
Introduction

1.1. General introduction

The main motivation of this book is the modeling of a class of dynamic systems usually called “discrete event systems” where the timing of the events is crucial. Events are viewed as sudden changes in the process, which is essentially a man-made system, such as automated manufacturing lines or transportation systems. Forming the core of the following sections, the synchronization of the tasks is expressed by the maximum of the ends of the tasks conditioning the beginning of another task. A classical example is the departure time of a train that should wait for the arrival of other trains in order to allow for the changeover of passengers. The basic idea of dioid algebra is to treat the maximum as an operation that is included in algebra, here the $(\max, +)$ algebra. A large part of this book concerns this topic. Another fundamental idea is to treat the maximum in a maximization of a criterion as in linear programming. We also focus on this second topic in this book.

1.2. History and three mainstays

This book is based on the three following mainstays:

- Well developed in the European and French scientific community, the topic of dioid algebra as $(\max, +)$ algebra has existed for the past three decades. Its main advantage is its *formalism* that allows us to clearly describe complex notions and the possibility of transposing theoretical results between dioids and practical applications. Only a few books have been written on this subject. The first was written by R.A. Cuninghame-Green in 1979. We can quote in graph theory the important books of Gondran and Minoux (Chapter 3 in [GON 84] and [GON 02]). At the origin of the “French school”, another book [BAC 92] was written from the system-theoretical point of view in 1992. Finally, the last book focuses on the transportation systems and was produced in 2006 by a team working in the Netherlands [HEI 06].

- In conventional algebra, linear programming is the oldest for which we can quote the famous French mathematician and physicist Joseph Fourier (1768–1830) for his work at the beginning of the 19th Century (Fourier–Motzkin algorithm). However, this topic is still current and the IT community has produced many algorithms that have efficiently solved bivariable linear systems for the past three decades.

- The third mainstay is the Petri net, which has its origin in Carl Adam Petri’s dissertation submitted in 1962 in West Germany (Carl Adam Petri 1926–2010). A Petri net is a graphical modeling tool that describes complex discrete event systems with various themes and application fields. This topic has been well developed for the past three decades in Europe.

1.3. Scientific context

We describe below the main concepts that will determine the following studies in the mathematical field and Petri

nets. The modeling of the time behaviors will also play an important role.

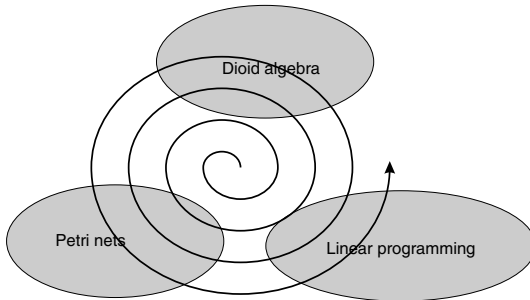


Figure 1.1. *Three topics*

1.3.1. *Dioids*

In the mathematical field, there are different structures with sets provided with one or two operations [GON 84]. We principally consider in this book the structure of the dioid, which is an idempotent semi-ring, and, in particular, the $(\max, +)$ algebra $(\mathbb{R} \cup \{-\infty\}, \oplus, \otimes)$, which is an example of a dioid. This is an interesting tool allowing us to study graph theory and to analyze discrete event systems. The structure of the ring is well-known as familiar examples are integers and real numbers, equipped with the usual addition and multiplication $(\mathbb{Z}, +, \times)$ and $(\mathbb{R}, +, \times)$, respectively), which are used in many fields such as automatic control. The common point is the structure of the semi-ring, which is the mainstay of the structure of the dioid and the ring (see Figure 1.2). The fundamental difference is the existence of symmetry for the first operation $(a + (-a) = (-a) + a = 0)$ of the rings that is replaced by idempotence $(a + a = a)$ for the first operation (maximum for the $(\max, +)$ algebra) of the dioids. This difference is crucial as it is proved that the set is reduced to the neutral element of the first operation $(-\infty$ for the operation $\max$ in the $(\max, +)$ algebra) if we assume that the structure

presents the two properties. This point shows the fundamental difference between automatic control for continuous systems and discrete event systems. The analogies between these two fields are limited to the common core, which is the structure of the semi-ring, even if the authors working in discrete event systems can find some general principles in automatic control allowing the resolution of some specific problems such as “estimation” and “control” in discrete event systems.

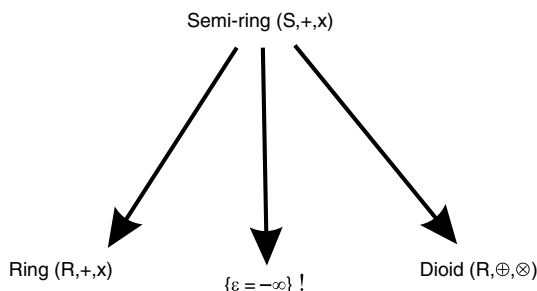


Figure 1.2. *Two algebras: common core and fundamental difference*

1.3.2. *Petri nets*

In the field of discrete event systems, the Petri net allows the modeling of many types of industrial process, transportation system, electronics system, etc. A classical classification of Petri nets can be made with respect to the following two fundamental types (T. Murata gives a good overview of this subject in [MUR 89]).

- The event graphs (each place has exactly one in-going transition and one out-going transition) highlight the synchronization, but no choice is possible. The introduction of time is possible in this model and complex synchronizations can be considered. We will choose this model in the following chapters.

– The state graphs (each transition has exactly one in-going place and one out-going place) consider the choices but do not model the synchronization. The logical aspect is an important characteristic in this model, but the introduction of time is not easy.

– The intersection of the two classes is not empty but only contains Petri nets that do not express either the synchronization or the choice. Therefore, event graphs and state graphs are two fundamental models that equally determine two research fields.

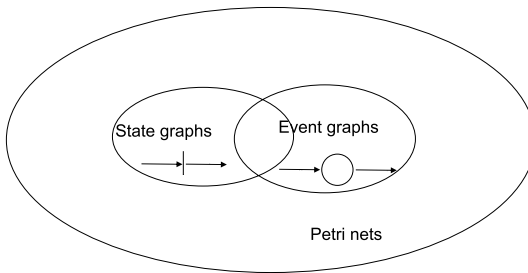


Figure 1.3. *Key structures in Petri nets*

1.3.3. *Time and algebraic models*

All processes naturally present a time evolution and the introduction of time in Petri nets constitutes an important motivation. There are also many everyday examples such as the arrival of students to a course (see Chapter 4 on control). However, the direct introduction of time in general Petri nets quickly leads to a combinatorial blow up of the state-space, which reduces the application field. Contrary to Model Checking, which generates an origin at each event (firing transition) in complex systems, we consider below a unique origin time. Moreover, we will consider specific time models where we can establish a complete algebraic model.

In the field of Petri nets, P-time event graphs, T-time event graphs and time stream event graphs allow us to describe complex synchronizations and to model processes in the production industry, microcircuit design, transportation systems, the food industry or multimedia under the time aspect. The state trajectory is not determined by the iteration of a state equation but is described by a space evolution expressed by the iteration of an interval. The bounds are defined by functions using the minimum, maximum and addition operations. These models are called “interval models”. The models obtained are not only $(\max, +)$ models but also $(\min, \max, +)$ models.

Note that the choice of analysis of these models comes from the choice of the use of the “dater”, that is, to describe each event by a numbered date or to note by a “counter”, i.e. the number of events at each moment. This choice affects the initial modeling, which is a delicate step as it can lead to an unusable model. Note that it is always possible to numerically convert a trajectory expressed with daters into counters, and vice versa. A consequence is that the following chapters will describe *complex* phenomena of *synchronization* for Petri nets presenting *simple structures*. The dual remark is symmetrical: the choice of the counters leads to the analysis of Petri nets presenting *complex structures, weights on the arcs but simple synchronizations*.

From a mathematical point of view, we consider the real space and use the following algebras, that is $(\mathbb{R} \cup \{-\infty\}, \max, +)$, and $(\mathbb{R} \cup \{-\infty\}, \min, \max, +)$, possibly completed with $+\infty$. On the other hand, the counters are defined over the integers and not the real numbers. In that case, we consider algebra as $(\mathbb{Z} \cup \{+\infty\}, \min, +)$ or $(\mathbb{N} \cup \{+\infty\}, \min, +)$ possibly completed with $-\infty$. Remember that the integer linear programming problems need specific techniques.

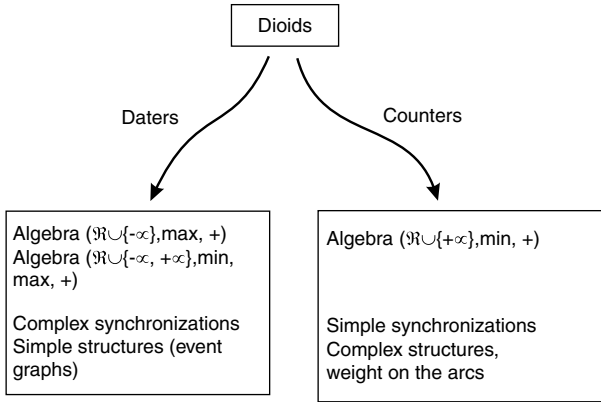


Figure 1.4. *Daters and counters*

1.4. Organization of the book

Figure 1.5 presents the chapters of the book focusing on time models based on daters. We consider the liveness problem as a bad modeling or incorrect parameters which can lead to the absence of a time evolution. The cycle time is the subject of Chapter 3 as it gives an interesting picture of the efficiency of the system. The aim is also to control the process and this point is described in the final chapters.

A comparison of the different classes of time event graphs will be presented at the beginning of chapters on consistency and control as our main interest is on the algebraic models that can describe a broad spectrum of Petri nets. In this book, we also focus on the efficiency of the proposed algorithms: different discussions about complexity and CPU time will be given in the chapters on liveness and predictive control. We also describe some pedagogical examples that will illustrate the main concepts and suggest some new ideas. Each chapter is almost autonomous and the reader only needs to read

the preliminary remarks of the chapter on consistency. The book ends with a bibliography and a list of frequently used symbols.

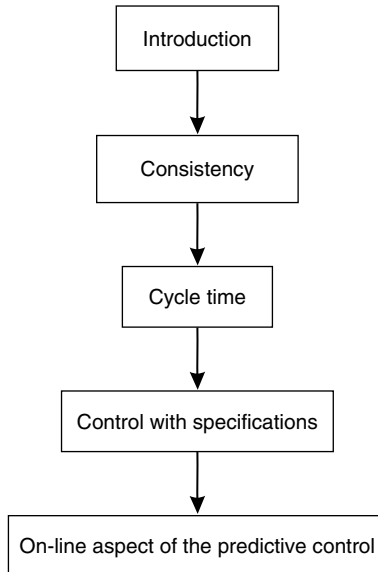


Figure 1.5. *Chapters of the book*