq -\mathbf{S}_{\mathcal{Y}}\mathbf{1}$.
By Corollary 13 we have that $v_i \leq \max(\bar{X}, q\bar{Y})$ for $i = 1, ..., q$, where $\bar{X}$ ($\bar{Y}$ respectively) is the maximum of the absolute values of the determinant of the $r \times r$ minors of the matrix $\mathbf{S}_{\mathcal{Y}}$ (the augmented matrix $(\mathbf{S}_{\mathcal{Y}}, \mathbf{S}_{\mathcal{Y}}\mathbf{1})$ respectively).
On the other hand, from Theorem 10 we have that $\bar{X}$ and $\bar{Y}$ are both bounded by $\prod_{i=1}^{r} s_i$.

Finally we have $|v_i| \leq q \prod_{i=1}^{r} s_i$. Thus $|\mathbf{v}|_1 \leq d_{max} = q^2 \prod_{i=1}^{r} s_i$.

□

Corollary **14** *The element α defined in definition 58 is finite for every network.*

Chapter 8

Studying structural properties of organisations

The main goal of the static analysis of Algebraic Chemisitry is to obtain the organisational structure, *i.e* the reduced set of candidates to be the fixed points of the dynamics (see Theorem 1). In this chapter we are going to study in detail some underlying properties of organisations. We are going to reveal the diverse *roles* of molecules inside the organisation. This would lead to a better understanding of the process of computing the organisational structure.

We are going to revisit concepts defined in Chapter 2 and Chapter 3. We are going to define some special types of sets and we are going to prove that any organisation can be uniquely decomposed as an almost non-overlapping union of this special types of sets.

8.1 Reactivity revisited

Extending the ideas of F.Centler *et.al* [10] detailed in Chapter, 3, we are going to study the reactivity of molecules. We are going to generalize this notions in order to understand how the organisations emerges in an AC.

Looking at the definition of reactive sets of Chapter 3, we see the definition is self-supported in the sense that in order to verify the reactivity of a set it is only necessary to look at the set and its firable reaction set. It is purposed a generalization, the molecules are or are not reactive *with respect to a set.*

Definition **63 (Reactivity and enzymes w.r.t a set)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC,* $X \subseteq \mathcal{M}$ *and* $m \in \mathcal{M}$*. Let* $\mathcal{R}_X^m = \mathcal{R}_{X \cup \{m\}}$.

- *m is non-reactive w.r.t X iff for all reactions $R = A \to B \in \mathcal{R}_X^m$, m is present neither in A nor in B.*

- *m is an enzyme w.r.t X iff for some reaction $R' = A' \rightarrow B' \in \mathcal{R}_X^m$, m is present in A and for all reaction $R = A \rightarrow B \in \mathcal{R}_X^m$, $\mathcal{A}(A, m) = \mathcal{A}(B, m)$.*
- *m is reactive w.r.t X iff for some reaction $R' = A' \rightarrow B' \in \mathcal{R}_X^m$, m is present either in A' or in B' and for some reaction $R = A \rightarrow B \in \mathcal{R}_X^m$, $\mathcal{A}(A, m) \neq \mathcal{A}(B, m)$.*

We say that Y is a non-reactive, enzymatic or reactive set w.r.t X, if for all $m \in Y$, m is non-reactive, enzyme or reactive w.r.t X respectively.

The following lemma states that for every set of molecules X, every molecule m is related to X in one and only one of the senses defined above.

LEMMA **19** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC. Let $X \subseteq \mathcal{M}$ and $m \in \mathcal{M}$. Then m has one and only one of the following properties:*

1. *m is non-reactive w.r.t X.*
2. *m is an enzyme w.r.t X.*
3. *m is reactive w.r.t X.*

Proof Note that definition 63 include all the possible cases. The only case which is not expliciteły considered in the definition is when for some reaction $R = A \rightarrow B$, m is present in B and $\mathcal{A}(A, m) = \mathcal{A}(B, m)$, but this means that $\mathcal{A}(A, m) > 0$, and thus m is present in A, then m is an enzyme.

□

DEFINITION **64 (Enzyme, reactive and non-reactive set)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $Y \subseteq X \subseteq \mathcal{M}$. If Y is the maximal cardinality non-reactive, enzymatic or reactive set w.r.t X we say Y is the non-reactive, enzymatic or reactive set of X respectively.*

Remark: The inflow and outflow of a reaction network is always reactive. Note also that for a given set of molecules X, if we add to X molecules which are non-reactive molecules w.r.t X, we do not change the firable reaction set $\mathcal{R}_X$, and thus the closure property of X, or the (semi-)self-maintainance of X would not to be changed. Enzymatic and reactive molecules can change the firable reaction set of X.

The last definitions of this section recall some classic definitions in Petri nets, the set of input transitions of a place, and the set of input places of a transition. The size of the input transitions of a set of places, and the size of the set of input places of a set of transitions are fundamental topological parameters in the study of Petri nets. [36].

Type	Closure	Semi-self-maint.	Self-maint.
Non-reactive	maintained	maintained	maintained
Enzyme	potentially lost	maintained	potentially lost
Reactive	potentially lost	potentially lost	potentially lost

Figure 8.1: The table represents the maintainance or the posible losing of a property of a set of molecules once an external molecule of the types specified in the table is added to the set.

Definition **65 (Activable reactions of a molecule in a reaction set)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC and* $m \in \mathcal{M}$. *We define*

$$Act(m, \mathcal{R}) = \{R : \ R = A \to B \in \mathcal{R} \ s.t \ m \in A\}.$$

We say $Act(m, \mathcal{R})$ *is the activable set of reactions of* m *in* $\mathcal{R}$.

Definition **66 (Required set of molecules by a set of reactions)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC and* $R = A \to B \in \mathcal{R}$. *We define*

$$Req(R) = \{m : \ m \text{ is present in } A\}.$$

We say $Req(R)$ *is the required set of molecules of* R. *Furthermore, for a sequence of reactions* $S = R_1 \cdots R_k$ *we define* $Req(S) = \bigcup_{i=1}^{k} Req(R_i)$.

The set of input places in PN corresponds to the set Req($\cdot$), and the set of input transitions corresponds to Act($\cdot, \mathcal{R}$). We extend the notion of input transitions, by letting as parameter the set of transitions from where the input transitions are extracted.

8.2 Closure and Support of molecules

In this section we are going to study more in depth the generated closure (Definition 18), and we are going to introduce the notion of support of a set, which is the *inverse* relation of the closure. We are going to prove that the generated closure induces an equivalent relation into the supports of a set. Furthermore, by defining a union operator between sets in this equivalent class of supports we will prove that it leads to an operationally equivalent mathematical structure with respect to the set of all closed sets (the closed structure) together with the operation $\sqcup_{CL}$ (see Definition 18), and thus a semi-lattice.

We are going to redefine in a set-function like manner some properties of Chapter 2. For a detailed analysis in the set function-like analysis of reaction networks see [6].

Definition **67 (Producible and consumible set of a flux vector)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC, let* $X \subseteq \mathcal{M}$ *and* $|\mathcal{R}_X| = t$. *We define* $P^*(X) = \bigcup\limits_{i=1}^{t} P_X^{e_i}$, *where* e_i *is the i-th canonical vector of* $\mathbb{R}^t$ *and* $P_X^{\mathbf{v}}$ *is defined in definition 4. Analogously we define* $C^*(X) = \bigcup\limits_{i=1}^{t} C_X^{e_i}$. *We say* $P^*(X)$ *is the producible set of molecules by* X, *and* $C^*(X)$ *is the consumible set of molecules by* X.

The following Lemma is deduced from the definition of closure and semi self maintaining set, combined with Definition 67.

Corollary **15** *$P^*(X) \subseteq X$ iff X is a closed set.*
$C^*(X) \subseteq P^*(X)$ iff X is a semi-self maintaining set.

We now are going to define operators in order to study the generated closure.

Definition **68 (Produced and accumulated support)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $Y \subseteq \mathcal{M}$ a set of molecules. We define the following relations:*

$$Psupp(Y, X) \Leftrightarrow X \subseteq \mathcal{M} \text{ and } Y \subseteq G_{CL}(P^*(X)),$$

$$Asupp(Y, X) \Leftrightarrow X \subseteq \mathcal{M} \text{ and } Y \subseteq G_{CL}(X).$$

We say X is a Produced-Support or an Accumulated-Support of Y respectively.

Lemma **20** *$Psupp(Y, X)$ implies $Asupp(Y, X)$.*

Proof $Psupp(Y, X)$ implies $Y \subseteq G_{CL}(P^*(X))$. As $P^*(X) \subseteq G_{CL}(X)$, by Lemma 2 we have $G_{CL}(P^*(X)) \subseteq G_{CL}(G_{CL}(X)) = G_{CL}(X)$. Then $Y \subseteq G_{CL}(X)$, this is equivalent to $Asupp(Y, X)$.

□

Remark: It is interesting to note that $Psupp(Y, X)$ implies $Asupp(Y, X)$ but the backward implication is not necesarily true. For example let $Y = \{a\}$, $X = \{b\}$ and the only reaction $b \to a$. We have $Asupp(Y, X)$ but $Psupp(Y, X)$ is not true. The following is a simple monotonicity result for operators defined above.

Lemma **21 (Monotonicity 1)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$be an AC. Let $X, Y \subseteq \mathcal{M}$. Then $Asupp(Y, X)$ implies $Asupp(G_{CL}(Y), X)$.*

Proof Follows straigth from lemma 2.

□

As an example of the usefulness of these properties we restate some ideas about semi-organisations in the next lemma.

Lemma **22** *The following statements are true:*

1. *If $Psupp(Y, X)$ for some $X \subseteq Y$, then $G_{CL}(Y)$ is a semi-organisation.*
2. *$G_{CL}(Y)$ is a semi-organisation iff for some $X \subseteq G_{CL}(Y)$ we have $Psupp(Y, X)$.*

Proof *1*). Hypotesis implies by Lemma 2 that $G_{CL}(X) = G_{CL}(Y)$. Hypotesis also implies that $Psupp(X, X)$, this means X is semi-selfmaintaining and thus $G_{CL}(X)$ is a semi-organisation. *2*). $\leftarrow$: Follows from 1). $\rightarrow$: Let $X = C^*(G_{CL}(Y))$, it is trivial that $Psupp(Y, X)$ and $X \subseteq G_{CL}(Y)$.

□

Now we are going to study the equivalence of sets under closure.

Definition **69 (Equivalence relation induced by closure)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC and* $X, Y \subseteq \mathcal{M}$. *Let* $\cong_{CL}$ *the relation*

$$X \cong_{CL} Y \Leftrightarrow G_{CL}(X) = G_{CL}(Y). \tag{8.1}$$

We say $\cong_{CL}$ *is the closure equivalence induced by* G_{CL} *on* $\langle \mathcal{M}, \mathcal{R} \rangle$.

Lemma **23** *For every AC* $\langle \mathcal{M}, \mathcal{R} \rangle$, *the closure equivalence induced by* G_{CL} *on* $\langle \mathcal{M}, \mathcal{R} \rangle$ *is an equivalence relation.*

Proof The reflexivity and symmetry are trivial, the transitivity follows from lemma 21.

□

Corollary **16** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC. Let* $X_1, X_2 \subseteq Y \subseteq \mathcal{M}$. *We have* $Asupp(Y, X_1) \wedge Asupp(Y, X_2)$ *iff* $X_1 \cong_{CL} X_2$.

Proof Follows from Definition 68 and Definition 69.

□

The previous corollary shows that there is a conceptual equivalence between the closure equivalence relation $\cong_{CL}$ and $Asupp(\cdot, \cdot)$ relation. The following definition, theorem and corollary states this conceptual equivalence in more detail.

Definition **70 (Quotient space induced by closure)** *Let* $\mathcal{M}/\cong_{CL}$ *be the quotient space of* $\mathcal{M}$ *under the equivalence relation* $\cong_{CL}$ *defined in definition 69. For all* $X, Y \in \mathcal{M}/\cong_{CL}$ *we define* $X \cup Y$ *as the class of* $G_{CL}(X \cup Y)$.

Theorem **12 (Structural equivalence induced by the closure)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC. Let* $\mathcal{M}/\cong_{CL}$ *be the quotient space of* $\mathcal{M}$ *under the equivalence relation* $\cong_{CL}$ *defined in Definition 69 and* $\mathcal{C}$ *the set of all closed sets in* $\langle \mathcal{M}, \mathcal{R} \rangle$ *(the closed-structure). Then there exist a bijection* ψ *between* $\mathcal{M}/\cong_{CL}$ *and* $\mathcal{C}$ *which holds* $\psi(X \cup Y) = \psi(X) \sqcup_{CL} \psi(Y)$ *for every* $X, Y \in \mathcal{M}/\cong_{CL}$, *where* $\cup$ *is the set union defined in Definition 70 and* $\sqcup_{CL}$ *is defined in Definition 18.*

Proof We define

$$\begin{aligned}\psi :&(\mathcal{P}_M(\mathcal{M}),\cup) \mapsto (\mathcal{C},\sqcup_{CL})\\ &X \mapsto G_{CL}(X).\end{aligned}$$

As $\mathcal{C} \subseteq \mathcal{M}$ we have ψ is an epiyective function. Note that $\psi(X) = \psi(Y)$ iff $G_{CL}(X) = G_{CL}(Y)$. This means Ker(ψ) is equivalent to $\cong_{CL}$. By classical isomorphism theorem we have $\mathcal{M}/\cong_{CL} \equiv \mathcal{C}$.

□

COROLLARY **17** *Let $\langle\mathcal{M},\mathcal{R}\rangle$ be an AC. $(\mathcal{M}/\cong_{CL},\cup)$ is a semi-lattice.*

Proof Let $\mathcal{C}$ be the closed-structure. By Lemma 3 we have that $(\mathcal{C},\sqcup_{CL})$ is a semi-lattice. The result follow from Theorem 12.

□

Remark: Theorem 12 cannot be stated for intersection, because in general

$$G_{CL}(X \cap Y) \neq G_{CL}(G_{CL}(X) \cap G_{CL}(Y)).$$

For example consider the reaction network $\langle\{a,b,c\},\{a \to b, c \to b\}\rangle$, $X = \{a\}$ and $Y = \{c\}$.

8.3 Types of organisations

In this section we are going to define some fundamental types of organisations. We are going also to prove that every organisation can be seen as a non-overlapping union of these types of organisations.

8.3.1 Non-reactive

As we saw in Section 8.1, given a set X, we have a maximal set Y of molecules which are non-reactive w.r.t X. Now we are going to study the sets which are non-reactive w.r.t to themselves. We are going to prove that non-reactive structure is a meet semi-lattice w.r.t to intersection, this means that biggest non-reactive set *represents* the whole non-reactive structue. The lemmata of these section are trivially proved, but will be useful for further analysis.

LEMMA **24** *Let Y be a closed set. Verifying if X is non-reactive w.r.t Y takes linear time w.r.t $|\mathcal{R}_{X\cup Y}| + |X|$.*

Proof We only need to verify if for some $m \in X$ $Act(m, \mathcal{R}_{X \cup Y}) \neq \emptyset$.

$\square$

Remark: Every set which contains inflow or outflow molecules cannot be non-reactive.

The following lemma states that sets which are non-reactive w.r.t to themselves are organisations.

LEMMA **25** *Let X be a non-reactive set w.r.t to X. Then X is an organisation.*

Proof If X is non-reactive w.r.t X then $\mathcal{R}_X = \{\emptyset\}$, and thus X is an organisation.

$\square$

Remark: Suppose we have two sets X_1 and X_2 both non-reactive w.r.t to themselves, then the intersection $X_1 \cap X_2$ is non-reactive w.r.t to $X_1 \cap X_2$. But we cannot ensure the same for the union of sets, i.e $X_1 \cup X_2$ could or could not be non-reactive w.r.t $X_1 \cup X_2$, For example consider the network $\langle \{a, b\}, \{a + b \to \emptyset\} \rangle$, $\{a\}$ and $\{b\}$ are non-reactive, but they together form a reactive set w.r.t to itself. This leads the following result:

COROLLARY **18** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and*

$$\mathcal{N} = \{N : \ N \subseteq \mathcal{M} \text{ s.t } N \text{ is non-reactive w.r.t itself}\}.$$

Then $(\mathcal{N}, \cap)$ is a semi-lattice.

Non-reactive organisations are the simplest possible organisations, because they do not react at all. From an enviromental-methaphorical point of view the *non-reactive organisations are stones on the grass*. Note that non-reactive molecules are *potentially reactive* for new chemical components which could came to the enviroment.

8.3.2 Overproduced sets

The overproduced sets are composed by molecules which are able to be produced in a strictly higher quantity than they are consumed, without affecting the production of other molecules.

DEFINITION **71 (Overproduced molecule)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ an AC, and $t \in \{1, ..., n\}$ such that there exist a flux vector $\mathbf{v} = (v_1, ..., v_k)$ with $v_j \geq 0$ for $j = 1, ..., k$ such that $(\mathbf{Sv})_i = \mathbf{f}_i \geq 0$ for $i = 1, ..., n$, but $\mathbf{f}_t > 0$, where $\mathbf{S}$ is the stoichiometric matrix associated to $\langle \mathcal{M}, \mathcal{R} \rangle$. We say m_t is a potentially overproduced molecule in $\langle \mathcal{M}, \mathcal{R} \rangle$. If the flux vector $\mathbf{v}$ which verifies the potential overproduction of m_t holds $v_j > 0$ for $j = 1, ..., k$, we say m_t is overproduced in $\langle \mathcal{M}, \mathcal{R} \rangle$. Furthermore, a set of only potentially overproduced molecules is called a potentially overproduced set and a set of only overproduced molecules is called an overproduced set.*

Remark: To verify that a molecule is potentially overproduced in a network, it is not required to fire all reactions of the network because Definition 71 requires a flux vector $\mathbf{v} \geq \mathbf{0}$. But to verify that it is overproduced requires fire at least one time every reaction, i.e $\mathbf{v} > \mathbf{0}$.

Lemma **26** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $F \subseteq \mathcal{M}$. Verify if F is a potential overproduced set in $\langle \mathcal{M}, \mathcal{R} \rangle$ is a Linear Programming problem.*

Proof To verify the potential overproduction of F the system there has to be found a non-negative flux vector $\mathbf{v}$ such that for all molecule $m_i \in F$ we have $(\mathbf{Sv})_i = \mathbf{f}_i > 0$. This is the first (fesibility) part of a LP problem.

$\square$

Lemma **27** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $O \subseteq \mathcal{M}$. Let $m \in O$ a potentially overproduced molecule in $\langle O, \mathcal{R}_O \rangle$. If O is self maintaining, then m is overproduced in $\langle O, \mathcal{R}_O \rangle$.*

Proof Let $\mathbf{v}$ the flux vector which verifies the potential overproduction of m, and $\mathbf{v}^*$ the flux vector which verifies the self maintainance of O. Then $\mathbf{v} + \mathbf{v}^*$ verifies the overproduction of m in $\langle O, \mathcal{R}_O \rangle$.

$\square$

Definition **72 (Maximal overproduced set)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $O \subseteq \mathcal{M}$ an organisation. We say $F \subseteq O$ is a maximal overproduced set of O if F is an overproduced set of O and for all $F' \neq F$ overproduced set of O we have $|F| > |F'|$. If for all $m \in O$, m is overproduced in $\langle O, \mathcal{R}_O \rangle$, we say O is a totally overproduced organisation.*

Now we are going to prove that the maximal overproduced set is unique.

Lemma **28** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $O \subseteq \mathcal{M}$ an organisation. Then there exists a maximal overproduced set F in $\langle O, \mathcal{R}_O \rangle$ and is unique.*

Proof If there are no overproduced molecules in O then the maximal overproduced set is the empty set. In other case the union of all overproduced molecules in $\langle O, \mathcal{R}_O \rangle$ leads to a maximal overproduced set. Now we are going to prove that the maximal overyproduced set is unique. Suppose that there are two maximal overproduced sets $F_1, F_2 \in O$ where $F_1 \neq F_2$, and let $\mathbf{v_1}, \mathbf{v_2}$ the flux vectors required to verify the overproduced property of F_1 and F_2 w.r.t $\langle O, \mathcal{R}_O \rangle$ respectively. Trivially, $\mathbf{v}_1 + \mathbf{v}_2$ verifies the overproduced property of $F_1 \cup F_2$ w.r.t $\langle O, \mathcal{R}_O \rangle$, and $F_i \subset F_1 \cup F_2$ for $i = 1, 2$. As the inclusion is strict we have a contradiction.

$\square$

Remark The overproduced molecules are able to increase its amount infinitely in $\langle \mathcal{M}, \mathcal{R} \rangle$ via sequences of reactions (flux vector), without decreasing the amount of the other molecules in the network. In Algorithm 6, the overproduced molecules are identified by the ω element in the labelling of the nodes.

LEMMA **29 (monotonicity 2)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $X \subseteq \mathcal{M}$ and $m \in P^*(X)$. If X is potentially overproduced, then $X \cup \{m\}$ is potentially overproduced.*

Proof Let $m \in P^*(X)$. If $m \in X$ then m is potentially overproduced. If $m \notin X$, then $m \in P^*(X) - X$, then there exist a reaction R_j firable by X which produces the molecule m, *i.e* there exist $R_j = A \to B \in \mathcal{R}_X$ such that $\mathcal{A}(A, m) < \mathcal{A}(B, m)$. Now, let $\mathbf{v}_X$ a flux vector which verifies the potential overproducibility of X such that $(\mathbf{S_1}\mathbf{v}_X)_i = f_i \geq 1$ for all $i = 1, ..., |X|$, where $\mathbf{S_1}$ is the stoichiometric matrix associated to $\langle X, \mathcal{R}_X \rangle$. As R_j is firable by X and there exists $r = \max_{m' \in X} (\mathcal{A}(A, m'))$, then $r\mathbf{v}_X + \mathbf{v}^*$ verifies the potential overproducibility of $X \cup \{m\}$, where $\mathbf{v}^* = (v_1^*, ..., v_k^*)$ with

$$v_i^* = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j. \end{cases}$$

Then $X \cup \{m\}$ is a potentially overproduced set.

□

LEMMA **30** *The closure of the inflow in a reaction network is a totally overproduced organisation.*

The following corollaries state that potentially overproduced sets can be considered as inflow in the network.

COROLLARY **19** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $X \subseteq \mathcal{M}$. If X is potentially overproduced in $\langle \mathcal{M}, \mathcal{R} \rangle$, then the closure $G_{CL}(X)$ is potentially overproduced in $\langle G_{CL}(X), \mathcal{R}_{G_{CL}(X)} \rangle$.*

COROLLARY **20** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $Y \subseteq \mathcal{M}$. If $X \subseteq Y$ is potentially overproduced in $\langle Y, \mathcal{R}_Y \rangle$, then X is potentially overproduced in $\langle \mathcal{M}, \mathcal{R} \rangle$.*

The following corollary states an interesting property: potential overproduced support contained in a set implies that the closure of the set is totally overproduced.

COROLLARY **21** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $X \subseteq Y \subseteq \mathcal{M}$. If Asupp$(Y, X)$ and X is a potential overproduced set of $\langle Y, \mathcal{R}_Y \rangle$, then $G_{CL}(Y)$ is a totally overproduced organisation.*

Proof The proof follows from Lemma 2 and Corollary 19.

□

The results presented above lead to a very important conclusion: The task of obtaining the organisational strucutre can be simplified by including the overproduced sets in the analysis, because once a overproduced set $F \subseteq \mathcal{M}$ is identified, every set G s.t $F \subseteq G$ will contain F as a potential overproduced set in it (see Lemma 20), and thus the verification of the self-maintainance of G does not requires to be concerned on the production of molecules in F. This means it has to be solved a LP problem with $|\mathcal{R}_G| + |G| - |F|$ equations instead of a LP problem of $|\mathcal{R}_G| + |G|$ equations (as it is shown in Corollary 1).

Corollary **22** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $X \subseteq Y \subseteq \mathcal{M}$. If X is overproduced in $\langle Y, \mathcal{R}_Y \rangle$, then to verify if Y is an organisation is an LP problem of $|\mathcal{R}_Y|$ variables and $|\mathcal{R}_Y| + |Y| - |X|$ equations.*

Remark: Note that the amount of equations in Corollary 22 is $|\mathcal{R}_Y| + |Y| - |X|$, but $|\mathcal{R}_Y|$ equations are trivial because those equations defines the flux vector as a non-negative one.

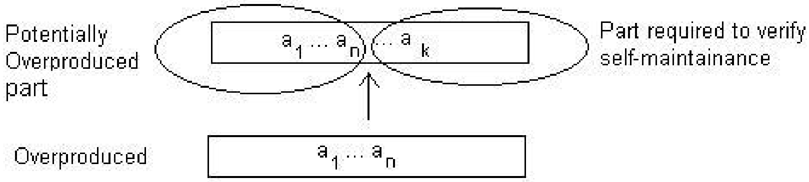

Figure 8.2: The figure shows the above results in a Hasse diagram picture. The set $\{a_1, ..., a_n\}$ is overproduced, and thus is potentially overproduced w.r.t $\{a_1, ..., a_n, ..., a_k\}$, thus to verify its self-maintainance it is not required to be concerned on the production of $\{a_1, ..., a_n\}$.

8.3.3 Cycles

Definition **73 (Cyclic set)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $\mathcal{O} \subseteq \mathcal{M}$ an organisation and F the maximal overproduced set of $\mathcal{O}$. We define the cyclic set of $\mathcal{O}$ as the maximal cardinality set without non-reactive molecules in $\mathcal{O} - F$.*

Remark: Note that the cycle of an organisation is unique because the non-reactive set is trivially unique and the maximal overproduced set is also unique (see Lemma 28).
Remark: Given a flux vector which verifies the self-maintainance of $\mathcal{O}$. The cyclic set has production rate equal to zero. But the cycle should not be confused with the set of molecules with production rate equal to zero, because the non-reactive molecules also has production rate equal to zero, but those molecules does not belong to the cycle set of $\mathcal{O}$. In this analysis we distiguish between the cyclic molecules and non-reactive molecules.

The following lemma states that the molecules in the cyclic set have to be either enzyme molecules or *consumed and produced* by some reactions. This means that no molecule can be only produced or only consumed in a cyclic set.

Lemma **31** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $\mathcal{O} \subseteq \mathcal{M}$ an organisation, let C be the cyclic set of $\mathcal{O}$. Then for every $m \in C$ we have either:*

1. *m is an enzyme w.r.t $\mathcal{O}$, or*
2. *m is consumed by some reaction $R \in \mathcal{R}_{\mathcal{O}}$ and produced by other reaction $R' \in \mathcal{R}_{\mathcal{O}}$.*

Proof Let $m \in C$, then m cannot be non-reactive. By Lemma 19 we have m is an enzyme or a reactive molecule w.r.t C. If m is an enzyme w.r.t to $\mathcal{O}$ we are in case 1). If m is reactive w.r.t $\mathcal{O}$, there exist a reaction $R = A \to B \in \mathcal{R}$ s.t $\mathcal{A}(A, m) \neq \mathcal{A}(B, m)$. If $\mathcal{A}(A, m) > \mathcal{A}(B, m)$, as $\mathcal{O}$ is an organisation, there has to exist some reaction $R' = A' \to B'$ s.t $\mathcal{A}(A', m) < \mathcal{A}(B', m)$, we are in case 2). If $\mathcal{A}(A, m) < \mathcal{A}(B, m)$, as m is not overproduced (because belongs to the cyclic set), there has to exist some reaction $R' = A' \to B'$ s.t $\mathcal{A}(A', m) > \mathcal{A}(B', m)$, we are in case 2).

$\square$

Definition **74 (Active cycle)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $O \subseteq \mathcal{M}$ an organisation and $m \in O$ a molecule in the cycle set of O. If m is not an enzyme w.r.t O we say m is a cycle active molecule w.r.t O. A set of cycle active molecules w.r.t O is called a set cycle active w.r.t O.*

Lemma **32** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, let O be an organisation in $\langle \mathcal{M}, \mathcal{R} \rangle$. Then there exist a unique decomposition of the cycle $C = E \cup D$ w.r.t O, where E is a set of enzyme molecules of O and D is a set cycle active. Furthermore $E \cap D = \emptyset$.*

Proof The proof follows straight from Lemma 31.

$\square$

Definition **75 (Potential active cycle)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $X \subseteq \mathcal{M}$. Let N, E be the non-reactive and enzymatic set w.r.t X. Let F a potentially overproduced set w.r.t X. We say that $X - (F \cup E \cup N)$ is the potential active cycle of X w.r.t F, or PAC of X w.r.t F.*

8.3.4 PAC and dependent connectivity

In this section we are going to study the PACs. We are going to define a special notion of connectivity which would allow us to separate the PAC of a set in partially non-overlapping sub-PACs, such that the self maintainance of the PAC of a set can be studied from the self maintainance of the sub-PACs.

Definition **76 (Dependent connected molecules)** *Let $Q = \langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $X \subseteq \mathcal{M}$, F be a potential overproduced set w.r.t X and E the enzymatic set w.r.t X. Two species s_i and s_j in X are dependent connected in $\langle X, R_X \rangle$ w.r.t E and F iff there exists a sequence of species $s_0, ..., s_p \in X - (E \cup F)$ such that $s_i = s_0$, s_k and s_{k+1} are directly connected (see Definition 27) in $\langle X, R_X \rangle$ for all $k = 0, ..., p-1$ and $s_p = s_j$.*

The notion of dependent connection is a restriction to the notion of connection stated in Definition 28. This restriction is oriented to capture the minimal part of the network which is responsible for the self maintainance of a set.

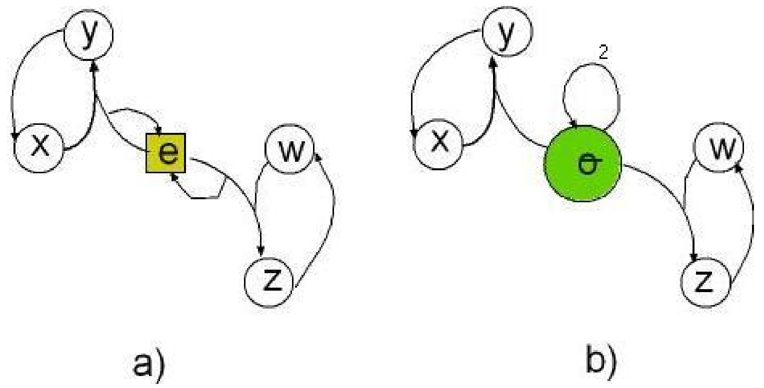

Figure 8.3: Note that in both figures all the molecules are connected following Definition 28. Note that in figure **a**) the self maintainance of $C_1 = \{x, y, e\}$ and $C_2 = \{z, w, e\}$ are independent because the molecule which connects C_1 with C_2 is an enzyme w.r.t $C_1 \cup C_2$. The same situation occurs in figure **b**) for sets $C'_1 = \{x, y, o\}$ and $C'_2 = \{z, w, o\}$ because they are connected by a potentially overproduced molecule w.r.t $C'_1 \cup C'_2$. That is why there is no *dependent connected* molecules neither in C_1 with molecules in C_2 nor in C'_1 with molecules in C'_2.

LEMMA **33** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $X \subseteq \mathcal{M}$, E be the enzymatic set w.r.t X, F an overproduced set w.r.t X and $m, \bar{m} \in \mathcal{M}$. m is dependent connected in $\langle X, \mathcal{R}_X \rangle$ w.r.t E and F to $\bar{m}$ iff $\bar{m}$ is dependent connected in $\langle X, \mathcal{R}_X \rangle$ w.r.t E and F to m.*

Proof Follows from the definition of dependent connected molecules.

□

DEFINITION **77 (Causal set of a molecule w.r.t sets of reactions)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $X \subseteq \mathcal{M}$, E be the enzymatic set w.r.t X, F a potentially overproduced set w.r.t X and $m \in X - (E \cup F)$. We define $Causal^*(m, \mathcal{R}_X)$ as the set of dependent connected molecules in $\langle X, \mathcal{R}_X \rangle$ w.r.t E and F to m. We define $Causal(m, \mathcal{R}_X) = Req(Act(Causal^*(m, \mathcal{R}_X)))$. We say $E^* = E \cap Causal(m, \mathcal{R}_X)$ and $F^* = F \cap Causal(m, \mathcal{R}_X)$ are the set of non-enzyme and the overproduced molecules w.r.t $Causal^*(m, \mathcal{R}_X)$ respectively.*

The following corollaries are derived from Lemma 33 and Definition 77.

COROLLARY **23** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $X \subseteq \mathcal{M}$, E be the enzymatic set w.r.t X, F an overproduced set w.r.t X and $m, \bar{m} \in \mathcal{M}$. $\bar{m} \in Causal^*(m, \mathcal{R}_X)$ iff $Causal^*(m, \mathcal{R}_X) = Causal^*(\bar{m}, \mathcal{R}_X)$.*

COROLLARY **24** *Let $R \in \mathcal{R}_X$, $m, \bar{m} \in X - (E \cup F)$ s.t $m \notin Causal(\bar{m}, \mathcal{R}_X)$. If $R \in Act(Causal(m, \mathcal{R}_X) \cap Causal(\bar{m}, \mathcal{R}_X), \mathcal{R}_X)$ then $R \in \mathcal{R}_{E \cup F}$.*

$Causal^*(\cdot, \cdot)$ provides a way to split a set X of molecules in dependent connected subsets. It is necessary first to determine the enzymatic set E and an overlyproduced set F w.r.t X in order

to generate such separation. Note that the enzymatic set is unique for every X, but we can choose different potentially overproduced sets to split X by using $Causal^*(\cdot,\cdot)$. Depending on how F was chosen, the separation of X would be different. As much elements are in F, more separable would be X. We are going to study at which point the results derived from the notions introduced in this section simplify the understanding of what constitutes an organisation.

LEMMA **34** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $X \subseteq \mathcal{M}$, E be the enzymatic set of X, F an overproduced set w.r.t X and D the active cycle of X w.r.t F. Then*

$$D = \bigcup_{m \in D} Causal^*(m, \mathcal{R}_X)$$

Proof Note that $D \subseteq \bigcup_{m \in D} Causal^*(m, \mathcal{R}_X)$. Let $m \in \bigcup_{m' \in D} Causal^*(m', \mathcal{R}_X)$, then for some molecule $m' \in D$ we have m is dependent connected to m', then m' is also dependent connected to m. This means m is a reactive, non over produced, and non-enzymatic molecule. Then $m \in D$.

□

DEFINITION **78 (Base and minimal base of Active cycles)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $X \subseteq \mathcal{M}$, E be the enzymatic set w.r.t X, F an overproduced set w.r.t X and D the active cycle of X w.r.t F. Any set $D' \subseteq D$ s.t $D = \bigcup_{m \in D'} Causal^*(m, \mathcal{R}_X)$ is called a base of D. Any minimal cardinality base of D is called a minimal base of D.*

LEMMA **35** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC, $X \subseteq \mathcal{M}$, E be the enzymatic set w.r.t X, F an overproduced set w.r.t X and D the active cycle of X w.r.t F. Let D', D'' be minimal bases of D. Then every molecule in D' is dependent connected to one and only one molecule of D''.*

Proof Let $m \in D'$ and suppose that there is no molecule in D'' dependent connected to m. By Corollary 23 we have $Causal^*(m, \mathcal{R}_X)$ is not contained in $\bigcup_{m' \in D''} Causal^*(m, \mathcal{R}_X)$. Then it has to be at least one molecule dependent connected to m in D''. Now suppose there is more than one molecule dependent connected to m in D''. Let $m_1, m_2 \in D''$ such molecules. As m_1 and m_2 are dependent connected to m, then m_1 and m_2 are dependent connected. By Corollary 23 we have $Causal^*(m_1, \mathcal{R}_X) = Causal^*(m_2, \mathcal{R}_X)$. Then D'' is not a minimal base of D.

□

A minimal base of D is a set of the representative non dependent sub-PACs of the cycle. We are going to prove that the self maintainance of the cycle can be obtained from the self maintainance of this non dependet sub-PACs.

Lemma **36** *Let $\langle \mathcal{M}, \mathcal{R}\rangle$ be an AC, $X \subseteq \mathcal{M}$, E be the enzymatic set of X, F an overproduced set w.r.t X, D the active cycle of X w.r.t F and D' a minimal base of D. Then*

$$Act(D, \mathcal{R}_X) = \bigcup_{m\in D'} Act(Causal^*(m, \mathcal{R}_X), \mathcal{R}_X)$$

Proof Is clear that $Act(D, \mathcal{R}_X) \supseteq \bigcup_{m\in D'} Act(Causal^*(m, \mathcal{R}_X), \mathcal{R}_X)$. Let $R \in Act(D, \mathcal{R}_X)$ then for some $m \in D$ we have $R \in (m, \mathcal{R}_X)$. From Definition 78 we know that there exist $m' \in D'$ s.t $m \in Causal^*(m', \mathcal{R}_X)$. Then by Corollary 23 we have

$$R \in Act(Causal^*(m', \mathcal{R}_X), \mathcal{R}_X) \subseteq \bigcup_{m\in D'} Act(Causal^*(m, \mathcal{R}_X), \mathcal{R}_X).$$

$\square$

Theorem **13 (Redefining self-maintainance from** *role* **of molecules)** *Let $\langle \mathcal{M}, \mathcal{R}\rangle$ be an AC, $X \subseteq \mathcal{M}$, E be the enzymatic set w.r.t X, F an overproduced set w.r.t X, D the active cycle of X w.r.t F and $D' = \{\bar{m}_1, ..., \bar{m}_d\}$ a minimal base of D. For $i = 1, ..., d$ let*

$$\begin{aligned} D_i &= Causal(\bar{m}_i, \mathcal{R}_X), \\ F_i^* &= Causal(\bar{m}_i, \mathcal{R}_X) \cap F \text{ and} \\ E_i^* &= Causal(\bar{m}_i, \mathcal{R}_X) \cap E. \end{aligned}$$

Let $\{\emptyset \to Y\} = \{\emptyset \to y$ for each $y \in Y\}$. X is self maintaining iff for all $i = 1, ..., d$ we have that D_i is is self maintaining in the subnetwork $\langle D_i, \mathcal{R}_{D_i} \cup \{\emptyset \to F_i^*\}\rangle$.

Proof $\Rightarrow$: As F is potentially overproduced then X is self maintaining in $\langle X, \mathcal{R}_X\rangle$ iff $\langle X, \mathcal{R}_X \cup \{\emptyset \to F\}\rangle$ is self maintaining. Let $\mathbf{v}^*$ a vector which verifies the self maintainance of X in $\langle X, \mathcal{R}_X\rangle$. Let $Act(D_i, \mathcal{R}_X) = \{R_{\alpha_1}, ..., R_{\alpha_l^i}\}$, then $\bar{\mathbf{v}}$ lead to a non-negative production on all the molecules of $Causal^*(m_i, \mathcal{R}_X)$ where

$$\bar{\mathbf{v}}_i = \begin{cases} \mathbf{v}_i^* & \text{if } i = \alpha_j \text{ for some } j, \\ 0 & \text{else} \end{cases}$$

As the rest of molecules belongs to F^*, to reach its non-negative production we use the reactions in $\{\emptyset \to F_i^*\}$.

$\Leftarrow$: Let $\mathbf{v_1}, ..., \mathbf{v_d}$ the flux vectors which verifies the self maintainance of $\langle D_i, \mathcal{R}_{D_i} \cup \emptyset \to F_i^*\rangle$, $i = 1, ..., d$ and $\mathbf{v}^F$ the flux vector which verifies the potential overproduction of F w.r.t X. Then there exist a non-negative β s.t $\beta \mathbf{v}^F + \sum_{i=1}^{d} \bar{\mathbf{v}}_\mathbf{i}$ verifies the self maintainance of X, where $\bar{\mathbf{v}}_\mathbf{i}$ is the vector $\mathbf{v_i}$ restricted to the coordinates corresponfing to $\mathcal{R}_X$.

$\square$

8.4 A Decomposition Theorem in AC

In this section we state the decomposition of organisations into non-reacitve, overproduced, enzymatic and cycle active sets, and the computational consecuences of this decomposition.

THEOREM **14 (Decomposition Theorem in AC)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$*be an AC,* $X \subseteq \mathcal{M}$*,* N *the non-reactive set of* X*,* E *be the enzymatic set w.r.t* X*,* F *an overproduced set w.r.t* X*,* D *the active cycle of* X *w.r.t* F *and* D' *a minimal base of* D*. Then*

$$X = N \cup E \cup \; F \bigcup_{m \in D'} Causal(m, \mathcal{R}_X).$$

COROLLARY **25** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC, let* $\mathcal{O}$ *be an organisation in* $\langle \mathcal{M}, \mathcal{R} \rangle$*. Then the decomposition of* $\mathcal{O}$ *stated in Theorem 14 is unique.*

Proof The maximal overproduced set, the non-reactive set and the enzyme set of $\mathcal{O}$ are unique. By Lemma 35 we have the decomposition of the cycle is also unique.

□

Theorem 25 permit understand the nature of the computation of organisations. The following corollary is the most important result of this thesis:

COROLLARY **26 (The most important result in this work)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC,* $X \subseteq \mathcal{M}$*,* N *the non-reactive set of* X*,* E *be the enzymatic set w.r.t* X*,* F *an overproduced set w.r.t* X*,* D *the active cycle of* X *w.r.t* F *and* D' *a minimal base of* D*. Verify if* X *is an organisation consist in solve* $|D|$ *Linear Programming problems, each problem* $i = 1, ..., |D|$ *of* $|\mathcal{R}_{D_i}|$ *variables and* $|\mathcal{R}_{D_i}| + |D_i|$ *equations.*

Remark: Corollary 26 it is proposed as the staring point to the development of new techniques to deal with Algebraic Chemistry as well as new algorithms to compute the organisational structure of reaction networks.

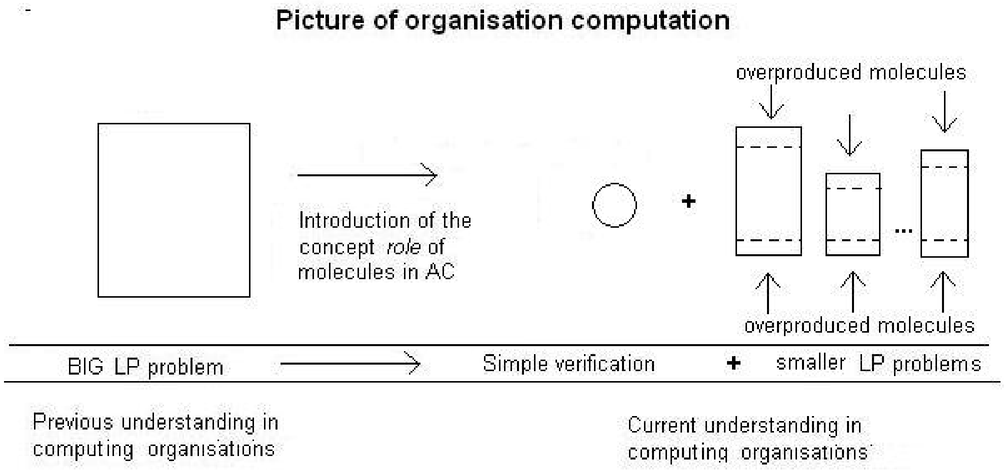

Figure 8.4: The picture shows the interest of the results of this Chapter. The problem of compute if a set X is an organisation (an LP problem) can be splitted in the a simple verification (the enzymatic and non-reactive sets) plus several smaller (w.r.t to the size of X) LP problems (a potential overproduced set and the self-maintainance of the PACs induced by the overproduced set and the enzymatic set).

Chapter 9

Structural properties revisited

In this chapter we are going to apply the results obtained in Chapter 8 to study some subsets of the organisational structure of a given network, and different types of networks in which its organisational structure holds some extra properties. We are going to prove that the totally-overproduced organisational structure is a lattice under some especial join and meet operators. We also will prove the differences and similiarities between consistent networks and non-active cycle networks.

9.1 Lattice of totally overproduced organisations

Following the definition of the generated (semi-)organisation done by Dittrich *et. al.* in [14], we are going to define the generated overproduced set of a set of molecules, the overproduced union and the overproduced intersection, in order to state structural properties for the totally overproduced organisation structure.

Definition **79 (Generated overproduced set)** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC,* $X \subseteq \mathcal{M}$ *and* $G_{OP}(X)$ *be the largest overproduced set contained in* $G_{CL}(X)$*. We say* X *generates the overproduced set* $\mathcal{Y} = G_{OP}(X)$*. Furthermore, we define the overproduced union and intersection of overproduced sets of molecules* $\mathcal{Y}_1, \mathcal{Y}_2 \subseteq \mathcal{M}$ *by:*
$\mathcal{Y}_1 \sqcup_{OP} \mathcal{Y}_2 = G_{OP}(\mathcal{Y}_1 \cup \mathcal{Y}_2)$*,*
$\mathcal{Y}_1 \sqcap_{OP} \mathcal{Y}_2 = G_{OP}(\mathcal{Y}_1 \cap \mathcal{Y}_2)$*.*

Remark: Note that the union and intersection operators above are only defined for overproduced sets.

Lemma **37** *Let* $\langle \mathcal{M}, \mathcal{R} \rangle$ *be an AC, and* $X, Y \subseteq \mathcal{M}$ *be two totally overproduced sets. Then* $X \sqcup_{OP} Y = X \sqcup_{CL} Y$*.*

Proof Note that as X and Y are totally overproduced sets, then by Lemma 21 we have $G_{CL}(X \cup Y) = X \sqcup_{CL} Y$ is a totally overproduced organisation.

$\square$

The following theorem states the good properties of the strictly-growing organisation structure.

Theorem **15 (Totally overproduced structure is a lattice)** *The totally overproduced structure is a lattice with respect to the operators $\sqcup_{OP}$ and $\sqcap_{OP}$.*

Proof Let $\mathcal{O}$ be the totally overproduced structure, and $X, Y \in \mathcal{O}$. We are going to prove

1. $X \sqcup_{OP} Y \in \mathcal{O}$ is the least upper bound for X and Y.
2. $X \sqcap_{OP} Y \in \mathcal{O}$ is the greatest lower bound of X and Y.

1): Follows trivially from the fact that closure-structure is a lattice with respect to $\sqcup_{CL}$ and Lemma 37.
2): By Definition 79 we have that $X \sqcap_{OP} Y$ is the greatest overproduced set contained in $G_{CL}(X \cap Y)$, we need to prove that $X \sqcap_{OP} Y \in \mathcal{O}$, this means that $X \sqcap_{OP} Y$ is closed. We are going to prove that $G_{CL}(X \sqcap_{OP} Y) = X \sqcap_{OP} Y$. As $Asupp(G_{CL}(X \sqcap_{OP} Y), X \sqcap_{OP} Y)$, we have from Theorem 21 that $G_{CL}(X \sqcap_{OP} Y)$ is overproduced. As $X \sqcap_{OP} Y \subseteq G_{CL}(X \sqcap_{OP} Y)$, and $G_{CL}(X \sqcap_{OP} Y) \subseteq X, Y$, then is $G_{CL}(X \sqcap_{OP} Y)$ the greatest overproduced set contained in $G_{CL}(X \cap Y)$, we have that $G_{CL}(X \sqcap_{OP} Y) = X \sqcap_{OP} Y$.

$\square$

In the next section we are going to explore the set of organisations which have no cyclic set.

9.2 Non-active cycle organisations and Consistent Networks

Definition **80 (Non-active cycle organisation)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC and $\mathcal{O} \subseteq \mathcal{M}$ be an organisation. We say $\mathcal{O}$ is a non-active cycle organisation if the active cycle set of $\mathcal{O}$ is empty. We also say that a network is a non-active cyclic network if and only if its organisational structure is equal to its non-active cycle organisatinal structure.*

The non-active cycle organisations are composed by a totally-overproduced set F, an enzyme set E w.r.t to F and a non-reactive set N w.r.t $F \cup E$.

As an example of the usefulness of the non-active cycle concept, we are going to prove the following:

- A restricted class of consistent networks called *reactive flow with persistent molecules* [14] is also a restricted class of non-active cycle networks.
- There exist an AC $\langle \mathcal{M}, \mathcal{R} \rangle$ which is a non-active cycle network but is not consistent, and there exist a network which is consistent but is not a non-active cycle network.

At this point we can state that non-active cycle networks and consistent networks are different, but its intersection is not empty. This suggests to study more in detail the definitions of consistent networks and non-active cycle networks in order to reveal in which way they differ.

Definition **81 (Reactive flow with persistent molecules)** *A reactive flow with persistent molecules is an AC $\langle \mathcal{M}, \mathcal{R} \rangle$ such that $\mathcal{M} = P \cup P'$ with $P \cap P' = \emptyset$ where:*

- *For $m \in P'$ there exist a reaction $R = m \to \emptyset \in \mathcal{R}$.*
- *For $m \in P$ there does not exist a reaction $R = A \to B$ such that $\mathcal{A}(A, m) > \mathcal{A}(B, m)$.*

We say P is the persistent molecules set and P' the non-persistent molecule set.

Lemma **38** *[14] A reactive flow with persistent molecules is consistent.*

Now we are going to prove the following:

Lemma **39** *In a reaction flow with persistent molecules $\langle \mathcal{M}, \mathcal{R} \rangle$, the cycle set of a self-maintaining set is empty.*

Proof Let X be a self maintaining set. We are going to prove that every molecule is overproducible or non-reactive. Let $m \in X$, if m is a persistent molecule, then as m is not consumed by any reaction, it can be either overproduced or non-reactive. If m is non-persistent, let $R^* = m \to \emptyset$. There must exist a flux vector in $\mathcal{R}_X - \{R^*\}$ which overproduces m, and thus by Corollary 20, m is overproduced.

□

Corollary **27** *Every reaction flow with persistent molecules is a non-active cycle network.*

Remark: Note that as the cycle set is empty the organisations in a Reaction flow with persistent molecules is composed only by overproduced and non-reactive molecules. Now we are going to show an AC consistence and non-active cycle networks are different:

Example 11. Let $Q = \langle \{a, b\}, \{a \to b, b \to a\} \rangle$ be an AC. The organisational structure is $\{\{\emptyset\}, \{a, b\}\}$, thus Q is consistent. Note that the organisation $\{a, b\}$ is reactive but not overproduced, thus Q is not non-active cycle.

Example 12. Let $Q = \langle \{a, b, c\}, \{a+b \to \emptyset, b+c \to \emptyset, b \to 2b, c \to 2c\} \rangle$. The organisational structure is $\{\{\emptyset\}, \{a\}, \{b\}, \{c\}, \{b, c\}\}$, thus Q is a non-active cycle netwotk, but the network is not semi-consistent, because the union $X \cup Y$ of (semi-)self-maintaining sets, where $X = \{a\}$ and $Y = \{b\}$ is not (semi-)self-maintaining, thus Q is not consistent. Furthermore, note that Q is not a reactive flow with persistent molecules either, because the molecules are no persistent nor persistent either.

Corollary **28** *Let X be the class of consistent AC, Y the class of non-active cycle AC and Z the class of reactive flow with persistent molecules networks, then $X \neq Y$ but $Z \subset X \cap Y$.*

Remark: Note the containment of Corollary 28 is strict.

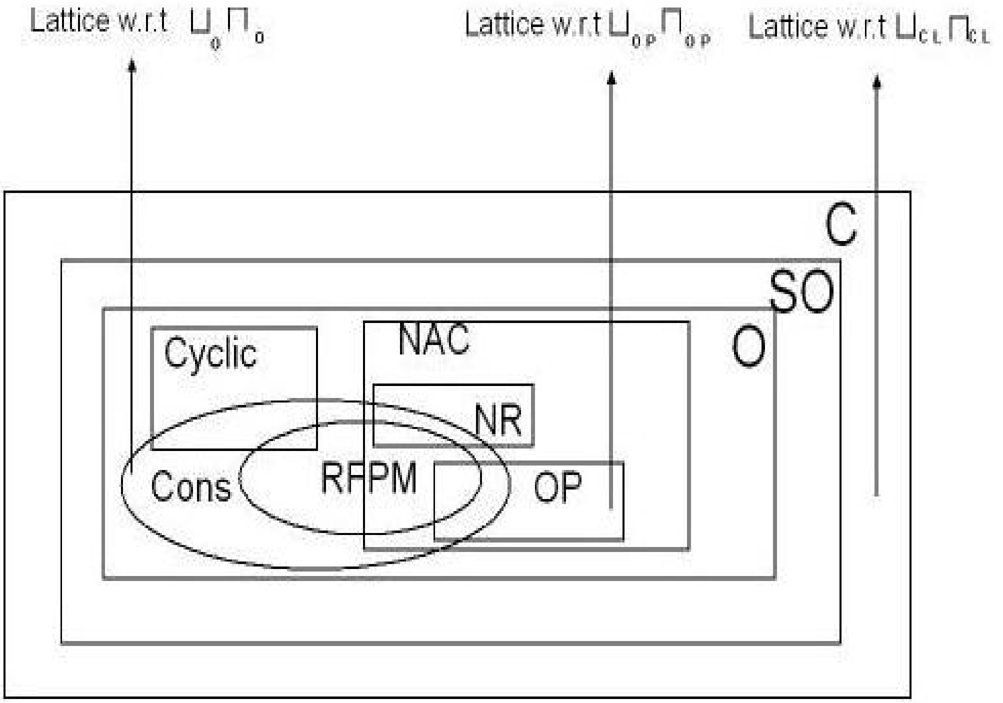

Figure 9.1: Containment of structures and properties. C is the closed-structure, SO is the semi-organisational-structure, O is the organisational structure, NR is the non-reactive-organisational structure, OP is the overproduced organisational structure, NAC is the non-active cycle organisational strucutre, Cyclic is the cycle organisational strucuture, RFPM are the reactive flows with persistent molecules and Cons are the class of consistent networks in this picture. Organisations in a consistent network can be of any type. Note that no cyclic organisation can be in NAC, also OP and NR are totally contained in NAC. Note that Cons and NAC has a non-empty intersection but no one of them is totally contained in the other and RFPM is stricly contained in the intersection of Cons and NAC. Furthermore C, OP and Cons posseses a lattice structure w.r.t their corresponding meet and union operators.

Chapter 10

Conclusion

10.1 Basic Conclusions

This thesis is an attempt to formalize the computational analysis of a bio-chemical model. This formalization on the one hand intends to capture the essence of the algorithmic aspects involved in the model, and on the other hand pursues strenghting the relation between the research in algorithmic and mathematical aspects of the disciplines related to Algebraic chemistries.

In the first part of this thesis it is presented the Algebraic chemistry formalism and it is summarized the current knowledge on computational aspects of the theory, complemented with several examples to simplify its understanding. In the second part it is discretized the Algebraic chemistry to capture the algorithmic escence of organisation computation (see Definition 34 and Corollary 4). It was reformulated, in concurrent processing formalisms (Petri Nets and Vector Addition Systems), this discrete version of Algebraic chemistry (see Theorem 6). We study the relation of some well known properties of Petri Nets with respect to the Algebraic chemistries formalism (see Chapter 5). Furthermore, this discrete framing was helpful to develop a structural analysis of organisations, central part of Algebraic chemistry theory, by using concepts and techniques established for those concurrent formalisms: It was developed and analised, in a discrete schema, an algorithm to verify self-maintainance and thus the property of been an organisation (see Algorithm 6 and Theorem 8). Hence the insights arosed from the development of the mentioned algorithm made possible a more depth analysis of the algebraic chemistries in which it is revealed the underlying roles of molecules when a set of molecules is considered as a subnetwork of the whole Algebraic chemistry (see Theorem 14). The role which play the molecules in a subnetwork defines the computational time required to verify if the subnetwork forms an organisation (see Lemma 24 and Lemma 26), and those results combined with Theorem 14 permit a better understanding of the organisation-verification problem and thus a simplification of the computation of the organisational structure (see Theorem 26). Furthermore, the result of this thesis makes possible the establishment of a quite tight separation of different classes of Algebraic chemistries, in terms of the types (roles) of molecules their organisational structure has, and thus (by

Theorem 26) in terms of how difficult is compute the organisations (see Chapter 9) in each class. All these results complement previous work in computation of organisations research and provides the starting point (and hopefully the motivation) for the development and implementation of new and more depurated algorithms to deal with Algebraic chemistries.
The future work is left as a different section due to the large amount of aspects which can be extended.

10.2 Future Work

During the developing of this thesis, several topics for future research were found. They all suggest advances in the current knowledge of organisation theory:

- **Complement the result of Chapter 8:** Several analysis can extend the results of this thesis. I suggest for example the following:

  DEFINITION **82 (Directly-causally connected)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC. Let $m, m' \in \mathcal{M}$. We say m is directly-causally connected in $\mathcal{R}$ w.r.t. m' iff there exist a reaction $R = A \to B \in \mathcal{R}$ s.t. m is present in A and m' is present in B.*

  DEFINITION **83 (Causally fired sequence of reactions)** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC. Let $S = R_1 \cdots R_t$ be a finite sequence of reactions, where $R_i \in \mathcal{R}$ for all $i = 1, ..., t$. We say S is a causally-fired sequence of reactions if for each pair $(R_i = A_i \to B_i, R_{i+1} = A_{i+1} \to B_{i+1})$ we have $\exists m_i \in A_i$ which is directly-causally connected to some $m' \in A_{i+1}$.*

  The following is a simple but very important fact about networks.

  LEMMA **40** *Let $\langle \mathcal{M}, \mathcal{R} \rangle$ be an AC. Let $m, m' \in \mathcal{M}$. Then one and only one of the following sentences is true:*

  1. *m and m' are not causally connected.*
  2. *m is causally connected to m' by a finite number of sequences of reactions.*
  3. *m is causally connected to m' and m' is causally connected to m.*

  Causal connection and causally-fired sequences of reactions are interesting concepts in Reaction networks. Both concepts tends to make clear the computational understanding of the consumption-production of molecules during the reaction process.

  Another interesting idea is trying to *bridge the gap* from the semi-self-maintaining property to the self-maintaining property. I conjecture that a semi-self-maintaining set such that each connected part has a unique maximal overproduced set, is self-maintaining.
  Several other ideas of formalizing the production-consumption process involving graph-theoretical concepts can be done to improve the algorithmic results of this work.

- **Extend results to infinite Algebraic chemistries using results from Order theory:** As we saw in this work, there are several mathematical structures which can hold an x-structure (from poset to lattice). The field of Order theory deals with these strcutures. It would be interesting to explore the Order theory from the restricted set universe of AC (Finite sets of elements related by reaction-rules). For example the topological parameters *maximal chain and antichain* are useful to bound the size of the organisational structure by using Dilworth's theorem (a well known result in Order theory). I have commented this idea to the head of the research group in computational systems biology, Marie-France Sagot, in my research visit to University Claude Bernard I (Lyon, France). She coincided with me in the interesting potential of this idea, and she showed enthusiasm in hear more about these kind of results in case I do research more on it.

- **Ecological study of networks:** When we combine sets of molecules, properties such as closure or self-maintainance can be lost, or gained. For example if one of the sets is an organisation, but after been joined to another set (eventually an organisation too) we lost the property of been an organisation, we can understand this process from the point of view of stability that both organisations *compete for living.* In fact there are organisations which are non-compatible. For example, let

  $$\mathcal{M} = \{s_1, s_2\} \text{ and } \mathcal{R} = \{a + b \to \emptyset\}.$$

  Clearly, $\{a\}$ and $\{b\}$ are organisations, but they together do not form an organisation. With Stephan Peter and Peter Dittrich, we explored this ideas in [42] by using the P-systems [39] as the reaction network paradigm. We found that the non-compatibility is an extensive phenomena in reaction networks. We also suggested that all the ecological interactions (depredation, cooperation, parasitism, etc) arise naturally when it is pursued the understanding of the co-existence of organisations in a reaction network.

  The concept of ecological interactions [54] provides another interesting point of view, which is the **ecological interactions of reactions**. The idea is to imagine the reactions as entities which for existing need to consume molecules, and as result they produce molecules (reactions eat molecules and produces molecules which in most cases are non-eatable). The coexistence of such entites (reactions) lead immediatly to the definition of what is an organisation from an ecological (even economical) point of view, because the resources are the molecules, and each entity (reaction understood as ecological specie) needs to consume and produce cooperatively their resources for the survival of the eco-system.

  I propose that these ideas (or methaphores) can be extensively studied as a source of insights in Algebraic chemistry theory.

- **Extend to infinite:** A natural extension of this work would be the algorithmic analisys of infinite ACs. The work of infinite networks can be captured in a finite schema by using *rule based systems* which consist basically in thinking molecules as words and reactions as language rules, thus any possible molecule (word) can be built from the basic molecules of the system (alphabet) by using the reaction rules (language rules). It has to be understood first how to extend the formalism (definition of self-maintainance, closure, etc) to deal with this perspective, and try to establish a linking with the dynamics of the system.

- **Include space:** The dynamical system which describes the evolution of an AC does not consider in this model the positions of the molecules. There are preliminary works done by the Jena Bio-systems group in which it is studied the spatial scaling required to see an organisation. They have simulations of simple systems constrained to evolve in a closed space (square) and they look at the average concentration of molecules at different scales by partitioning the space in squares, changing the size of the partitioning square, they change the resolution of what is being saw. It is shown that as bigger is the scale (square) bigger will be the organisations that are seen (see [50]). Another interesting approach is the paradigm of Membrane computing [39] (P-systems) in which the molecules are separated by compartments. In [42] we use the approach of P-systems to study how the interaction between different compartments lead to the reinforcement or destruction of organisations.

Bibliography

[1] Aceto L., Fokkink W., Ingólfsdóttir A.: On a question of A. Salomaa: the equational theory of regular expression, Theoretical Computer Science **v.209(1-2)**, (1998), 163-178.

[2] Acuña V., Chierichettia F., Lacroix V., Marchetti-Spaccamelaa A., Marie-France Sagot M-F. and Stougie L.: Modes and cuts in metabolic networks: Complexity and algorithms, Biosystems **v.95(1)**, (2009), 51-60.

[3] Angeli D., De Leenheer P., Sontag E.: A Petri Net Approach to Persistence Analysis in Chemical Reaction Networks, Biology and Control Theory: Current Challenges, ISBN 978-3-540-71987-8, (2007), 181-216.

[4] Angeli D.: On modularity and persistence of chemical reaction networks, Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, Print ISBN: 978-1-4244-3123-6, (2008), 2650-2655.

[5] Best E., Esparza J.: Model checking of persistent Petri nets, Computer Science Logic, ISBN 978-3-540-55789-0, (1992), 35-52.

[6] Benkö G., Centler F., Dittrich P., Flamm C., Brbel M. R., Stadler P., Stadler F.: A Topological Approach to Chemical Organizations., Artificial Life, **v.15**, No. 1, (2009), 71-88.

[7] Borosh I. and Trevig L.B.: Bounds on positive integral solutions of linear diophantine equtions. Proc. AMS **v.55**, n.2, March 1976.

[8] Borosh I. and Trevig L.B: Small solutions of positive linear diophantine equtions. Discrete Mathematics North-Holland **v.58**, (1986), 215-220.

[9] Cartensen B. and Valx R.: Infinite behaviour and fairness in Petri Nets, Advances in Petri Nets, ISBN 978-3-540-15204-0, (1984), 83-100.

[10] Centler F., Kaleta C., Speroni O. and Dittrich P.: Computing chemical organizations in biological networks. Systems biology **v.24(14)**, (2008), 1611-1618.

[11] Centler F., Speroni P., Matsumaru N. and Dittrich P., Chemical Organizations in the Central Sugar Metabolism of Escherichia coli, Mathematical Modeling of Biological Systems, (2006), Modeling and Simulation in Science Engineering and Technology, Postproceedings of ECMTB 2005.

[12] Crazzolara F. and Winskel G.: Petri Nets With Persistence, Electronic Notes in Theoretical Computer Science (ENTCS), **v.121**, (2005), 143-155.

[13] Crespi-Reghizzi S., Mandrioli D., Commutative Grammars, Springer Milan, Calcolo **13(2)**, (1976), 173-189.

[14] Dittrich P., Speroni di Fenizio P.. Chemical Organisation theory. Bulletin of Mathematical Biology **v.69**, (2007), 1199-1231.

[15] Esparza J.: Petri Nets, Commutative Context-Free Grammars and Basic Parallel Processes, Fundamenta Informaticae **30**, (1997), 23-41.

[16] Esparza J. and Nielsen, M.:Decidability Issues for Petri Nets- a Survey, Journal of Information Processing and Cybernetics **v.30(3)**, (1994), 143-160.

[17] Gagneur J. and Klamt S.: Computation of elementary modes: a unifying framework an the new binary approach. BMC Bioinformatics, **v.5**, (2004), 1-21.

[18] Guldberg C. M. and Waage P.: Studies Concerning Affinity C. M. Forhandlinger: Videnskabs-Selskabet i Christiana (1864), 35.

[19] Huynh D.: The complexity of equivalence problems for commutative grammars, Information and Control **v.57(1)**, (1983), 21-39.

[20] Klamt S., Stelling J.: Combinatorial complexity of pathway analysis in metabolic networks. Molecular Biology Reports - Springer, (2002), 233-236.

[21] S. Rao Kosaraju: Decidability of reachability in vector addition systems Proceedings of the fourteenth annual ACM symposium on Theory of computing, (1982), 267-281.

[22] Hack M.: The equality problem for vector addition systems is undecidable. Theoretical Computer Science **v.2**, (1976), 77-95.

[23] Heiner M., Gilbert D. and Donaldson R.: Petri Nets for Systems and Synthetic Biology. Formal Methods for Computational Systems Biology - Springer, (2008), 215-264.

[24] Hopcroft J., Pansiot J.-J.: On the reachability problem for 5-dimensional vector addition systems, Theoret. Comput. Sci. **v.8(2)**, (1979), 135-159.

[25] Horn F. and Jackson R.: General mass action kinetics, Archive for Rational Mechanics and Analysis, **v.47(2)**, (1972), 81-116

[26] Howell R., Rosier L., Yen Hsu-Chun: A taxonomy of fairness and temporal logic problems for Petri nets, Mathematical Foundations of Computer Science, ISBN 978-3-540-50110-7, (1988), 351-359.

[27] Ibarra O., Dang Z., Egecioglu O.: Catalytic P systems, semilinear sets, and vector addition systems,Theoretical Computer Science **v.312(2-3)**, (2004), 379-399.

[28] Johnson C. and Newmann M.: A surprising Determinantal inequality for real matrices. Math Ann., **v.247**, 1980, 179-186.

[29] Karp R. M., Parallel Program Schemata, Journal of Computer and System Sciences **v.3**, (1969), 147-195.

[30] Lacroix V., Cottret L., Thébault P, and Sagot M-F.: An Introduction to Metabolic Networks and Their Structural Analysis, Transactions on computational biology and bioinformatics, **v. 5**, (2004), 594-617.

[31] Landweber L. H., Robertson E. L.: Properties of Conflict-Free and Persistent Petri Nets, Journal of the ACM (JACM) **v.25(3)**, (1978).

[32] Matsamaru N., Centler F., Speroni di fenizio P., Dittrich P.: Chemical Organization Theory as a Theoretical Base for Chemical Computing, Proceedings of the 2005 Workshop on Unconventional Computing: From Cellular Automata to Wetware, (2005), 75-88. Information and Control **v.66(1-2)**, (1985), 103-121.

[33] Matsamaru N., Centler F., Speroni P. P. Dittrich P.: Chemical Organization Theory as a Theoretical Base for Chemical Computing. International Journal of Unconventional Computing **v.3(4)**, (2006), 285-309.

[34] Matsumaru N., Centler F., Speroni P., Dittrich P.: Chemical organization theory applied to virus dynamics, it - Information Technology, **48(3)**, 154-160, (2006).

[35] Matsumaru N., Centler F., Speroni P., Dittrich P.: A Case Study of Chemical Organization Theory Applied to Virus Dynamics , J. T. Kim (Ed.), Systems Biology Workshop at ECAL 2005, Workshop Proceedings CD-ROM, 5-9 Septmebner 2005, Kent, UK .

[36] Murata. T.: Petri nets and its applications. Proceedings of the IEEE, **v.77 (4)**, (1989), 541-580.

[37] Sacerdote, G. S. and Tenney, R. L.: The decidability of the reachability problem for vector addition systems (Preliminary Version), Proceedings of the ninth annual ACM symposium on Theory of computing, (1977), 61-67.

[38] Rackoff C.: The Covering and Boundedness problems for vector addition systems. Theoretical Computer Science, **v.6**, (1978) 223-231.

[39] Paun G., Rozenberg G.: A guide to membrane computing, Theoretical computer science, **v.287**, (2002), 73-100.

[40] Petri C. A., 1962. Kommunikation mit automaten. Ph.D. thesis, University of Bonn, Bonn.

[41] Peter S., Dittrich P.: On the relation between organizations and limit sets in chemical reaction systems, Advances in Complex Systems (submitted).

[42] Peter S., Veloz T., Dittrich P.: Feasibility of Organizations -A Refinement of Chemical Organization Theory, Eleventh International Conference on Membrane Computing, Accepted paper (In press).

[43] Reddy V.N, Mavrovouniotis M. L. and Liebmant M. N.: Petri Net Representations in Metabolic Pathways, ISMB-93 Proceedings, (1993), 328-336.

[44] Reddy V.N, Liebmant M. N. and Mavrovouniotis M. L., Qualitative analysis of biochemical reaction systems,Comput Biol. Med., **26(1)**, (1996), 9-24.

[45] Rosier L. and Yen H.: A multiparamater analysis of the boundedness problem for vector addition system. Journal of Computer and System Sciences **v.32**, (1986), 105-135.

[46] Ruths D., Muller M., Jen-Te T., Nakhleh L., Ram P.: The Signaling Petri Net-Based Simulator: A Non- Parametric Strategy for Characterizing the Dynamics of Cell-Specific Signaling Networks. PLoS Computational Biology, **v.4**, (2008), 1-15,.

[47] Schuster S., Dandekar T., Fell D. A.: Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. Trends Biotechnol., **17**, (1999), 5360.

[48] Skoog F., Miller C.O.: Chemical regularion of growth and organ formation in plant fissue cultured. In vitro. Symp. Soc. Exp. Biol., **11**, (1957), 118-131,.

[49] Speroni P.: Chemical Organizations Theory (Doctoral thesis), Chair of Systems Analysis, Department of Computer Science, University of Jena, Germany, (2007).

[50] Speroni P., Dittrich P.: Chemical Organizations at Different Spatial Scales, Proc. European Conference on Artificial Life (ECAL 2007), LNCS 4648, (2007), p. 1-11.

[51] Sporn M. and Roberts A.: Autocrine growth factors and cancer, Nature, **v.313**, (1985), 745 - 747.

[52] Stanley R. P.: Enumerative Combinatorics, vol.1, Cambridge Studies in Advanced Mathematics 49, Cambridge University Press, ISBN 0-521-66351-2

[53] Princeton University, Linear Programming: Foundations and Extensions, http://www.princeton.edu/ rvdb/LPbook/.

[54] *http://en.wikipedia.org/wiki/Biological_interaction*