Université catholique de Louvain
Faculté des sciences appliquées
Département d'ingénierie mathématique

Center for Systems Engineering and Applied Mechanics

Dynamics, Information and Computation

Jean-Charles Delvenne

Thesis submitted in partial fulfillment
of the requirements for the degree of
Docteur en sciences appliquées

Dissertation committee:
-Prof. Jean-Didier Legat, Dean of the Faculté des Sciences Appliquées
(President)
-Prof. Vincent Blondel (Advisor)
-Prof. Jean Bricmont
-Prof. Philippe Delsarte
-Prof. Pascal Koiran
-Prof. Petr Kůrka
-Prof. Sandro Zampieri

December 2005

Structure of the thesis

The heart of this thesis is composed of four independent chapters, related by a common flavor of systems theory and computer science.

Chapter 3: Almost periodic configurations and undecidable dynamics. We describe Turing machines, tilings and infinite words as dynamical systems and analyze some of their dynamical properties. It is known that some of these systems do not always have periodic configurations; we prove that they always have almost periodic configurations and we quantify almost periodicity. We then study the decidability of dynamical properties for these systems. In analogy to Rice's theorem for computable functions, we derive a theorem that characterizes dynamical system properties that are undecidable. As an illustration of this result, we prove that topological entropy is undecidable for Turing machines and for tilings.

Chapter 4: Decidability and universality in symbolic discrete-time dynamical systems. Many different definitions of computational universality for various types of dynamical systems have flourished since Turing's work. We propose a general definition of universality that applies to arbitrary discrete time symbolic dynamical systems. Universality of a system is defined as undecidability of a model-checking problem. For Turing machines, counter machines and tag systems, our definition coincides with the classical definition. It yields, however, a new definition for cellular automata and subshifts. Our definition is robust with respect to perturbations on the initial condition, which is a desirable feature for physical realizability.
We derive necessary conditions for undecidability and universality. For instance, a universal system must have a sensitive point and a proper subsystem. We conjecture that universal systems have an infinite number of subsystems. We also discuss the idea according to which computation should occur at the 'edge of chaos' and we exhibit a universal chaotic system.

Chapter 5: Complexity of control on finite automata. We consider control questions for input-output automata. In particular, we find estimations of the minimal number of states of an automaton able to control a given automaton. We prove that, on average, feedback closed-loop control automata do not have fewer states than open-loop control automata when the control objective is to steer the controlled automata to a target state.

Chapter 6: An optimal quantized feedback strategy for scalar linear systems. We give an optimal (memoryless) quantized feedback strategy for stabilization of scalar linear systems, in the case of integral slope. As we do not require the quantization subsets to be intervals, this strategy has better performances than allowed by the lower bounds recently proved by Fagnani and Zampieri. We also describe a general setting, in which we prove a necessary and sufficient condition for the existence of a memoryless quantized feedback to achieve stability, and provide an analysis of Maxwell's demon in this context.

The four main chapters are preceded by an introduction and prerequisites, and are followed by conclusions.

The introductory chapter describes the contributions in an informal and intuitive manner, along with a summary of the background. Chapter 2 contains the basic material needed in the subsequent chapters, or gives references to this material. Short conclusions are drawn in Chapter 7.

All the results appearing in this thesis and for which no source or reference is indicated have been published or accepted for publication. Section 3.2 and a part of Chapter 5 have grown from my unpublished master thesis, supervised by Prof. Vincent D. Blondel. The results of Chapter 3 have been published as a journal paper [DB04b]:

J.-Ch. D. and V. D. Blondel. Quasiperiodic configurations and undecidable dynamics for tilings, infinite words and Turing machines. *Theoretical Computer Science*, 319:127–143, 2004.

The results of Chapter 4 have been obtained in collaboration with Prof. Petr Kůrka and Prof. Vincent D. Blondel and are accepted as a journal article [DKB05b]:

J.-Ch. D., P. Kůrka and V. D. Blondel. Computational universality in symbolic dynamical systems. *Fundamenta Informaticae*. Accepted in 2005.

They are also published in proceedings [DKB05a] under a short form:

J.-Ch. D., P. Kůrka and V. D. Blondel. Computational universality in symbolic dynamical systems. In M. Margenstern, editor, *MCU 2004*, Lecture Notes

in Computer Science 3354, 104–115, Springer-Verlag 2005.

The results of Chapter 5 have been published in proceedings [DB04a]:

J.-Ch. D. and V. D. Blondel. Complexity of control on finite automata. In B. De Moor, B. Mortmans, J. Willems, P. Van Dooren, and V. D. Blondel, editors, *Proceedings of the Symposium on Mathematical Theory of Networks and Systems (MTNS 2004)*, Leuven, 2004.

They are also to appear as a journal article [DB06]:

J.-Ch. D. and V. D. Blondel. Complexity of control on finite automata. *IEEE Transactions on Automatic Control*. Accepted in 2005.

Chapter 6 was accepted as a journal article [Del06]:

J.-Ch. D. An optimal quantized feedback strategy for scalar linear systems. *IEEE Transactions on Automatic Control*. Accepted in 2005.

All these results have also benefited from discussions with Jarkko Kari (who pointed out Example 1), Eugene Asarin (who suggested Proposition 15), Anahi Gajardo, Sandro Zampieri, Alexander Megretski, Petr Kůrka, Vincent D. Blondel, and many more.

The drafts of the above mentioned articles have amply served to constitute a preliminary version of the present text.

This thesis does not include all the work performed during the last years. I have also collaborated to the following journal paper [HADB05]:

J. M. Hendrickx, B. D. O. Anderson, J.-Ch. Delvenne and V. D. Blondel. Directed graphs for the analysis of rigidity and persistence in autonomous agent systems. *International Journal of Robust and Non-Linear Control*. Accepted in 2005.

Acknowledgements

A Ph.D. thesis is a long-standing task, and although considered as an individual achievement, it cannot be completed without the help of many people.

First and foremost, I wish to thank my advisor and mentor Vincent Blondel. Vincent's support and advice have played a central role in both mathematical and nonmathematical aspects of my research. His thorough knowledge of the Republic of Science has been of invaluable help, and by his constantly renewed effort to teach me how to write clear articles in correct English, Vincent would qualify for sanctification.

I am thankful to Philippe Delsarte and Petr Kůrka for taking part in my steering committee, and to Jean Bricmont, Pascal Koiran, and Sandro Zampieri for accepting to be members of the jury. I also thank the Dean of the Faculty Jean-Didier Legat for assuming the charge of president of the jury.

The course of my research has been influenced by many scientific encounters, in Louvain-la-Neuve or abroad, from which fruitful discussions and constructive comments arose. Particularly, my gratitude goes to Eugene Asarin, Jean Bricmont, Pieter Collins, Anahi Gajardo, Jarkko Kari, Petr Kůrka (with whom the content of Chapter 4 was developed), Iven Mareels, Alexander Megretski, Cris Moore, Gunnar Tufte and his colleagues of NTNU (Trondheim), and Sandro Zampieri. Petr welcomed me in Prague several times, and Sandro offered me the opportunity to spend some time in Padua. I will never forget their hospitality during these delightful stays.

The secretary staff, that is, Isabelle, Michèle, Lydia and Dominique, deserves a special mention for outstanding administrative work, especially when faced to negligent and distracted researchers such as me.

Throughout the completion of my thesis, I have benefited from a supportive environment, thanks to my family, friends and colleagues at Cesame. Above all, I want to thank Anne-Claire, who understands my passion for mathematics while saving me from complete addiction.

Pecunia nervus rerum. This thesis presents research results of the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office. It has been also supported by the ARC (Concerted Research Action) "Large Graphs and Networks", of the French Community of Belgium. The scientific responsibility rests with its authors. I have been holding a FNRS fellowship (Belgian Fund for Scientific Research).

Contents

1

Introduction

A simple observation is the seed from which this thesis has grown out. On the one hand, a dynamical system is a 'thing that evolves in time'. On the other hand, a computer is a 'thing that performs calculations' and writes down partial results in its memory. So a computer must change its memory at every new operation, and may thus be seen as a dynamical system. Conversely, an arbitrary dynamical system might be perceived as computing a particular function across time, the state of the system at every moment being the memory. This let us hope that the theory of systems and the field of computer science interfere in a fruitful way, allowing to solve problems of systems with methods and tools from computer science or conversely.

Of course, such connections are already actively studied. For instance, cellular automata have been of interest for both their computational and dynamic properties since their discovery in the fifties. However the words 'system' and 'computer' are susceptible of many interpretations. A system may have an input or not, may be finite or infinite, may be endowed with a topology or a probability measure, etc. A computer may be modelled by a Turing machine, a finite automaton, can interact with its environment or not, etc. Such diverse models may intermingle very diversely, leaving room for plenty of interesting questions.

Such a broadly defined theme is of course not to be explored exhaustively, and we can only propose an impressionist picture. As such, little effort has been made to provide an unified framework that would underlie the results presented here. Hence Chapters 3 to 6 can be read independently.

1.1 Almost periodic configurations and undecidable dynamics

Turing machines, tilings and subshifts, common tools of theoretical computer science, may also be seen as dynamical systems and in Chapter 3 we analyze some of their dynamical properties. Several questions and properties of these objects appear at first to be different and are given a unifying presentation in this chapter. We do not provide explicit correspondences from one class to the other, but show instead that several properties for these dynamical systems can be proved in a common abstract context. The dynamical systems that we consider are tilings of the plane, Turing machines and infinite words. We describe them briefly below.

A Wang tile is a unit square with colored borders. Given a finite set of Wang tiles, we consider the set of all tilings of the plane $\mathbb{Z}^2$. In a tiling of the plane every integer grid point of the plane is the center of a Wang tile and adjacent borders of tiles have the same color (see Figure 1.1 for a simple example). Note that no rotation of the tiles is allowed. Once a tiling of the plane is given, other tilings can be obtained by horizontal and vertical shifts of the entire tiling. The dynamical system resulting from a set of Wang tiles is given by the set of all possible tilings of the plane, together with the horizontal and vertical shifts. The problem of determining if a given tile set can tile the plane (the domino problem) was proved undecidable by Berger [Ber66]. Tiles sets that cannot tile the plane and those that can tile the plane in a periodic way are both recursively enumerable, as can be shown (see Section 2.9). From this it follows that some tile sets may tile the plane without being able to do so periodically; a fact that disproves the conjecture initially made by Wang that no such sets of tiles exist [Wan61].

The Turing machine model that we consider in this chapter is a Turing machine model with one or more tapes, infinite on the left and on the right, filled with symbols taken from a finite alphabet, and a head reading one symbol on each tape and able to write a new symbol and shift every tape independently to the left or to the right. This model differs from the perhaps more traditional one in that the machine is multi-tape, there is no 'blank' symbol, and all tapes are *entirely* filled with symbols taken from the alphabet. The head is characterized by an internal state and acts deterministically. Among the possible states is the so-called halting state. An example is shown on Figure 1.2. A more detailed description of Turing machines is given in Chapter 2. Let us fix now some Turing machine and consider the set of configurations of the machine that never reach the halting state (a Turing machine configuration is given by a tape content *and* a state of the head). This is a dynamical system for the global transition function of the machine. The mortality problem for Turing machines is the problem of determining if the machine possesses a configuration that never reaches the halting state, i.e., whether or not the corresponding

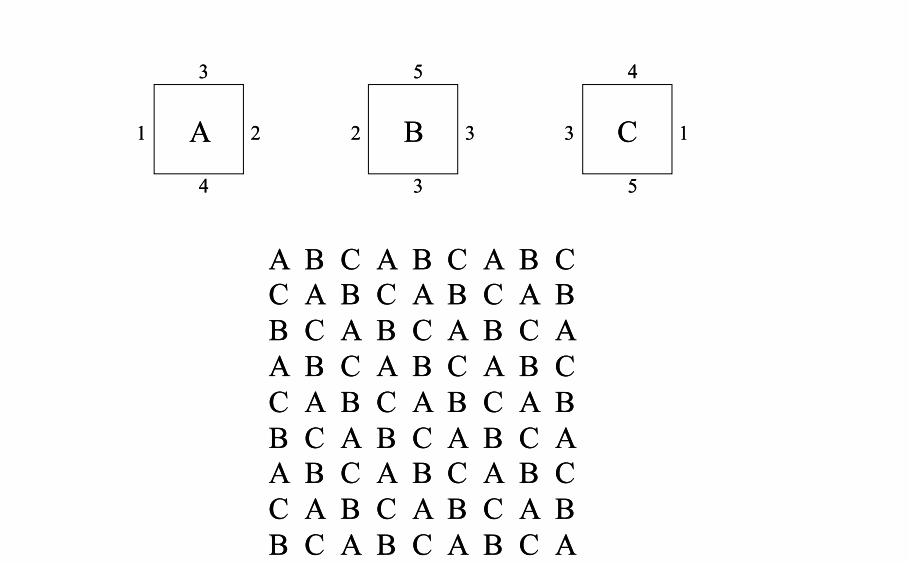

Fig. 1.1. A set of three Wang tiles and a piece of tiling generated by the tile set. Colors are represented by numbers. This is Wang's original example [Wan61].

dynamical system is empty. As explained below, this problem is the Turing machine equivalent to the domino problem and was proved undecidable by Hooper [Hoo66]. As for tiles, it follows from this result that there are Turing machines whose immortal configurations are all non-periodic. Kůrka [Kůr97b] has conjectured that no such Turing machine exists that has no halting state. This conjecture is disproved by Blondel, Cassaigne and Nichitiu [BCN02] where a never-halting Turing machine with six states and four letters and no periodic configurations is constructed. For tilings, it is known that there is a set of 13 tiles with five colors that tile the plane and can only do so in a nonperiodic way [ČI96]. The decidability of the problem of determining if a Turing machine with no halting state possesses a periodic configuration is yet unsettled but is conjectured to be undecidable in [BCN02].

Our final example is that of infinite words. A (right) infinite word on a finite alphabet A is a function $w : \mathbb{N} \to A$. The shift is the map σ defined by $\sigma(a_0a_1a_2a_3\ldots) = a_1a_2a_3\ldots$. Any set of infinite words that is closed in the product topology—explained in more details in Chapter 2—and that is stable under the shift constitutes a dynamical system called a *subshift* . In particular,

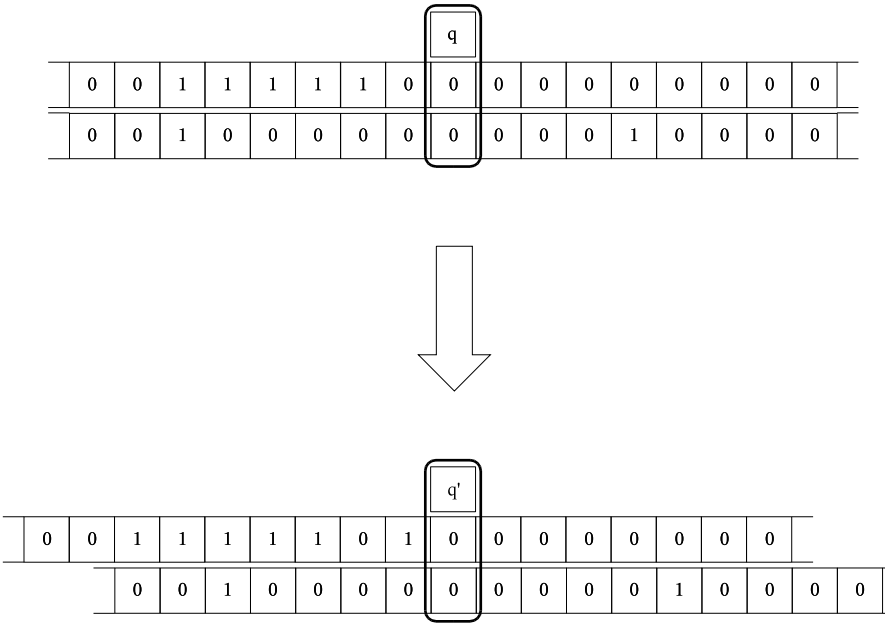

Fig. 1.2. A two-tape Turing machine. Here the head is jumping from state q to state q', writing a 1 in the cell of the upper tape, writing a 0 in the cell of the lower tape, then shifting the upper tape to the left and the lower tape to the right.

the set of infinite words that do not contain as subwords words taken from a particular finite set is called a subshift of *finite type*.

Thus we have three natural examples of dynamical systems. All three systems are constructed in the following way: In the set of all possible configurations, we identify a subset of 'bad' configurations Z. This subset Z is characterized by a finite number of constraints on symbols present in a finite number of places of the configurations. For Turing machines, Z is the set of configurations whose state is the halting state, for a tile set T, Z it is the set of sequences $\mathbb{Z}^2 \to T$ having a tiling error at the origin, and for a subshift, Z is the set of infinite words having a forbidden subword as a prefix. Then we consider the set of all configurations that never reach Z under the system transformations. The resulting sets are respectively the set of immortal configurations of a Turing

machine, the set of tilings or the subshift of finite type. In this context Hooper's problem ('Is there an immortal configuration?') and Berger's problem ('Is there a tiling of the plane?') are the same questions. The corresponding question for infinite words is: 'Is a given subshift of finite type empty?'. This last problem is easily shown to be decidable.

Motivated by those observations, we are lead to think that Turing machines, tilings and subshifts may be fruitfully studied in the common framework of topological dynamics. In this chapter, we aim at unifying questions and theorems that are usually stated separately, and at deriving abstract theorems about dynamical systems that directly apply to tilings, Turing machines and infinite words.

In the first part of the chapter, we prove from classical results of topological dynamics that the class of dynamical systems that we consider always have almost periodic configurations. An almost periodic configuration is a configuration whose associated trajectory regularly comes back near its starting point (in particular, periodic configurations are almost periodic). We also prove that dynamical systems that have an almost periodic configuration that is not periodic have uncountably many such configurations. We then define almost-periodicity functions, which are a way to measure the regularity of almost periodic configurations. Abstracting from the case of tilings and subshifts we give a general definition that can be applied to any dynamical system and we prove that global convergence is a property that can be characterized in terms of almost-periodicity function.

In the second part of the chapter, we study the decidability of dynamical properties of dynamical systems. We are interested in this part by the following general question: *What properties of dynamical systems are decidable?* (A more precise meaning to this question is given in Section 3.4.) In the theory of computability the answer to the question: *What properties of computable functions are decidable?* is given by Rice's Theorem: *All nontrivial properties of computable functions are undecidable*, where a property is trivial if it is verified by all computable functions or by none of them. In this chapter we propose a Rice-style theorem for a class of dynamical systems that includes Turing machines and tilings. Here again we undertake a general approach that holds for any 'rich enough' family of dynamical systems. Our result allows us to prove the undecidability of questions related to the number of invariant sets and the topological entropy of tilings and Turing machines.

Let us note here that several results analogous to Rice's theorem are available in the literature for particular families of dynamical systems. Kari [Kar94] proved that every non-trivial property of the limit sets for cellular automata is undecidable. Cervelle and Durand [CD00] showed that it is undecidable to test whether two tile sets generate the same tilings, even when one of the tile sets is fixed and (almost) arbitrary. However, in this last reference 'tilings' are

understood in a way that is different from ours and the proofs given in [CD00] cannot be transposed to our set-up.

1.2 Decidability and universality in symbolic discrete-time dynamical systems

Computability is often defined via universal Turing machines. As in Chapter 3, a Turing machine can be regarded as a discrete-time dynamical system, i.e., a set of configurations together with a transformation acting on this set. A configuration consists of the state of the head and the content of the tape. Computation is done by observing the trajectory of an initial point under iterated transformation.

There is no reason why Turing machines should be the only dynamical systems capable of universal computation. Indeed, such capabilities have been also claimed for artificial neural networks [Sie99, Koi96], piecewise linear maps [KCG94], analytic maps [KM99], cellular automata [Wol02], piecewise constant vector fields [AMP95], billiard balls on particular pool tables [FT82], or a ray of light between a set of mirrors [Moo90]. For all these systems, many particular definitions of universality have been proposed. Most of them mimic the definition of computation for Turing machines: an initial point is chosen, then we observe the trajectory of this point and see whether it reaches some 'halting' set; see for instance [BC96, SF98]. However, many variants of these definitions are possible and lead to different classes of universal discrete-time dynamical systems. In particular, there is no consensus for what it means for a cellular automaton to be universal. Moreover, in the presence of noise many of these systems loose their computational properties [AB01, Gác97, MO98]; see [Moo98a, Moo98b, Orp97] for definitions of analog computation and issues relative to noise and physical realizability.

Another field of investigation is to make a link between the computational properties of a system and its dynamical properties. For instance, Wolfram classifies the cellular automata into four categories, according their dynamical properties revealed by their space-time diagrams, and conjectures that 'universal' cellular automata are in the class four; see [Wol02]. It has also been suggested that a 'complex' system must be on the 'edge of chaos': this means that the dynamical behavior of such a system is neither simple (i.e., a globally attracting fixed point) nor chaotic; see [CY89, Lan90, MHC94]. Other authors nevertheless argue that a universal system may be chaotic; see [Sie99].

The basic questions we would like to address are the following:

- How to define computational universality for discrete-time dynamical systems?
- What are the dynamical properties of a universal discrete-time dynamical system?

A long-term motivation is to answer these questions from the point of view of physics. What physical systems are universal? Is the gravitational N-body problem universal [Moo90]? Is the Navier-Stokes equation universal [Moo91]? However in this chapter we focus on *symbolic* discrete-time dynamical systems that are *effective*, i.e., systems defined on the Cantor set $\{0,1\}^{\mathbb{N}}$ or a subset of it, whose transformations are computable. Some motivating examples of such systems are Turing machines, cellular automata and subshifts.

Turing's machine was originally meant as a model of a computation performed by a human operator using paper and pencil [Tur36]. We adapt Turing's reasoning to the case where the human operator does not compute by himself/herself with a pencil and paper, but relies on a dynamical system to make the computation. The system is said to be computationally universal if the observations made by the human operator on the system allow him/her to solve any problem that could also be solved by a universal Turing machine. We conclude that a system is universal if some property of its trajectories, such as reachability of a halting set, is r.e.-complete. This is an extension of Davis' definition of universal Turing machine [Dav56].

In this contribution, rather than considering point-to-point or point-to-set properties, we consider set-to-set properties. Typically, given an initial set and a halting set, we want to know whether there is at least one configuration in the initial set whose trajectory eventually reaches the halting set. We require the initial and halting sets to be clopen (closed and open) sets of the Cantor state space. Clopen sets can be described in a natural way with a finite number of bits. Considering clopen sets can be justified by supposing that the human operator cannot separate points that are too close to each other, so we can only check whether the initial state of the system is in some clopen set. Hence the set of initial points that can be distinguished is a clopen set.

We model a human operator who has finitely possible mental states by a finite automaton, similarly to Turing's assumption. Therefore, we do not restrict ourselves to the property 'Is there a trajectory going from U to V?' alone, but consider all properties that can be observed by some finite automaton. Checking whether such a property is verified is a 'model-checking problem'. Model-checking problems are usually defined for finite or countable systems; see [JGP99] for instance. Finally, a universal system is a system that has an r.e.-complete model-checking problem.

This definition addresses the two issues raised above. Firstly, it is a general definition directly applicable to any (effective) symbolic system. Secondly, dealing with clopen sets rather than points takes into account some constraints of physical realizability, such as robustness to noise. With this definition in mind, we prove necessary conditions for a symbolic system to be universal. In particular, we show that a universal symbolic system is not minimal, not equicontinuous and does not satisfy the shadowing property. We conjecture that a universal system must have infinitely many subsystems, and we show

that there is a chaotic system that is universal, contradicting the idea that computation can only happen at the 'edge of chaos'.

1.3 Complexity of control on finite automata

In control theory, feedback is used to obtain a desired behavior from a system. A variety of arguments have been proposed to explain why feedback is useful. Most arguments rely on the notions of uncertainty and lack of information: feedback is useful in particular when the system we want to control is not perfectly known, or when the signals between the system and its environment suffer from noise. In Chapter 5, we study another possible interest of feedback, namely, that feedback may result in a reduction of the controller size or complexity.

We assume that the system we wish to control is perfectly known, and that we would like to steer it from a given initial state to a target state in the simplest possible way. Does the availability of feedback lead to smaller, less complex controllers? And how can this gain of complexity be measured?

The following motivating example is adapted from an example appearing in Egerstedt and Brockett [EB03]. Imagine that one must guide a driver from one particular crossroad to another crossroad in a city. One possible sequence of instructions may be: Turn right onto Regent St., Turn left onto Oxford St., Turn right after 3.3 km.

Another possible description would be to enumerate what direction to follow at every crossroad encountered: Straight, Straight, Right, Straight, Straight, Left, Straight, Straight, Straight, Straight, Straight, Right.

What is the difference between these two descriptions of the same path? The first description needs observations on the environment to be effective: street names, milestones. On the other hand, the second description needs more (but simpler) instructions.

We may see a city as a graph whose vertices are crossroads and edges are streets. We start from an initial vertex and at every step we follow the edge whose input label corresponds to the given input, and we read the output label attached to the edge. In the above example, the edge input labels are Straight, Left or Right, while the output label could for instance be the names of the streets. Such a graph is a particular example of an input-output automaton. An input-output automaton is an input-output discrete-time system with a finite state space and finite input and output alphabets. At every time step an input letter is read, the state is changed accordingly, and an output letter is produced. These automata may also be viewed as approximations of continuous state space systems, where space, inputs and outputs are quantized by finitely many values.

In the context of this chapter, we will consider the objective of driving an automaton from an initial to a target state, and the complexity of an automaton will be identified with its number of states. An automaton can be controlled by another automaton, either in open-loop or in feedback. In order to steer an automaton from an initial state to a target state, feedback control automata can in some cases be much less complex than open-loop control automata. In the main result of this chapter, we prove that, even though this may be the case for particular automata, it is not true on average. More precisely, we prove in our main result that the average number of states needed to drive an n-state automaton from one state to another is

- in the order of $\ln n$ for an open-loop control automaton;
- between the order of $\ln n / \ln \ln n$ and the order of $\ln n$ for a feedback control automaton.

Therefore, although it is not proved that open-loop and feedback have the same order of complexity, little room is left for improvement. As a conclusion, measuring the output is of little use on a system with a random structure. However it is easy to find examples with particular structures on which feedback leads to controllers of very low complexity. We suggest that it could also be the case for classes of automata, such as quantized linear systems for example.

Our results are in contrast with those presented in [EB03], where the authors also compare the complexity of open-loop versus feedback controllers, and reach the opposite conclusion. The reason for these apparently contradictory results is that the model used in [EB03] is very different from ours. The results presented in [EB03] apply to a sophisticated variant of automata called free-running feedback automata. Essentially, these are automata that do not read an input letter at every time step, and for which an input letter contains a complete description of the feedback law to be applied to the automaton. Thus the input alphabet may be quite large. The complexity of control is the length of the input word realizing the objective, multiplied by the logarithm of the input alphabet cardinality. Then, for such devices, with hypotheses on the structure of the automaton, a strategy based on a mixture of open-loop and feedback is proved to be less complex than pure open-loop.

The work presented here is also in the spirit of Chapter 6 where systems have a continuous state space, but discrete time, discrete inputs and discrete outputs. The goal is then to study the amount of information needed to control the system. In this chapter we pursue similar goals but with quite different methods.

Before we can control a system, we must sometimes identify it. In 1956, Moore [Moo56] proposed some algorithms to identify a black-box automaton, on which we can test inputs and observe the corresponding outputs. Then Trakhtenbrot, Barzdin and Korshunov developed a probabilistic point of view and showed that automata are much easier to identify on average than in the

worst case. They also designed efficient algorithms able to identify 'most' automata; see [TB73]. We stick to a similar framework, but explore one step beyond: supposing that the automaton is correctly identified, how can we control it in the least complex way?

1.4 An optimal quantized feedback strategy for scalar linear systems

The theory of control with communication constraints, which is at the core of Chapter 6, deals with the following question. We want to control a system but the amount of information that can be transmitted from the output to the input is limited, e.g., because it must transit through a small bandwidth transmission line. This problem has been the subject of intense research since 1990, and has been formalized in several different ways; see, e.g., [BL00, Del90, EM01, FZ03, FZ04a, NEMM04, TM04, TL04]. A sample of these results is given in Chapter 6.

The following setting is described and studied by Fagnani and Zampieri [FZ03, FZ04a, FZ05]. We are given a discrete-time scalar linear system

$$x_{k+1} = ax_k + u_{k+1}, \tag{1.1}$$

where $k \geq 0$ and $|a| > 1$. We suppose that x_0 is in $[-1, 1]$. A *feedback map* is a map

$$u_{k+1} = \Gamma(x_k).$$

Note that such a feedback map is said to be *memoryless* because Γ takes into account only the last value x_k of the state, instead of the whole history $x_0, x_1, \ldots, x_k$. It is said to be *quantized* if Γ is piecewise constant with finitely many discontinuities.

We want to design a memoryless quantized feedback map such that if we plug the feedback map into Equation 1.1, the state of the system eventually reaches $[-1/C, 1/C]$, for some $C > 1$, and stays in this interval for almost every initial condition $x_0 \in [-1, 1]$, with respect to normalized Lebesgue measure ('almost every' means 'all but a subset of measure zero'). In other words, the control must succeed with probability one. The parameter C is called the *contraction rate*. The number of subintervals on which the feedback map Γ is constant, called *quantization intervals*, is denoted N.

For almost every $x_0 \in [-1, 1]$ we can define the *first entrance time* as the smallest integer t for which $x_t, x_{t+1}, x_{t+2}, \ldots$ are in $[-1/C, 1/C]$. The average first entrance time with respect to normalized Lebesgue measure on $[-1, 1]$ is denoted by T.

The fundamental question is the following: given a triple (C, N, T), can we design a quantized feedback strategy with N quantization intervals, achieving a contraction ratio of C in average time T?

Both necessary and sufficient conditions on C, N, T are given in [FZ05]. For instance, for any $r > 0$, there exists some $s > 0$ such that

$$N \leq r \log_2 C \Rightarrow \lceil T \rceil \geq s \log_2 C. \tag{1.2}$$

Several families of feedback maps are also exhibited, and their optimality is proved. For instance, a family with the following parameters is designed:

$$N/|a| = \Theta(\log C), T = \Theta(\log C).$$

The case of higher dimensions is discussed in [FZ04a].

In this chapter, we relax the condition that the quantization subsets should be intervals. In other words, we partition the interval $[-1, 1]$ into N subsets on which the feedback map Γ is constant, and these subsets are not necessarily intervals. This allows us to design a family of feedback maps that do better than allowed by (1.2). In fact we show that when the slope a is integral, we can design a feedback map with parameters C, N, T if and only if $T \log N \gtrsim \log C$.

2

Prerequisites

In this chapter we explain some of the fundamental notions needed for the following chapters. Elements of general topology, measure theory, dynamical systems, computability, language and automata theory are covered, and we describe Turing machines, cellular automata and tilings. Naive set theory and elementary calculus are not reviewed.

Most of this material is classical and may be skipped by a reader familiar with the concepts, except perhaps the terminology and classification of systems (Sections 2.3 and 2.4), and the presentation of Turing machines (beginning of Section 2.7) and tilings (Section 2.9) as dynamical systems.

2.1 Topology

We briefly recall some definitions and facts about general topology. Many textbooks are available on this subject, for instance [Kel55]. The notions below are mainly useful for Chapters 3 and 4.

A *topological space* is a set endowed with a family of subsets closed under arbitrary (finite or infinite) unions, finite intersections and that contains the empty set and the full set. The family is called a *topology* and the subsets are the *open sets*. The *closed sets* are the complements of the opens sets; they are closed under finite unions and arbitrary intersections. A *neighborhood* of a point is a subset that includes an open set to which the point belongs. The *closure* of a set S is the smallest closed set including S. The closure of a set always exists, because it is the intersection of all closed sets in which the set is included. The *interior* of S is the greatest open set contained in S. The interior of a set always exists. The *border* of S, denoted ∂S, is the closure of S minus the interior of S. An *open cover* of a topological space is a family of open sets whose union is X. A topological space is *compact* if from every open cover a finite open cover can be extracted. Equivalently, a space is compact if any family

of closed subsets for which finite intersections are nonempty has a nonempty intersection. A point x is *isolated* if $\{x\}$ is an open set. A *perfect* space has no isolated point. A subset of a topological space is *dense* if its closure is the full space. A topology $\mathcal{T}$ is *coarser* than another topology $\mathcal{T}'$ on the same set, or $\mathcal{T}'$ is *finer* than $\mathcal{T}$, if $\mathcal{T} \subseteq \mathcal{T}'$. The *discrete* topology is such that every subset is open, and is the finest possible topology. A *basis* of the topology is a family of sets that generates the topology by arbitrary unions.

A map between two topological spaces is *continuous* if the preimage by the map of any open set is open. Recall that the *preimage*, or *inverse image*, of a set S by a map f, denoted $f^{-1}(S)$, is the set of points x in the domain of f such that $f(x) \in S$. Equivalently, a map is continuous if the preimage of any closed set is closed. If the topology of the destination set is generated by a basis, the map is continuous if and only if the preimage of any set of the basis is an open set. A continuous bijective map whose inverse map is continuous is a *homeomorphism*, and two spaces between which a homeomorphism exists are said to be *homeomorphic*. Homeomorphic spaces can be seen as 'essentially the same, as topological spaces'.

The *product* of a family of topological spaces is the set given by the cartesian product of all underlying sets of the family, endowed with the coarsest topology that makes the projection maps continuous. A basis for this *product topology* is composed of all finite intersections of preimages of open sets by the projection maps. A subset of a topological space can be endowed with the *subset topology* made of all intersections of open sets with the subset.

A *Hausdorff* space is a topological space in which any two points have disjoint neighborhoods. A *metric* or *distance* on a set X is a map $d : X \times X \to \mathbb{R}$ that satisfies the following axioms:

- symmetry: $d(x, y) = d(y, x)$ for all x, y;
- positive definiteness: $d(x, y) \geq 0$ with equality iff $x = y$, for all x, y;
- triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$.

The *ultrametric inequality* $d(x, z) \leq \max(d(x, y), d(y, z))$ is strictly stronger than the triangle inequality.

The *open ball* of *radius* $\epsilon > 0$ *centered* at x is the subset $B(x, \epsilon) = \{y \,|\, d(x, y) < \epsilon\}$. The *closed ball* $B[x, \epsilon]$ is the subset $\{y \,|\, d(x, y) \leq \epsilon\}$.

A set endowed with a distance is a *metric space*; it is endowed with the topology generated by the basis of all open balls as basis. A metric space is obviously Hausdorff: for two points x, y, the balls $B(x, \epsilon)$ and $B(y, \epsilon)$, for $\epsilon = d(x, y)/2$, are disjoint neighborhoods. We can check that closed balls are indeed closed for the topology.

The metric restricted to the subset of a metric space yields the subset topology. The product of countably many metric spaces $(X_i, d_i)_{i \in \mathbb{N}}$ is again a metric space with the metric

$$d((x_i)_{i\in\mathbb{N}}, (y_i)_{i\in\mathbb{N}}) = \sum_{i\in\mathbb{N}} \frac{1}{2^i} \frac{d_i(x_i, y_i)}{1 + d_i(x_i, y_i)}.$$

The proof is straightforward, although somewhat tedious.

A sequence of points $(x_n)_n$ *converges* to a point x, called a *limit* of the sequence and denoted $\lim_n x_n$, if every neighborhood of x contains all but finitely many members of the sequence. In a Hausdorff space a sequence has at most one limit. A metric space is compact if and only if every sequence in the space has a converging subsequence.

The image by a continuous map of a compact Hausdorff space is compact. A subset of a compact Hausdorff space is compact for the subset topology if and only if it is closed. These facts, although nontrivial, can be proved in an elementary way by unwinding down the definitions. A bijective continuous map between compact Hausdorff spaces is a homeomorphism. Indeed, the preimage of closed subset by the inverse map is the direct image of the subset by the map. And the direct image of a closed (thus compact) subset is compact (thus closed).

Lebesgue number lemma states that for any open cover of a compact metric space, there is a number $\delta > 0$ such that for any point x the ball $B(x, \delta)$ is included in some element of the cover. Such a δ is called a *Lebesgue number* of the cover.

Baire category theorem states that in a compact metric space, a countable union of closed sets of empty interior has an empty interior. Baire category theorem is actually more general than that, applying to so-called 'complete' metric spaces and to so-called 'locally compact' Hausdorff spaces. But the above version is enough for us. As an application, we can prove that a perfect compact metric space is uncountable. Indeed, a point is a closed set of empty interior, and if there were only countably many points, then the full set would have an empty interior, which is a contradiction. For instance, the interval $[0, 1]$ endowed with the usual metric $d(x, y) = |x - y|$ is uncountable.

A proof of Baire category theorem goes like this. Given a countable family $Y_1, Y_2, Y_3, \ldots$ of closed subsets with empty interior of a compact metric space and an arbitrary open set U, take a compact set U_1 with nonempty interior, e.g., a closed ball, included in $U \setminus Y_1$. Now take a compact set U_2 with nonempty interior included in $U_1 \setminus Y_2$, etc. The intersection of all U_i is nonempty, included in U, and does not meet $\bigcup_j Y_j$, hence no open set U is included in $\bigcup_j Y_j$, which has an empty interior.

2.2 Measure theory

Elements of measure, probability, and information theories are summarized in this section. A reference on measure theory is [Hal50], and [Abr63] on information theory. These concepts are useful in Chapters 5 and 6.

Given a set X, a *σ-algebra* on X is a family of subsets of X closed under complementation and countable unions and that contains the empty set. A *measure* μ on X endowed with a σ-algebra is an assignment of a number $\mu(Y) \in \mathbb{R} \cup \{+\infty\}$ to any set Y of the σ-algebra such that $\mu(Y) \geq 0$, $\mu(\emptyset) = 0$ and $\mu(\bigcup_{i\in\mathbb{N}} Y_i) = \sum_{i\in\mathbb{N}} \mu(Y_i)$ for every countable family of pairwise disjoint sets $(Y_i)_{i\in\mathbb{N}}$. A set endowed with a σ-algebra and a measure is called a *measure space*, and the sets of the σ-algebra are called the *measurable sets*.

A *measurable function* between two measure spaces is a map such that the preimage of a measurable set is measurable. It is said to *preserve measure* if the preimage of a measurable set is a measurable set of same measure.

A finite or countable set can be endowed with the *uniform measure*: every set is measurable and measured by its cardinality. Any topological space can be endowed with the σ-algebra of *Borel sets*, which is the smallest σ-algebra that contains the open sets. Such a smallest σ-algebra always exists because the intersection of a family of σ-algebras is a σ-algebra.

The set $\mathbb{R}^n$ is usually endowed with the *Lebesgue measure*. We require the measure of a box $]a_1, b_1[\times\cdots\times]a_n, b_n[$ to be $(b_1 - a_1) \times \cdots \times (b_n - a_n)$. The measure of a Borel set Y is the infimum value of $\sum_{i\in\mathbb{N}} \mu(Y_i)$, where $(Y_i)_i$ is a countable family of boxes such that $Y \subseteq \bigcup_i Y_i$. The measure of a subset of a Borel set of measure zero is zero as well.

The σ-algebra of *Lebesgue-measurable sets* is the smallest σ-algebra containing the Borel sets and subsets of Borel sets with measure zero. It can be proved that the Lebesgue measure is a measure well defined on the Lebesgue-measurable sets.

We now define the *(Lebesgue) integral* of a measurable function $f : X \to \mathbb{R}$, for any measure space X of measure μ. Given a set $Y \subseteq X$, its *characteristic function* takes value 1 on Y and 0 elsewhere. The integral of the characteristic function is defined as $\mu(Y)$; it might be infinite. The integral of a positive linear combination of finitely many characteristic functions is defined as the linear combination of the integrals. If $f \geq 0$, then the integral of f is defined as the supremum of the integral of g, where $g \leq f$ runs over all positive linear combinations of finitely many characteristic functions. If $f = f^+ - f^-$, where $f^+ \geq 0$ and $f^- \geq 0$, and the integral of the measurable functions f^+ and f^- are not both infinite, the integral of f is defined as the difference of the integrals of f^+ and f^-. For instance, the integral of a function defined on $\mathbb{N}$ with the uniform distribution, is simply the sum of the function.

A measure space for which the full set has measure one is said to be a *probability space*, and the measure is a *probability measure* or *probability distribution*. *Measurable sets* are identified *events* and the measure of a set to the *probability* of an event.

Any measure such that the measure of the full set is finite can be normalized to a probability measure. For instance a finite set can be endowed with the *uniform distribution*.

A *random variable* is a measurable function from a probability space to $\mathbb{R}$. The *expected value*, or *average value*, of a random variable is its integral, whenever it exists.

An important result of measure theory is *Lebesgue's dominated convergence theorem*. Given a measure space X, a sequence of measurable functions $(f_n : X \to \mathbb{R})_{n\in\mathbb{N}}$, a function f such that $f(x) = \lim_n f_n(x)$ for all x, a measurable function g such that $|f_n| \leq f$. Then f is measurable and the integral of f_n converges to the integral of f.

Given a finite or countable set X endowed with a probability distribution, the (Shannon) *entropy* in base $b > 1$ of the distribution is the quantity $-\sum_x p_x \log_b p_x$, where p_x is the probability of $\{x\}$ and with the convention $0 \log_b 0 = 0$. It is always nonnegative and it is equal to 0 exactly when the distribution is concentrated on a unique element. For a finite set, it takes its maximal value $\log_2 |X|$ for the uniform distribution.The notation $|X|$ indicates the cardinality of X. The intuitive meaning of the entropy is the average information carried by an event, or the uncertainty about the outcome of picking up an element of X at random.

2.3 Dynamical systems

The mathematical notion of system is originally intended to model physical systems that evolve in time. As Nature can be modelled in many ways, many definitions of systems have been proposed. In this section and the next, we list those needed in subsequent chapters.

A fairly general definition, used in Chapter 3, is the following. A *dynamical system* (X, G) is given by a set X, whose elements are called *points*, *states* or *configurations*, and a monoid G acting on X. By a *monoid* we mean a set of elements endowed with an associative internal law $. : G \times G \to G : (g_1, g_2) \mapsto g_1.g_2$ and an identity element. The monoid *acts* on X if to every $g \in G$ is associated a transformation f_g (the *action* of g) of X such that for any $g_1, g_2 \in G$, we have $f_{g_1.g_2} = f_{g_1} \circ f_{g_2}$. To simplify notations, we often identify an element of the monoid g to its action f_g. The *orbit* or *trajectory* of a configuration x is the set $\{g(x) \mid g \in G\}$. An *invariant* subset of X is a subset Y of X that is invariant for all transformations in G, i.e., $\forall g \in G : \ g(Y) \subseteq Y$.

If G is the monoid $\mathbb{R}^+$ endowed with $+$ and 0, then we have a *continuous-time* dynamical system. Continuous-time dynamical systems do not appear in this thesis. If G is the monoid $\mathbb{N}$ endowed with $+$ and 0, then we have a *discrete-time* dynamical system. Discrete-time dynamical systems appear mainly in Chapter 4. A discrete-time dynamical system is completely described by the set X and the map $f_1 : X \to X$ associated to 1, thus a discrete-time dynamical system can be denoted as (X, f_1) or $f_1 : X \to X$. A discrete-time dynamical system can be thought of as modelling the following situation. At time zero,

the system is in some arbitrary initial state x; then at every step of time the state is transformed via the map f_1; at time step t (for any $t \in \mathbb{N}$) the system is in state $x_t = f_t(x) = f_1^t(x)$ (the power denotes iterated composition). Note that for discrete-time dynamical systems, what we call the orbit of x in the literature and also in this thesis is sometimes the sequence $(f_1^t(x))_{t\in\mathbb{N}}$ rather than the set $\{f_1^t(x) : t \in \mathbb{N}\}$. The distinction is inessential.

Tilings are examples of dynamical systems whose monoid is $\mathbb{Z}^2$ endowed with $+$ and $(0, 0)$; see Section 2.9 and Chapter 3.

In most cases, we put some additional structure on the set X. From now, we will always assume that X is a compact metric space and that every action of an element of the monoid is a continuous map from X to X.

A *subsystem* of the system (X, G) is a closed subset of X invariant under every action of the monoid G. Endowed with the subset topology and the monoid G, whose actions are restricted to the closed subset, the subsystem is itself a system. A *closed orbit* is the closure of an orbit. Closed orbits are obvious examples of subsystems. Indeed, an orbit is invariant for every action of the monoid, and since every action is continuous, the closed orbit is also invariant under every action. Another example of subsystem is the *limit set* $\cap_{g\in G} g(X)$. The limit set is invariant and closed, hence is a subsystem.

Consider the systems (X, G) and (Y, G) and assume that there is a continuous map $h : X \to Y$ such that for all $g \in G$, $f_g \circ h = h \circ f_g$, where f_g is the action of g. If h is surjective then (Y, G) is a *factor* of (X, G). If h is bijective then (Y, G) and (X, G) are *conjugated* and f is a *conjugacy*. Note that a bijective continuous map between compact metric spaces is a homeomorphism (see Section 2.1). A factor of a system can informally be seen as a simplification of the system. Conjugated systems can informally be seen as essentially identical.

Products of dynamical systems will also be considered, however we prefer not to give an overgeneral definition and redefine a well-suited version every time we need it.

A *minimal* dynamical system is a dynamical system that has no proper subsystem, except the empty set. A point whose orbit is finite is said to be *eventually periodic*. If the orbit is finite and minimal, then the point is said to be *periodic*. Minimal dynamical systems can be characterized as follows.

Proposition 1 *A dynamical system (Y, G) is minimal if and only if the closed orbit of any point is Y.*

Proof. If a dynamical system is minimal, every closed orbit (itself a subsystem) must be the full system. Conversely, if a dynamical system is not minimal then there exists a proper nonempty subsystem; any configuration of this subsystem generates an orbit, the closure of which does not fill the whole space. □

Consider now the set of all nonempty subsystems of a dynamical system and order them by inclusion. Then minimal subsystems are precisely the systems minimal for this order. Given a totally ordered family of nonempty subsystems, their intersection is again a subsystem; indeed, it is closed and invariant. From compactness, it is also nonempty.

Zorn's lemma states that in any partially ordered set, if any totally ordered subset has a lower bound, then there exists a minimal element. Zorn's lemma is equivalent to the axiom of choice, which states that the cartesian product of a family of nonempty subsets is nonempty; see [HJ84]. Applying Zorn's lemma to the nonempty subsystems ordered by inclusion, we get a theorem of Birkhoff:

Proposition 2 *Every nonempty dynamical system has a minimal subsystem.*

The study of minimal systems is pursued in Section 3.2.

2.4 Input-output discrete-time systems

Input-output discrete-time dynamical systems, input-output systems for short, are more general than discrete-time dynamical systems. They are needed in Chapters 5 and 6. An input-output discrete-time dynamical system is given by a set X of *points*, *states* or *configurations*, a set U of *inputs* and a set Y of *outputs*, a map $f : X \times U \to X$ and a map $h : X \times U \to Y$. Note that if we forget the output set Y and the function h, then an input-output system is essentially the same as the system (X, U^*), where the monoid U^* is the set of finite sequences of elements of U, with concatenation as internal product, and where the action of $u \in U \subseteq U^*$ is the map $x \mapsto f(x, u)$. Such a system can be thought of as modelling the following situation.

At every step of time, the environment feeds the system with an element $u \in U$, the state of the system is updated from x to $f(x, u)$ and y, meant as the observation performed by the environment during the transition, is set to $h(x, u)$. We must know the state of the system at time zero and the input given by the environment at every time in order to compute all the successive values of x and y.

Sometimes we assume that the environment is itself an input-output discrete-time system, called the *controller*, whose input is the output of the system and whose output is the input of the system; see top of Figure 2.1. The system and the environment alternatively update their states. To determine the sequence of states of the system in the course of time, it is enough to know the initial states of the system and the environment, and the first input given to the system by the environment.

We can also model the environment by a *feedback map* which computes the next value of u as a function of the past values of y; see bottom of Figure 2.1. This is equivalent to modelling the environment as a system. Indeed every

such function can be represented by an input-output system whose state space is the set of all possible finite sequences of past values of y, and conversely, every input-output system with an initial state can be represented by function mapping a finite sequence of inputs (here a sequence of values of y) to the last output of the system (here u).

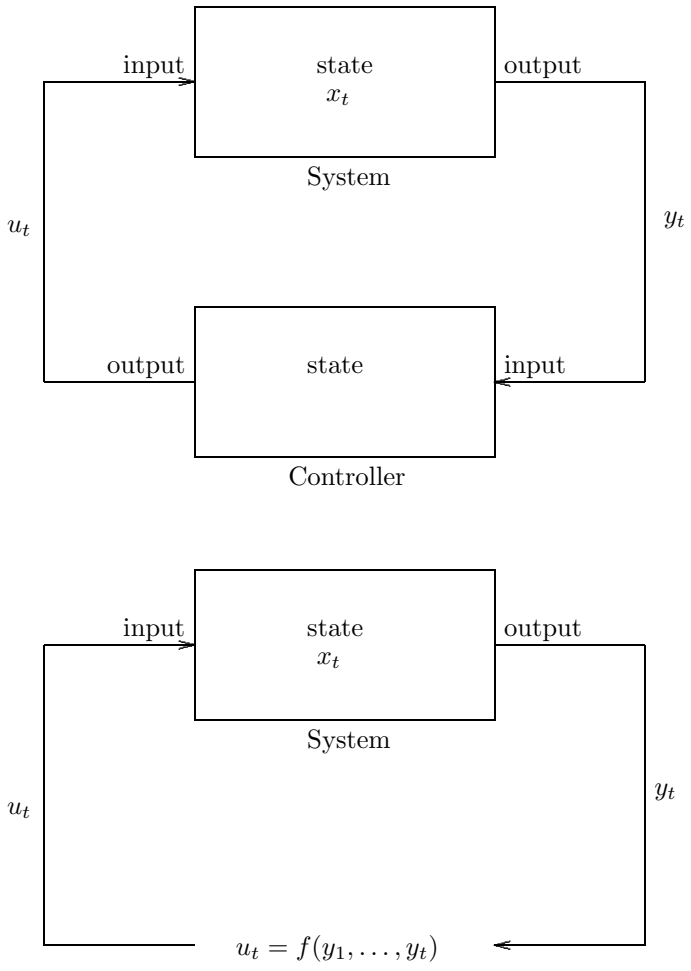

Fig. 2.1. An input-output dynamical system is interacting with an environment, modelled by another input-output dynamical system or by a feedback map. The values of u, x, y are indexed by the time t.

When X, U and Y are finite sets and we specify an initial state in X, then the system is called an *input-output finite automaton*, which is the central object of Chapter 5. An example of input-output finite automaton is given on Figure 5.1.

For input-output systems, we can put an additional structure on the set X. In Chapter 6 we suppose that X is endowed with a probability measure and that for every $u \in U$, the map $f(.,u) : X \to X$ is measurable and preserves measure, i.e., the preimage of a measurable set is a measurable set of same measure.

Such systems are used in control theory, where a typical problem is the following: given an input-output discrete-time system, design an environment (under the form of another input-output system or of a feedback map) such that the initial state is finally driven to a prescribed final state. Such a problem is studied in Chapters 5 and 6.

2.5 Languages and automata

We review the basics of the theory of formal languages and finite automata, as we need these notions in all subsequent chapters. Extensive treatments can be found in [Lot97, PP04].

An *alphabet* is a finite set, the elements of which are called the *letters* or *symbols*. A (finite) *word* over the alphabet A is a finite sequence of letters. By convention, there is exactly one word of length 0, called the *empty word*. The set of words over A is denoted A^*. An *infinite word* over A is a sequence of letters indexed by the natural integers. The set of infinite words over A is denoted $A^{\mathbb{N}}$ or A^ω. A *language* over A is a subset of A^* and an *ω-language* over A is a subset of $A^{\mathbb{N}}$. The *concatenation* of a word u and a (finite or infinite) word v is the word uv. The word v is a *subword* of the (finite or infinite) word t if t can be written as uvw; if u is the empty word then v is a *prefix* of t.

If L is a language and L' is a language or an ω-language, then LL' denotes the language or ω-language obtained by concatenation of all elements of L with all elements of L'. The expression L^n denotes the language $LL \dots L$ (n times). The expression L^* denotes $L \cup L^2 \cup L^3 \cup \cdots$ and L^ω denotes $LLLL\dots$. Finally, a letter a sometimes denotes the language $\{a\}$, depending on the context. For instance, a^n stands for $aa \dots a$ (n times) or $\{aa \dots a\}$, and the expression 0^*10^ω is the set of infinite words over $\{0,1\}$ with exactly one occurrence of 1.

A *prefix language* is a language in which no word is a prefix of another word. Suppose that a prefix language over N letters is endowed with a probability distribution. *Shannon's noiseless coding theorem* states that the expected length of a word is at least the entropy in base N of the distribution. Shannon's theorem provides a lower bound on the shortest possible encoding of a finite or countable set of events into words over a finite alphabet. Let us see a

proof. Let L be a binary (say) prefix language. Play head and tail with a fair coin an infinite number of times. The probability that the sequence of tosses begins by a binary word u is $q_u \triangleq 2^{-\text{length}(u)}$. For any $u, v \in L$ the events that the sequence begins by u or by v are disjoint. Hence the probability that the sequence starts by a word of L is $\sum_{u\in L} q_u \leq 1$. The expected length of a word of L is $\sum_{u\in L} p_u \text{length}(u) = -\sum_{u\in L} p_u \log_2 q_u$. Hence

$$\begin{aligned} -\sum_{u\in L} p_u \log_2 p_u - \sum_{u\in L} p_u \text{length}(u) &= \sum_{u\in L} p_u \log_2(q_u/p_u) \\ &\leq (\ln 2)^{-1} \sum_{u\in L} p_u(q_u/p_u) - 1) \\ &= (\ln 2)^{-1}(\sum_{u\in L} q_u - 1) \\ &\leq 0, \end{aligned}$$

which is the desired result. We used the fact that $\ln 2 \log_2 x = \ln x \leq x - 1$. A reference on information and coding theory is [Abr63].

A *finite automaton* is given by a finite *set of states* Q, a set of *initial* states $Q_0 \subseteq Q$, a set of *final* states $Q_1 \subseteq Q$, an *input alphabet* A and a transition function $\Delta : Q \times A \to 2^Q$. The transition function is extended to $\Delta : Q \times A^* \to 2^Q$ by $\Delta(q, ua) = \Delta(\Delta(q, u), a)$. A language $L \subseteq A^*$ is *regular* if there exists a finite automaton which *accepts* L, i.e., $u_0u_1u_2 \dots u_n \in L$ iff there exists $q_0q_1q_2 \dots q_{n+1}$ with $q_0 \in Q_0$, $q_{i+1} \in \Delta(q_i, u_i)$ and q_{n+1} in Q_1. A finite automaton can be represented graphically. For instance, the automaton accepting the language 0^*10^* is represented on Figure 2.2. *Kleene's theorem* states that regular languages over an alphabet A are exactly those languages that can be defined by an expression using the empty word, the letters of A, concatenation, union, and $.^*$ operation; see [PP04]. Such expressions are called *regular expressions*. The proof of Kleene's theorem is constructive: it provides an explicit way to construct a regular expression from a finite automaton and conversely.

A *deterministic* finite automaton is a finite automaton with only one initial state and such that the subset $\Delta(q, a)$ has exactly one element for every q and a. So we can write Δ as a function from $Q \times A$ to Q. Notice the similarity with input-output automata defined in Section 2.4; the output function is here replaced by a set of final states. It is easy to prove that a language is accepted by a finite automaton if and only if it is accepted by a deterministic finite automaton. The trick is to replace the set of states Q of a finite automaton by the power set 2^Q; then the transition function becomes deterministic.

An ω-language $L \subseteq A^{\mathbb{N}}$ is *ω-regular* if there exists a finite automaton which *accepts* L, i.e., $u \in L$ iff there exists at least one sequence $q_0q_1q_2 \dots$ such that $q_0 \in Q_0$, $q_{i+1} \in \Delta(q_i, u_i)$ for all i, and $q_j \in Q_1$ for infinitely many j. A finite automaton intended to accept an ω-language is called a *Büchi* automaton (there

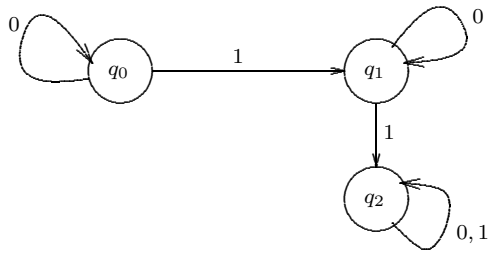

Fig. 2.2. A deterministic finite automaton accepting the language 0^*10^*. The vertices are the states, and the edges are labelled with the input alphabet. The initial state is q_0 and the final state is q_1. On the input word 0010, the automaton successively passes through the states q_0, q_0, q_0, q_1 and q_1 which is a final state, hence the input word is accepted.

is no formal difference between a finite automaton and a Büchi automaton). It is not always possible to accept an ω-regular language with a deterministic Büchi automaton. For instance, the ω-language $\{0,1\}^*0^\omega$ of infinite words with finitely many 1s is accepted by the nondeterministic Büchi automaton represented on Figure 2.3, but cannot be accepted by a deterministic Büchi automaton. We prove the latter fact, following [PP04]. Let $\Delta : Q \times \{0,1\} \to Q$ be the transition function of a deterministic Büchi automaton accepting $\{0,1\}^*0^\omega$, with initial state q_0 and final states Q_1. Since 10^ω is in the ω-language, $\Delta(q_0, 10^i) \in Q_1$, for some i. Since 10^i10^ω is in the ω-language, $\Delta(q_0, 10^i10^j) \in Q_1$, for some j. Since $10^i10^j10^\omega$ is in the ω-language, $\Delta(q_0, 10^i10^j10^k) \in Q_1$, for some k, etc. Then the word $10^i10^j10^k1\ldots$ is also accepted by the automaton, but is not in the ω-language: contradiction.

An adaptation of Kleene's theorem states that ω-regular languages over an alphabet A are exactly those languages that can be expressed by an *ω-regular expression*, that is, an expression of the form $S_1(T_1)^\omega \cup \cdots \cup S_n(T_n)^\omega$, where S_i and T_i are regular expressions. The proof is constructive; see [PP04].

A *Muller automaton* consists of a finite set of states Q, a transition function $\Delta : Q \times A \to Q$, an initial state $q_0 \in Q$ and a family $\mathcal{F}$ of subsets of Q. Thus a Muller automaton is basically a deterministic finite automaton with several sets of final states. A language $L \subseteq A^{\mathbb{N}}$ is *accepted* by a Muller automaton if $u \in L$ iff the set $\{q \in Q : \ \forall n, \exists m > n : \Delta(q_0, u_0 \ldots u_{m-1}) = q\}$ of states visited infinitely often when u is read as input belongs to $\mathcal{F}$. For example, the ω-language $\{0,1\}^*0^\omega$ is accepted by the Muller automaton represented on Figure 2.4.

McNaughton's theorem essentially states that a language $L \subseteq A^{\mathbb{N}}$ is accepted by a Büchi automaton if and only if it is accepted by a Muller automaton; see [PP04] for instance. Hence Muller automata are exactly as powerful as Büchi automata. The proof of McNaughton's theorem is constructive. Although Büchi

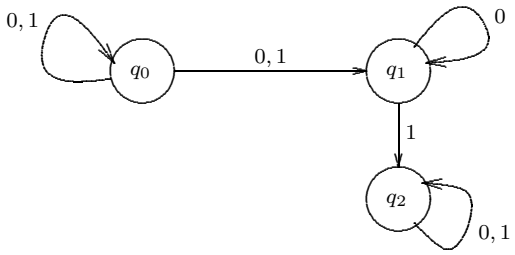

Fig. 2.3. A Büchi automaton accepting the ω-language $\{0,1\}^*0^\omega$. The initial state is q_0 and the final state is q_1. Notice the nondeterminism: from q_0, whatever the input is, the automaton can jump either to q_0 or to q_1.

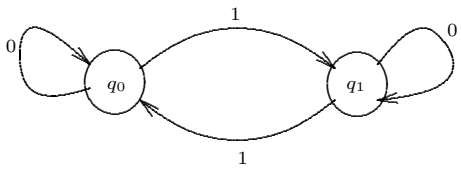

Fig. 2.4. A Muller automaton accepting the ω-language $\{0,1\}^*0^\omega$. The initial state is q_0 and the family of subsets is $\{\{q_0\},\{q_1\}\}$.

automata are simpler to define, Muller automata are deterministic, which is sometimes an advantage. Note that several other formalisms, including the so-called μ-calculus and monadic second-order formulae, have been proved equivalent to Büchi and Muller automata; see [PP04, Kai96, JGP99].

2.6 Symbolic spaces and symbolic discrete-time dynamical systems

As mentioned above, we consider dynamical systems taking place on compact metric spaces. Of special interest for Chapters 3 and 4 are the so-called symbolic spaces. A *symbolic space* is a compact Hausdorff space for which the closed open sets (called the *clopen sets*) form a countable basis: every open set is a union of clopen sets. A typical example of symbolic space is the *Cantor set* $\{0,1\}^{\mathbb{N}}$ endowed with the metric $d(x,y) = 0$ if $x = y$ and $d(x,y) = 2^{-n}$ if $x \neq y$, where n is the index of the first bit on which x and y differ. For instance, $d(0^\omega, 0^n 1 \ldots) = 2^{-n}$. This metric satisfies the ultrametric inequality. The topology given by this metric is often called the *product topology* on the Cantor set, because it is indeed the topology of the product of $\mathbb{N}$ copies of the discrete space $\{0,1\}$, as defined in Section 2.1. Let us prove that the *Cantor*

space, i.e., the Cantor set endowed with the product topology, is indeed a symbolic space.

If $w \in \{0,1\}^*$ (the set of finite binary words), then $[w]$ denotes the set of all infinite sequences beginning by w. In fact, sets of this form, usually called *cylinders*, are exactly the (open and closed) balls of the metric space, as readily seen. Hence cylinders are clopen sets and form a basis for the metric space.

To show compactness, let us take a sequence of points. There is a subsequence either in [0] or in [1], or both. Let x_0 be 0 or 1 such that there is a subsequence in $[x_0]$. Similarly, there is subsequence of this subsequence in $[x_0x_1]$, for $x_1 = 0$ or 1. There is a subsequence of this new subsequence in $[x_0x_1x_2]$,etc. Now take one point of the sequence in $[x_0]$, one further point of the sequence in $[x_0x_1]$, one further point of the sequence in $[x_0x_1x_2]$, etc. This subsequence converges to the configuration $x_0x_1x_2\ldots$.

Note that any clopen set of the Cantor space $\{0,1\}^{\mathbb{N}}$ is a finite union of cylinders, since any clopen set is compact (because closed in a Hausdorff compact space) and is the union of cylinders (because open).

The same definition of distance is extended almost immediately to the spaces $\{0,1\}^* \cup \{0,1\}^{\mathbb{N}}$, $A^{\mathbb{N}}$, $Q \times A^{\mathbb{Z}}$, $A^{\mathbb{Z}^d}$ where Q and A are finite and d is a positive integer. For instance, the set $A^{\mathbb{Z}^2}$ is endowed with the metric $d(u,v) = 0$ if $u = v$ and $d(u,v) = 2^{-n}$ if $u \neq v$, with n the smallest integer for which there are i and j such that $n = \min\{|i|,|j|\}$ and $u_{ij} \neq v_{ij}$. If a square pattern of odd size $2s+1$ is centered at the origin, then the set of configurations extending that pattern is an open ball of radius 2^{-s}. Conversely, to an open ball of radius $r < 1$ is associated a square pattern of size $2\lfloor -\log_2 r \rfloor + 1$.

Closed subsets of the Cantor space, endowed with the subset topology, are symbolic spaces themselves. The converse is well known to hold as well:

Proposition 3 *Every symbolic space is homeomorphic to a closed subset of the Cantor space. Every perfect symbolic space is homeomorphic to the Cantor space.*

Proof. Let X be an effective symbolic space, and let $(P_n)_{n\in\mathbb{N}}$ an enumeration of the clopen sets of X. For every point $x \in X$, construct the infinite sequence $g(x) \in \{0,1\}^{\mathbb{N}}$, where $g(x)_n = 1$ if and only if $x \in P_n$. Then the map $g : X \to \{0,1\}^{\mathbb{N}}$ is injective, because every two points are separated by a clopen set (i.e., there is a clopen set that contains one point but not the other). It is continuous because the preimage of a clopen set of $\{0,1\}^{\mathbb{N}}$ is clopen in X, and because the clopen sets are a basis of the topology of $\{0,1\}^{\mathbb{N}}$. As X is compact, $g(X)$ is closed.

If the space is perfect, then we construct another map $h : X \to \{0,1\}^{\mathbb{N}}$. We may write X as a partition of two clopen sets $X = X_0 \cup X_1$, where X_0 is the clopen set of smallest index to be different from X and $\emptyset$; this is always possible thanks to perfectness. Suppose that we have already constructed X_u,

where u is a binary word. Let n be the first index such that $X_u \cap P_n$ differs from both X_u and $\emptyset$. Set $X_{u0} = X_u \cap P_n$ and $X_{u1} = X_u \setminus P_n$. For $x \in X$ let $h(x) \in \{0,1\}^{\mathbb{N}}$ be such that $x \in X_u$ for all prefixes u of $h(x)$. We claim that $h : X \to \{0,1\}^{\mathbb{N}}$ is a homeomorphism.

First we prove that $h(x)$ is well defined. From construction, X_u and X_v have a nonempty intersection if and only if u is a prefix of v or v is a prefix of u. Hence for a point x, the set of words u such that $x \in X_u$ is totally ordered by the relation 'is a prefix of', and there is a unique limit infinite word.

We now consider the map $h' : \{0,1\}^{\mathbb{N}} \to X$ sending the configuration $u_0u_1u_2\ldots$ to the unique point of $X_{u_0} \cap X_{u_0u_1} \cap X_{u_0u_1u_2} \cap \cdots$. Let us show that it is indeed unique. Any two distinct points x and y in $X_{u_0} \cap X_{u_0u_1} \cap X_{u_0u_1u_2} \cap \cdots$ are separated by some clopen set P_m. As only finitely many clopen sets have an index smaller than m, P_m must have been taken into consideration in the process of refining $X_{u_0u_1\ldots u_i}$ into $X_{u_0u_1\ldots u_i0}$ and $X_{u_0u_1\ldots u_i1}$ (for some i). As P_m separates x from y (both in $X_{u_0u_1\ldots u_i}$), then we have, say, $X_{u_0u_1\ldots u_iu_{i+1}} = X_{u_0u_1\ldots u_i} \cap P_m$. Then x or y belongs to $X_{u_0u_1\ldots u_iu_{i+1}}$, but not both: contradiction.

We easily check that h and h' are inverse maps. They are continuous, because they map clopen sets to clopen sets. Hence h is a homeomorphism. □

For instance, the perfect space $A^{\mathbb{Z}^d}$, where A is finite and d is a positive integer, is homeomorphic to $\{0,1\}^{\mathbb{N}}$. The set of finite and infinite words $\{0,1\}^* \cup \{0,1\}^\omega$ is homeomorphic to the set $\{10,11\}^*(00)^\omega \cup \{10,11\}^\omega \subseteq \{0,1\}^\omega$.

Hence every symbolic space is metrizable with a metric that satisfies the ultrametric inequality and such that there are finitely many balls of radius ϵ, for any $\epsilon > 0$.

A *symbolic discrete-time system* is a continuous transformation of a symbolic space.

An *invariant* subset of the discrete-time symbolic system $f : X \to X$ is subset $Z \subseteq X$ such that $f(Z) \subseteq Z$. A closed invariant subset Z endowed with the subset topology is a symbolic space, and f restricted to Z is a *subsystem* of $f : X \to X$. The discrete-time symbolic systems $f : X \to X$ and $g : Y \to Y$ are *conjugated* if there exists a homeomorphism $h : X \to Y$ such that $h \circ f = g \circ h$. Two conjugated systems are seen as 'essentially the same'. If $h : X \to Y$ is a surjective map (rather than bijective), then the system $g : Y \to Y$ is said to be a *factor* of $f : X \to X$ and h is the *factor map*. The factor g can be seen as a 'simplification' of f. These are of course the specializations of the corresponding notions for general dynamical systems defined in Section 2.3.

The identity on any symbolic space is perhaps the simplest example of symbolic discrete-time system. Another simple symbolic system is the map $f : \{0,1\}^\omega \to \{0,1\}^\omega : x \mapsto 0x$, where all orbits converge to 0^ω.

A one-sided or two-sided *shift* is a discrete-time dynamical system on $A^{\mathbb{N}}$ or $A^{\mathbb{Z}}$ (where A is a finite alphabet) with the map $\sigma : A^{\mathbb{N}} \to A^{\mathbb{N}}$ or $\sigma : A^{\mathbb{Z}} \to A^{\mathbb{Z}}$

defined by $\sigma(x)_i = x_{i+1}$. This map is obviously continuous. A *subshift* is a subsystem of a shift, i.e., a closed subset of the Cantor space that is invariant under the shift map. In particular, a subshift is an ω-language and the set of subwords appearing at least one infinite word of the subshift is the *language* of the subshift. Subshifts are important partly because any discrete-time symbolic system can be 'approximated' by a subshift, as we now explain.

From any symbolic discrete-time dynamical system, we can generate one-sided subshifts in a natural way. A *clopen partition* of a symbolic space is a partition of the space into a finite number of disjoint clopen sets. A partition $\mathcal{C}$ is *finer* than $\mathcal{C}'$, or $\mathcal{C}'$ is *coarser* than $\mathcal{C}$, if every clopen set of $\mathcal{C}$ is included in some clopen set of $\mathcal{C}'$. Given a clopen partition $\mathcal{C} = \{C_1, \dots, C_N\}$ of X, the subshift *induced* by this partition is the set of infinite words $c_0c_1c_2c_3 \dots \in \mathcal{C}^{\mathbb{N}}$, such that there is a point in c_0 whose trajectory goes successively through c_1, c_2, c_3, ... Note that here C_1, say, is both a subset of X and a symbol from a finite alphabet. Thus $C_1C_3C_1$, for instance, denotes a word of three symbols over the alphabet $\mathcal{C}$. The language of the subshift is also said to be *induced* by the partition. An induced subshift is a factor of the system and conversely any factor subshift is induced by the clopen partition $\{h([a_1]), \dots, h([a_N])\}$, where h is the factor map and $\{a_1, \dots, a_N\}$ is the alphabet of the subshift.

A subshift is *sofic* if its language is regular. A subshift is sometimes given by a set of *forbidden words*: the subshift is described as the largest set of infinite words that do not have any of the forbidden words as subword. A subshift is of *finite type* if it can be defined by a finite set of forbidden words. In other words, a subshift a finite type is the set of infinite words whose orbit under the shift map does not intersect Z, for some clopen set Z; such a set is always closed, because it is the intersection of all sets $\sigma^{-n}(Z^c)$, where $.^c$ denotes the complement. Sofic subshifts are exactly factor subshifts of subshifts of finite type, as shown by Weiss; see [Kůr04] for a proof.

In the next sections, other nontrivial examples of symbolic discrete-time dynamical systems are given: Turing machines and cellular automata.

2.7 Turing machines and computability

We now detail the definition of Turing machine, insisting on the fact that Turing machines are dynamical systems. We then review the basic theory of computability. These notions are useful in Chapters 3 and 4.

A (multi-tape, deterministic) *Turing machine* is given by a head and a finite set of tapes, infinite on left and right sides, divided into cells. Every cell contains a symbol taken in a finite alphabet. At any time the head can read the symbol of exactly one cell on each tape, write a (possibly) new symbol in the same cell, and make each tape move one cell to the right or one cell to the left. The head is described by a finite set of internal states and a transition function. Given

the set of symbols currently read on the tapes and the present internal state of the head, the transition function tells the head which operations to perform on each tape and updates the internal state. An example of Turing machine is given on Figure 1.2.

The space of configurations of a Turing machine is thus given by $X = Q \times A_1^{\mathbb{Z}} \times \cdots \times A_k^{\mathbb{Z}}$, where k is the number of tapes, $A_1, \ldots, A_k$ the alphabets and Q the set of internal states of the head. The head is formally an input-output system with $A_1 \times \cdots \times A_k$ as input set, $A_1 \times \cdots \times A_k \times \{\text{Left}, \text{Right}\}^k$ as output set and Q as state set.

The space X is a compact metric space, with a metric similar to the metric of the Cantor space $\{0,1\}^{\mathbb{N}}$. As it is a perfect symbolic space, it is homeomorphic to the Cantor space. Hence a Turing machine is a symbolic discrete-time dynamical system. This corresponds to the model of Turing machine with moving tape studied by Kůrka [Kůr97b].

A variant often encountered in the literature is the *Turing machine with blank symbol*. Every tape is supposed to be of finite length at any time. Hence the configurations are $A^* \times Q \times A^*$ if one tape (and similarly for several tapes). The first occurrence of A^* denotes the content of the tape to the left of the head and the second A^* denotes the right part of the tape. Of course, the head may increase the length of the tape. One may think the rest of the infinite tape as filled with all blank symbols. In other words, one may think of A^* as $A^* b^\omega$, where b is the blank symbol.

However, A^* is not naturally equipped with a compact topology, so we consider the compactification $W = A^* \cup A^{\mathbb{N}}$, i.e., the set of finite and infinite binary words. Then the Turing machine function is also defined on $W \times Q \times W$ (if there is only one tape), which is a compact space, whose isolated points are $A^* \times Q \times A^*$. Note that an isolated point is clopen in $W \times Q \times W$. In conclusion, a Turing machine with a blank symbol is a symbolic system on the space $W \times Q \times W$.

Note that the 'state' of a Turing machine usually refers to the the internal state of the head, rather than the whole content of the head and of the tapes, for which we rather use the term 'configuration'.

Considering Turing machines as dynamical systems is a rather recent point of view, adopted for instance in [BCN02, Hoo66, Kůr97b, Moo91]. Turing machines were originally designed as a model of computation, that is, as a formal definition of algorithm or effective procedure. We now quickly review the classical theory of Turing machines and computability; more details can be found in references such as [Cut80, Rog87].

Turing [Tur36] argued in the following manner to conceive his machine. An algorithm is informally defined as a finite set of nonambiguous and finitary instructions to manipulate finite objects, such as symbols, integers, graphs, etc. The aim of an algorithm is usually to compute a function (e.g., 'Given $n \in \mathbb{N}$, what is the square of n?') or to decide membership to a class of finite objects

(e.g., 'Given $n \in \mathbb{N}$, is n a perfect square?'). By a finite object, we mean that it can be characterized by a finite description. Turing imagined a human being applying an algorithm to compute a function or decide a property. The human being is assumed to be at every step of the algorithm in a unique state of mind, and is supposed to have finitely many possible states of mind. Hence he/she can only distinguish finitely many symbols. The data of the problem is written on a sheet of paper under the form of a finite sequence of symbols. He/she writes or deletes intermediate and final results on other sheets of paper, available in any quantity. Paper is modelled by the tape(s), and the human operator is modelled by the head. We now describe in little more detail how a Turing machine is used to perform a formal computation.

First note that all finite objects (pairs of integers, rational numbers, finite words, automata, etc.) can be encoded into natural integers, or equivalently into binary words, in an effective way. What is meant exactly by 'effective' remains nonformalized.

Properties, sets and problems are customarily identified. Given a natural integer (or a word or a Turing machine or a graph, etc.), called the *instance*, the *problem* of checking whether a *property* is verified by a natural integer (or, etc.) is identified to the set of *positive instances*, i.e., of natural integers (or, etc.) that satisfy the property. A subset of $\mathbb{N}$ can also be identified to the characteristic function $\mathbb{N} \to \{0, 1\}$. And a function $k : \mathbb{N} \to \mathbb{N}$ can be identified to the set of couples $(n, k(n))$. Hence it is enough to say what an algorithmically computable function is, or to say what an algorithmically decidable set is.

So a Turing machine is used to compute a function $k : \mathrm{Dom}(k) \subseteq \mathbb{N} \to \mathbb{N} : n \mapsto k(n)$ (we allow partial functions). A convention must be taken as how to encode the initial data —a natural integer n— on the tape. For instance, n is written in binary on the right side of the head, and the rest of the tape is filled with blank symbols; the other tapes, if any, are also filled with blank symbols. A similar convention can be taken if there is no blank symbol. The head is initially in some *initial state*, and we let the Turing machine run on this initial configuration. There is a special state of the head, called the *halting state*. Whenever we reach it, we look at the content of the tape(s). From this content we can decode in an 'effective' way an integer, which is $k(n)$. The partial functions for which such a Turing machine exists are called *computable*, and every Turing machine computes a partial function. It is important to note that a computable function may be partial, because a Turing machine may never reach the halting state from initial configuration.

A computable function is in a sense a finite object, since it can be described by a corresponding Turing machine (although this description is not unique). Hence it makes sense to ask whether a given partial function is defined on a given number. Turing proved this problem, called the *halting problem*, to be undecidable, i.e., its characteristic function is not computable. An equivalent form of the halting problem is the following. Given a Turing machine and an

initial finite configuration ('finite' because almost all cells are filled with a blank symbol, or another fixed symbol), does the Turing machine reach the halting state?

A set is *recursively enumerable* or *r.e.* if it is the domain of a computable partial function. The class of r.e. problems is denoted Σ_1.

A problem is *(many-one) reduced* to another problem if there exists a total computable function, called a *(many-one) reduction* that maps positive instances of the first problem to positive instances of the second problem and negative instances of the first problem to negative instances of the second problem. Intuitively, the second problem is at least as hard as the first one: if you can solve the second problem, you can solve the first one by first translating an instance of the first problem into an instance of the second problem. We consider two problems to be equivalent if they can be reduced to one another.

Given a class Σ of problems, for instance the r.e. problems, a *Σ-hard problem* is one to which every problem of Σ can be reduced. A *Σ-complete* problem is a problem of Σ that is Σ-hard. Intuitively, a Σ-complete problem is 'the most difficult problem of Σ'. The class *co-Σ* denotes the problems whose complement is in Σ.

The halting problem is r.e.-complete, almost by definition. A common technique to prove that a problem is undecidable is to show that it is r.e.-complete, by a reduction from the halting problem.

According to Turing, a Turing machine is *universal* if it can simulate any other Turing machine, i.e., it computes the partial function that maps a Turing machine m and a number n to the output computed by m when given n as initial data. Such Turing machines, both with or without blank symbol, even with one tape, are known to exist. If we fix once for all a universal Turing machine, an equivalent form of the halting problem is to check, given a finite initial configuration, whether the universal Turing machine halts on this configuration.

An *oracle Turing machine* is one which has a special tape called the *oracle tape*, whose content is interpreted as an *oracle* function from $\mathbb{N}$ to $\mathbb{N}$. For instance an arbitrary total function $k : \mathbb{N} \to \mathbb{N}$ can be encoded in the form $10^{k(0)}10^{k(1)}10^{k(2)}1\ldots$ at the right side of the head. Note that not every initial content of the oracle tape encodes a correct function from $\mathbb{N}$ to $\mathbb{N}$. The Turing machine can use the oracle tape to query the value of the oracle function on a natural integer. Of course, the function computed by an oracle Turing machine depends on the oracle given as initial content of the oracle tape. If the oracle function is computable, the function computed by the Turing machine is computable too. An oracle Turing machine provided with a noncomputable oracle function may compute a noncomputable function. A universal oracle Turing machine is one which can simulate any other oracle Turing machine, in a sense similar to universal Turing machines. Universal Turing machines with an oracle tape and another tape, encoding configurations with or without blank symbol, can be constructed.

Church-Turing thesis claims that Turing machines are the right formalization of 'algorithm' or 'effective procedure'. Hence, according to the thesis, every procedure described in English or another natural language that computes a function or decides a problem and that is 'effective' (in the intuitive meaning) can be translated into a corresponding Turing machine. Church-Turing thesis cannot be formally proved, but is supported by several strong arguments. One of them is the fact that a Turing machine is a model of a human being performing a computation, as explained above. In addition to that, no intuitively computable function has ever been found that cannot be computed by a Turing machine. Another argument is the fact that other attempts to formalize the notion of computable function, such as Church's λ-calculus or Kleene's recursive functions, have led to the same class of functions than those computable by a Turing machine. Church-Turing thesis is widely accepted by computer scientists, and it is adopted in this text without restriction. In particular, we shall consider the words 'computable', 'effective' or 'recursive' as mere synonyms.

Church-Turing thesis simplifies proofs a lot. For instance, consider the problem of primality: we must determine whether a given integer n is prime. There is an simple, informal algorithm to solve it: for all numbers i between 2 and $n-1$, check with the method of euclidian division whether i divides n. This algorithm can clearly be carried out in a mechanical way by a human being. Hence Church-Turing allows us to conclude that there is a Turing machine that, when given the encoding of an integer, decides primality.

As another example, the existence of a universal Turing machine becomes trivial if we admit Church-Turing thesis, since simulating a given machine on a given initial configuration until it stops is obviously effectively feasible by a human brain, thus it is feasible by some Turing machine.

The undecidability of the halting problem is also proved easily. Suppose by contradiction that it is decidable, Then you can build a Turing machine that, given n, halts if the Turing machine n does not halt on the data n, and does not halt if the Turing machine n halts on the data n. But this Turing machine is coded by a number, say m. Does m halt on the data m? The answer leads to a contradiction. The existence of the machine m is provided by Church-Turing thesis.

A set which is r.e. and co-r.e. is decidable, and that can also be proved easily. Indeed, if a natural integer (say) n is in the set, there is an effective procedure to prove it in finite time. If n is not in the set, then there is an effective procedure to prove it as well. Given n, we can launch both procedures in parallel and see which one ends first. Church-Turing thesis prevents us from the necessity to describe a Turing machine that simulates two Turing machines in parallel.

Rice's theorem [Ric53] goes as follows. A 'nontrivial property' of computable functions is a subset of computable functions different from the empty set and the full set of computable functions. Such a property is undecidable, i.e., one

cannot decide whether a given a Turing machine computes a function that satisfies the property.

Let us outline the proof. We suppose, without loss of generality, that the empty function (i.e., the nowhere defined function) does not satisfy the nontrivial property, and let k be a partial function that does. Given a Turing machine m and an initial finite configuration n, construct a new Turing machine that, given a data p, simulates m on n until it stops and then computes $k(p)$. The function computed by the new Turing machine satisfies the nontrivial property if and only if m halts on n. Thanks to Church-Turing thesis, the construction of the new Turing machine is computable. Hence we have reduced the halting problem to the problem of deciding whether a computable function satisfies the nontrivial property. Hence the latter problem is undecidable.

2.8 Cellular Automata

We now define cellular automata, which are used as examples of symbolic discrete-time systems in Chapter 4.

Given a vector $v \in \mathbb{Z}^d$, the *shift* map associated to v is the map from $A^{\mathbb{Z}^d}$ to $A^{\mathbb{Z}^d}$ that maps $x = (x_u)_{u\in\mathbb{Z}^d}$ to $x = (x_{u+v})_{u\in\mathbb{Z}^d}$. This is a generalization of the shift map on $A^{\mathbb{Z}}$, defined in Section 2.6 . A *cellular automaton* is a symbolic discrete-time system on $A^{\mathbb{Z}^d}$ that commutes with all shifts maps, for an alphabet A and an integer $d \geq 1$ called the *dimension*. The identity, the shifts and the map sending every configuration to $a^{\mathbb{Z}^d}$ (for a letter a) are trivial cellular automata.

A more combinatorial definition is the following. Given a finite set of vectors $V \subseteq \mathbb{Z}^d$ and a *table* $A^V \to A$, we define a cellular automaton as a map on $A^{\mathbb{Z}^d}$ with the following property. The image of a configuration $x = (x_u)_{u\in\mathbb{Z}^d}$ is the configuration $y = (y_u)_{u\in\mathbb{Z}^d}$ such that for every u, the letter y_u is given by the table applied to the list $(x_{u+v})_{v\in V}$.

The equivalence between these two definitions was shown by Hedlund [Hed69] in 1969.

Introduced in the 1950s and 1960s by Ulam [Ula52], von Neumann [vN66] et Zuse [Zus69], cellular automata have received a lot of attention over the years, as discrete-time dynamical systems and as models of computation.

For instance, a great deal of work has been done on the classification of cellular automata according to their dynamical and computational properties; let us mention the classifications of Wolfram [Wol02], Čulik et al. [ČHY90], Gilman [Gil87], Kůrka [Kůr97a].

Conway's *Game of Life* [Gar70] is a classical example. On $\mathbb{Z}^2$, every cell is 'dead' or 'alive'; at every step a cell that is alive remains so if two or three among the eight neighbors are alive and dies otherwise; a dead cell becomes alive if exactly three neighbors are alive.

Another well-known example is the one-dimensional *automaton 110*. The table is the following. The next value of a cell is 0 if the left neighbor of the cell, the cell and the right neighbor are in state 000, 100 or 111; otherwise the next value is 1. The evolution of a configuration is represented on Figure 2.5. For a detailed account, see [Wol02].

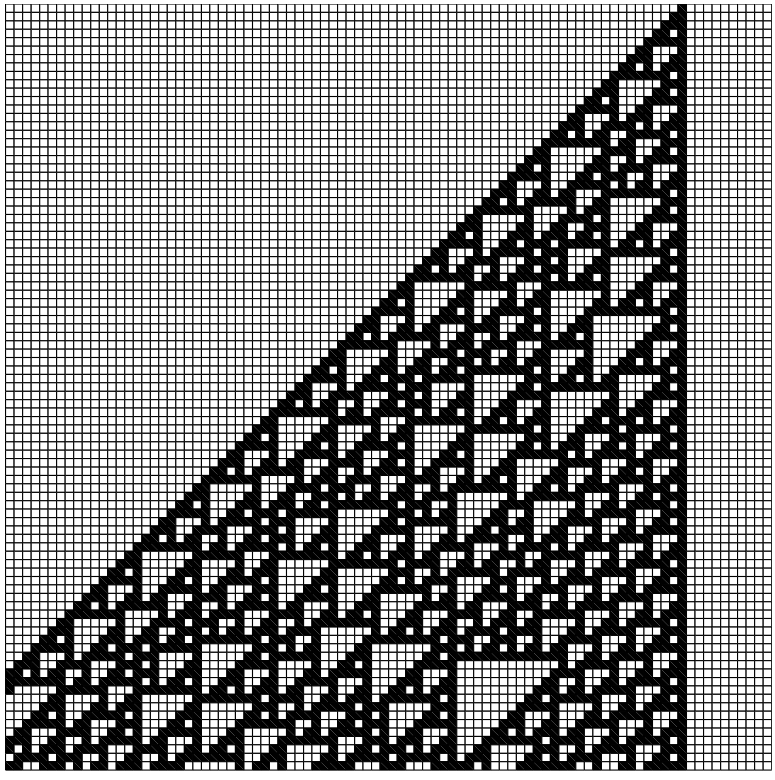

Fig. 2.5. The space-time diagram of the automaton 110. An empty cell is a 0, a black cell is a 1. The top row contains the initial configuration: all cells are set to 0 except one. Every row is the image of the row just above. This figure was produced with the help of Christian Schwarzer's Matlab Random Boolean Network Toolbox [Sch].

Game of Life and automaton 110 have been proved to be computationally universal [BCG85, DR99, Wol02], in the sense that given two spatially ultimately periodic configurations x and y, checking whether x belongs to the trajectory of y is r.e.-complete. By spatially ultimately periodic we mean that the configuration, as an element of $\mathbb{Z}^d$, is composed of an indefinitely repeated finite pattern, except perhaps on a finite number of symbols.

2.9 Tilings

A Wang tile is a square of unit size with colored borders. Let T be a finite set of colored Wang tiles. A tiling is a configuration of $T^{\mathbb{Z}^2}$ such that adjacent borders of tiles always have the same colors.

The set $T^{\mathbb{Z}^2}$ is endowed with the monoid of all shifts (this monoid is isomorphic to $\mathbb{Z}^2$ with $+$ and $(0,0)$). The shifts for $T^{\mathbb{Z}^2}$ are defined in Section 2.8.

The set of tilings is a subset of $T^{\mathbb{Z}^2}$ that is invariant under all shifts. Let Z be the clopen set of all configurations of $T^{\mathbb{Z}^2}$ that have a tiling error at tile $(0,0)$. Then the set of tilings is the closed set $\cap_{v\in\mathbb{Z}^2}\sigma_v^{-1}(Z^c)$, where σ_v is the shift associated to v and Z^c the complement of Z. Hence the set of tilings is a dynamical system. Tilings may be thought of as subshifts of finite type in dimension two. A simple example of set of Wang tile is given on Figure 1.1.

Tilings were introduced by Wang [Wan61] to study the decidability of a fragment of first-order logic. Wang asked whether the *domino problem* of checking if a given tile set can tile the whole plane is decidable. Berger [Ber66] answered negatively, and Robinson [Rob71] provided a simpler proof. Both proofs work by embedding the space-time diagram of a Turing machine into a tiling.

The compactness of $T^{\mathbb{Z}^2}$ leads to an interesting consequence. If the set of tilings is empty, then a set $\bigcap_{v\in V}\sigma_v^{-1}(Z^c)$ is already empty, for some finite $V\subseteq\mathbb{Z}^2$. Hence, if a tile set cannot tile the plane, it cannot tile a sufficiently large finite square of the plane. Subsequently, we can eventually know it by trying to tile larger and larger squares. The family of tile sets that cannot tile the plane is therefore recursively enumerable.

Furthermore, if a tile set can tile the plane periodically, then we can eventually know it, by testing all periodic patterns.

If it is true that every tile set that can tile the plane can tile it in a periodic way, as we might expect by considering simple examples, and as Wang himself conjectured, then the domino problem would be decidable, because both tiles sets that can tile the plane and tiles sets that cannot would be recursively enumerable. Hence the result proved by Berger and Robinson proves the existence of tiles sets that can tile the plane only in a nonperiodic way, and in fact both proofs explicitly construct such tiles sets. Nowadays small such tile sets, with no more than 13 tiles, have been constructed [ČI96, ČIK97].

3

Almost Periodic Configurations and Undecidable Dynamics

3.1 Introduction

In this chapter, we consider families of dynamical systems. Recall that a dynamical system is for us a compact metric space on which a monoid acts continuously, as stated in Section 2.3. The notions of metric space, compactness and continuity are explained in Section 2.1.

To illustrate our results, we are especially interested by three families of dynamical systems: subshifts of finite types, tilings and Turing machines restricted to immortal configurations. Subshifts of finite type are defined in Section 2.6 and tilings in Section 2.9. A Turing machine is here a discrete-time dynamical system $f : Q \times A_1^{\mathbb{Z}} \times \cdots \times A_k^{\mathbb{Z}} \to Q \times A_1^{\mathbb{Z}} \times \cdots \times A_k^{\mathbb{Z}}$, where Q is the set of internal states and $A_1, \ldots, A_k$ are the tape alphabets; see Section 2.7 for a full description. In the rest of this chapter, we suppose for simplicity of notation that $A_1 = \cdots = A_k = A$. There is no blank symbol. An internal state of Q is called the halting state. If we call Z the set of configurations whose internal state is the halting state, immortal configurations are those that never fall in Z. The set of immortal configurations, which is $\bigcap_{n\in\mathbb{N}} f^{-n}(Z^c)$, is closed and invariant under f, thus defines a dynamical system.

Hence the three kinds of systems we consider are constructed similarly, as the subsystems of all configurations avoiding a certain open set Z.

For a gentle introduction to this chapter, including motivation and literature background, we report the reader to Section 1.1.

Section 3.2 is a background section where we take a closer look at minimal dynamical systems, already defined in Section 2.3. We define almost periodic configurations, which are a well-known generalization of periodic configurations, and we describe the link between minimality and almost-periodicity. In Section 3.3, a function quantifying almost-periodicity is proposed, that generalizes similar functions already known for tilings and subshifts, and we prove that global convergence to a single configuration can be characterized in terms

of the almost-periodicity of that configuration. In Section 3.4 we sketch a Rice-like theorem, giving sufficient conditions for a property to be undecidable in a class of dynamical systems. In particular, the number of invariant subsets and the topological entropy of tilings and of Turing machines restricted to immortal configurations are uncomputable. Section 3.5 proposes some paths for future work.

3.2 Minimal dynamical systems and almost periodicity

In this section we define almost periodic configurations and prove that minimal systems (defined in Section 2.3) are exactly closed orbits of almost periodic configurations. We also show that minimal systems are perfect and uncountable, under some condition. These results are classical; see for instance [Bro76, Kůr04] in the case of discrete-time dynamical systems and [EEN00] for the general case. We also propose a function that quantifies almost periodicity and investigate basic properties of this function.

3.2.1 Minimal dynamical systems

Let us introduce *finite-to-one* dynamical systems, i.e., systems for which every action of the monoid is a finite-to-one transformation. Turing machines, subshifts and tilings are finite-to-one systems (the latter are even one-to-one systems). We prove that minimal finite-to-one dynamical systems are either finite or uncountably infinite. Recall that an isolated point is an open singleton, and a perfect space is a space without isolated point.

Proposition 4 *A minimal finite-to-one dynamical system is finite if and only if it has an isolated point. Infinite minimal finite-to-one dynamical systems are perfect and uncountable.*

Proof. Let (Y, G) be a minimal dynamical system. If $x \in Y$ is isolated then $\{x\}$ is an open set and, from minimality, $\bigcup_{g\in G} g^{-1}(x) = Y$ (otherwise the set $Y \setminus \bigcup_{g\in G} g^{-1}(x)$ of points that cannot reach $\{x\}$ would be a nonempty invariant closed subset of Y). From compactness, a finite cover may be extracted. But each set $g^{-1}(x)$ of the cover is finite, hence Y must be finite. Conversely, if Y is finite, then all points are obviously isolated.

It is known that any perfect compact metric space is uncountable (as a corollary of Baire category theorem; see Section 2.1). We thus infer that a minimal finite-to-one dynamical system is either finite or uncountable. □

Recall from Section 2.3 that a finite minimal dynamical is said to be periodic. Note that if we do not ask the system to be finite-to-one, the proposition above is no longer valid, as shown by the following counterexample.

Example 1 *Let Y be the set $\{1, 1/2, 1/4, 1/8, \ldots, 0\}$, f be the transformation $x \mapsto 1$, g be the transformation $x \mapsto x/2$ and G the monoid generated by f and g. Notice that the system is not finite-to-one because f is not, and (Y, G) is an infinite minimal system with isolated points.*

3.2.2 Almost periodic configurations

A configuration x of a dynamical system (X, G) is said to be *almost periodic*, or *uniformly recurrent*, if for every neighborhood U of x there is a finite subset $H \subseteq G$ such that for every y in the orbit of x, the set Hy of all images of y under the actions of the elements of H, has some configuration in U. In other words, a configuration x is almost periodic if one can choose a finite number of transformations that send every point of the orbit near x.

Notice in particular that periodic configurations are almost periodic. Indeed, the orbit of a periodic configuration x is a minimal system, and from Proposition 1, any point of the orbit can be brought to x under the action of some element of the monoid. As the orbit is finite, finitely many such elements are enough to bring any configuration of the orbit to x.

We can easily find a combinatorial characterization of almost periodicity for our particular examples of dynamical systems. For this we need the notion of pattern. For tilings on the set of tiles T, a *pattern* [Dur97] is a partial function $\mathbb{Z}^2 \to T$ of finite domain. For an infinite word a *pattern* is a finite subword. For a Turing machine, a *pattern* is the conjunction of a finite set of constraints on the content of cells (i.e., a finite partial function $\mathbb{Z} \to A$ for each tape, where A is the tape alphabet) and/or on the state. To every pattern we can associate a clopen sets in a natural way: the set of configurations extending some pattern is a clopen set. Clopen sets are studied in Section 2.6. The following proposition is a direct translation of the general definition of almost periodicity in terms of patterns.

Proposition 5 *An infinite word w is almost periodic if and only if for every pattern u of w there is an integer k such that the pattern u appears in every pattern of length k.*

A configuration of $T^{\mathbb{Z}^2}$ (for a finite set T) is almost periodic if and only if for every pattern u of the configuration there is an integer k such that u appears in every $k \times k$ square pattern.

A configuration of a Turing machine is almost periodic if and only if for every pattern u occurring in the configuration there is an integer k such that the pattern u occurs infinitely often, and the time between two successive occurrences is at most k.

The next proposition relates minimal subsystems and almost periodicity.

Proposition 6 *Let x be a configuration of a dynamical system . The following statements are equivalent:*

1. *x is almost periodic;*
2. *the closed orbit of x is minimal;*
3. *the preimages of any neighborhood of x cover of the closed orbit of x.*

Proof. (1)⇒(2) Let x be an almost periodic configuration of (X, G) and y be in the closed orbit of x. Then for any closed neighborhood U of x there is a finite subset H of G such that $\bigcup_{h\in H} h^{-1}(U)$ covers the orbit of x. But this set, finite union of preimages of closed sets, is closed (recall from Section 2.1). It therefore also covers the closed orbit. Since this applies for any closed neighborhood U, we deduce that x is in the closed orbit of y. Thus the closed orbits of x and y are the same. As it is true for any y in the closed orbit of x, we find from Proposition 1 that the closed orbit of x is minimal.

(2)⇒(3) For any nonempty open set U of a dynamical system (X,G), the set $X \setminus \bigcup_{g\in G} g^{-1}(U)$ of all points that cannot fall into U is invariant and closed. Thus it must be empty if the system is minimal, and the preimages of U cover the space. If U is an arbitrary neighborhood of x, the interior of U is a nonempty open set and the same conclusion applies: the preimages of U cover the closed orbit of x.

(3)⇒(1) If the preimages of an open neighborhood of x cover the closed orbit of x, then from compactness, a finite number of preimages already cover the closed orbit. As it is true for any open neighborhood of x, thus for every neighborhood of x, x is almost periodic. □

By Zorn's lemma and compactness, in every dynamical system there exists a minimal subsystem, as detailed in Section 2.3. According to the above proposition, minimal subsystems exactly correspond to closed orbits of almost periodic configurations, and so we have the following well-known corollary, due to Birkhoff.

Proposition 7 *Every nonempty dynamical system has an almost periodic configuration.*

From Proposition 4 we also get the following.

Proposition 8 *If a finite-to-one dynamical system has an almost periodic configuration that is nonperiodic, then it has uncountably many such configurations.*

We can apply these results to tilings, infinite words and Turing machines:

Proposition 9 *Every tile set that can tile the plane can tile it in an almost periodic way. If a tile set can generate a nonperiodic almost periodic tiling, then it generates uncountably many of such tilings.*

Every Turing machine that has an immortal configuration has an almost periodic immortal configuration. If a Turing machine has a nonperiodic almost

periodic immortal configuration then it has uncountably many of such configurations.

In every subshift there is an almost periodic word.

The result for infinite words, as well as the characterization of Proposition 5, have been known for long; see [MH36, Kůr04] for instance. The results for tilings are proved by Durand [Dur97], where the uncountability part is established using combinatorial arguments. Our topological proof has the advantage that it is applicable to any dynamical system. The result for Turing machines appeared in [DB04b] for the first time. It nicely complements the result proved in [BCN02] that Turing machines do not always have periodic configurations.

It is interesting to note that, contrarily to tilings and Turing machines, subshifts of finite type always have a periodic configuration. Indeed, suppose that all forbidden patterns have length less than n. Any configuration of the subshift contains a word of the form uvu, where u has length n. Then the periodic configuration $(uv)^\omega$ is in the subshift as well, because it does not contain any forbidden word.

This suggests that by some aspects at least, the dynamics of tilings is closer to that of Turing machines than to that of infinite words, even though tilings can be thought of as subshifts of finite type in two dimensions.

3.3 Almost-periodicity functions

To study almost periodic configurations in greater length, a special function is defined for tilings in [Dur97] and for words in [MH36, Cas01] (the function is called *recurrence function* in the latter reference). In both cases, these functions quantify how far an almost periodic configuration is from periodicity. In this subsection we introduce an almost-periodicity function in the context of dynamical systems and prove that our definition essentially coincides with those for tilings and infinite words.

The *almost-periodicity function* $Q_T : \mathbb{N} \to \mathbb{N}$ of an almost periodic tiling of the plane is the function that to n associates the smallest k such that every $(2k+1) \times (2k+1)$ pattern of the configuration contains all $(2n+1) \times (2n+1)$ patterns observable on the configuration. Note that this is not exactly the function introduced in [Dur97]; we restrict ourselves to pattern of odd size, in order to simplify the presentation. We can prove [Dur97] that an almost periodic configuration is periodic if and only if its almost periodicity function is bounded by $n \mapsto n + c$ for some constant c. The 'only if part' is easy. The converse is proved along the following line. Take larger and larger patterns centered at $(0, 0)$. Each pattern can be shifted by a vector in $\mathbb{Z}^2$ with both components between 1 and $2c$, such that the pattern and the shifted pattern agree on their common domain. Infinitely many patterns must have the same

vector, hence the whole tiling is invariant under one shift map. We repeat the argument to find another shift map with an independent translation vector. A tiling which is periodic along two independent directions is periodic.

Moreover, Cervelle and Durand [CD00] prove that for any increasing time-constructible function $f : \mathbb{N} \to \mathbb{N}$, there is a tile set such that all tilings generated are almost periodic and have f for almost-periodicity function. A function $n \mapsto f(n)$ is *time-constructible* if there is a Turing machine whose halting time on data n is defined and is exactly $f(n)$. The idea of the proof is to simulate this Turing machine in the way provided by Berger's proof. By simulating a Turing machine that enumerates a nonrecursive r.e. set, we can construct [CD00] a tile set whose almost periodic tilings have an almost periodic function that grows faster than any total recursive function.

A function that quantifies almost-periodicity is also defined for infinite words; see [Cas01, MH36]. The *recurrence function* $Q_W : \mathbb{N} \to \mathbb{N}$ of an almost periodic infinite word of $A^{\mathbb{N}}$ is the function that to n associates the smallest k such that every subword of length k contains all subwords of length n. Some properties of this function are derived in [Cas01]. In particular, it is shown that, as for tilings, an almost periodic configuration is periodic if and only if its recurrence function is bounded by $n \mapsto n + c$ for some constant c.

We now define almost-periodicity functions for arbitrary dynamical systems. Let (Y, G) be a minimal subsystem of the dynamical system (X, G). The *almost-periodicity function* of Y relatively to (X, G) is the function $Q : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$ defined by $\epsilon \mapsto \inf_{y \in Y}\{\lambda_{y,\epsilon}\}$, where

$$\lambda_{y,\epsilon} = \sup\{\lambda > 0 | \forall x \in Y \, \exists g \in G : B(x, \lambda) \subseteq g^{-1}(B(y, \epsilon))\}.$$

Notice that, thanks to Proposition 6, for every $y \in Y$ the collection of open sets $\{g^{-1}(B(y, \epsilon)) : g \in G\}$ is an open cover of Y, and $\lambda_{y,\epsilon}$ is a Lebesgue number of this cover; see Section 2.1 for a definition of Lebesgue number. This function depends on the metric of the space, not just on the topology.

Proposition 10 *When specialized to tilings and words, the almost-periodicity function defined above satisfies* $Q(2^{-n}) = 2^{-Q_T(n)}$ *and* $Q(2^{-n}) = 2^{-Q_W(n)}$.

Proof. We only prove $Q(2^{-n}) = 2^{-Q_T(n)}$; the proof for infinite words is similar. Let T be a finite set of tiles and suppose that $u \in T^{\mathbb{Z}^2}$ is an almost periodic configuration whose almost-periodicity function relatively to $T^{\mathbb{Z}^2}$ is $\epsilon \mapsto Q(\epsilon)$ and fix a square pattern of size $2n + 1$ present in u. If $k = Q_T(n)$ then this pattern is extended by every $(2k + 1) \times (2k + 1)$ square pattern of u. It is also extended by every $(2k + 1) \times (2k + 1)$ square pattern of every configuration in the closed orbit of u, because every such configuration is the limit of elements of the orbit of u.

This exactly means that for every point of the closed orbit of u, the open ball of radius 2^{-k} around this configuration is included in a preimage (by some

shift) of the open ball of radius 2^{-n} around some (fixed) configuration of the orbit of u. Indeed recall from Section 2.6 that the set of configurations extending a $(2k+1) \times (2k+1)$ square pattern is open ball of radius 2^{-k} in $T^{\mathbb{Z}^2}$.

Thus $Q_T(n) = k$ means that for every configurations v, w of the closed orbit of u, $B(w, 2^{-k})$ is included in the preimage by some shift of $B(v, 2^{-n})$, and that k is the smallest integer to have this property. But this also means that $Q(2^{-n}) = 2^{-k}$. □

It would be interesting to see to what extent properties of the almost-periodicity function for tilings and infinite words may be generalized to arbitrary dynamical systems. For instance, remark that for tilings and infinite words, the almost-periodicity function Q_T or Q_W is greater than the identity. Formulated in our context this would suggest that almost-periodicity functions are bounded above by the identity function. However, this turns out to be false in general, as shown by the following example. Let

$$f : [-1, 1] \to [-1, 1] : x \mapsto \frac{x}{2}.$$

The only almost periodic point of this dynamical system is the origin, and the corresponding almost periodicity function is identically equal to infinity.

More generally, we can exactly characterize the dynamical systems that have an almost periodic configuration with the infinite almost-periodicity function. Let us say that a subsystem Y of a dynamical system (X, G) is *globally attracting* if for every open neighborhood U of Y, there is a $g \in G$ such that $Gg(X) \subseteq U$ (once g is applied the remaining part of the orbit remains in U). In the map f on $[-1, 1]$ defined above, 0 is globally attracting.

Let us call a monoid *weakly commutative* if $\forall g \in G : gG = Gg$. In particular, commutative monoids and all groups are weakly commutative. If the monoid is weakly commutative, a subsystem Y is globally attracting iff Y contains the limit set $\cap_{g \in G} g(X)$ (as defined in Section 2.3). Indeed, if Y is globally attracting then for any open neighborhood U of Y there is a $g \in G$ such that $g(X) \subseteq U$; so the limit set in included in U. As U is an arbitrary open neighborhood of Y, the limit set is included in Y. Conversely, if Y contains the limit set then for any open neighborhood U of Y, there exist $g_1, \dots, g_n \in G$ such that $g_1(X) \cap \dots \cap g_n(X) \subseteq U$ (from compactness). From weak commutativity, for any i and every $h \in G$ there is an $h' \in G$ such that $hg_1 \dots g_n(X) = g_i h'(X) \subseteq g_i(X)$. Thus $hg_1 \dots g_n(X) \subseteq \cap_i g_i(X) \subseteq U$. Hence for $g = g_1 \dots g_n$, we have $Gg(X) \subseteq U$, and Y is globally attracting.

Proposition 11 *A minimal subsystem of a dynamical system with a weakly commutative monoid has its almost-periodicity function identically equal to infinity if and only if it is a globally attracting point.*

Proof. Suppose x is a globally attracting point of (X, G), and choose an open

neighborhood U of x. Then there is a $g \in G$ such that $g(X) \subseteq U$, i.e., $g^{-1}(U) = X$. Then $B(x, +\infty) = X$ is included in some preimage of U.

Conversely, if a minimal subsystem Y of (X, G) has an almost-periodicity function $\epsilon \mapsto \infty$, then for every $y \in Y$ and every $\epsilon > 0$ there is a $g \in G$ such that $g^{-1}(B(y, \epsilon)) = X$. Thus if there are distinct $y_1, y_2 \in Y$ then for any ϵ, there are g_1 and g_2 in G such that for all $x \in X$, $Gg_1(x) \subseteq B(y_1, \epsilon)$ and $Gg_2(x) \subseteq B(y_2, \epsilon)$. From weak commutativity $g_1 g_2(x)$, equal to $g_2' g_1(x)$ for some g_2', is in $B(y_1, \epsilon)$. But it is also in $B(y_2, \epsilon)$, which is impossible if ϵ is small enough. Hence Y is a singleton. This singleton is globally attracting, since $g(X) \subseteq U$ implies $Gg(X) = gG(X) \subseteq U$. □

In the proof of the result, we need to assume that the monoid is weakly commutative. We do not know if the statement remains valid when this assumption is removed.

The existence of a globally attracting point may thus be read in the almost-periodicity functions of the system. We do not know what other properties of systems may be characterized in terms of their almost-periodicity functions. Also, we do not know what conditions should be put on a system to ensure that the almost-periodicity functions of minimal subsystems are bounded above by identity.

3.4 Undecidable properties

In this section we give sufficient conditions for dynamical properties of dynamical systems to be undecidable, and illustrate the interest of these conditions by deriving new undecidability results for questions related to the topological entropy and the number of invariant subspaces of Turing machines and tilings.

3.4.1 Families of dynamical systems

A *family of dynamical systems* is a set $(Y_n, G_n)_{n \in \mathbb{N}}$ of dynamical systems, indexed by the natural integers. A system (Y, G) for which Y is empty is said to be an *empty system*. In the sequel we assume that the families we consider always contain an empty system. Examples of families of dynamical systems are given by tilings, subshifts of finite type and Turing machines restricted to immortal configurations, as described in Section 3.1.

We say that a dynamical system property is *decidable* if for every given n, we can algorithmically decide whether or not (Y_n, G_n) satisfies the property. One of the most basic property we may want to test is whether the system is empty. This problem is undecidable for tilings [Ber66] and for Turing machines (even with only one tape) [Hoo66]: there is no algorithm to decide whether the system of valid tilings built on a given tile set is empty, and there is no algorithm to

decide whether the system of immortal configurations of a given Turing machine is empty. By contrast, the emptiness problem for subshifts of finite type is decidable, because if it is nonempty we can prove it by exhibiting a periodic infinite word, and if it is empty we can prove that all long enough words contain a forbidden pattern. Motivated by this common feature of Turing machines and tilings, we investigate what properties are undecidable for dynamical systems.

This point of view is to be compared to Chapter 4 where we study the decidability of properties of a single system, instead of a family.

3.4.2 A Rice-style theorem for dynamical systems

Not all properties of all dynamical systems are undecidable. In order to derive a general undecidability result we therefore need to impose conditions on the families and the properties that we consider.

Firstly, we will consider dynamical systems that have an undecidable emptiness problem and for which it is possible to effectively compute cartesian products (tilings and Turing machines restricted to immortal configurations satisfy these conditions, see below). Secondly, we will consider properties that are invariant under conjugacies and that are not affected by cartesian product. The former condition, invariance under conjugacies, is quite natural; we do not know if our theorem remains valid if we remove the second condition.

Let us now formalize all this. We need several definitions. A system property that is invariant under conjugacy is said to be a *dynamical property*. Conjugacy between dynamical systems was defined in Section 2.3.

The (cartesian) *product* of the two dynamical systems (X, G) and (Y, G) is the system $(X \times Y, G)$. Notice that products of systems are defined only for systems with the same monoids and that the product of two systems is always empty when one of the systems is. Products of systems may or may not be computable. We say that *products are computable* for a family of dynamical systems if given two indices i, j, an index k_{ij} can be effectively computed such that the system of index k_{ij} is conjugated to the product of the systems i and j. Products are computable for tilings as well as for Turing machines. Indeed, to make the product of two sets of tilings, just make the product of the corresponding sets of tiles: this is an effective operation. To make the product of two Turing machines, just make the disjoint union of the corresponding tapes, and the product of the internal states of the heads.

Finally, we say that a dynamical property is *not affected by product* if the product of any system that has the property with a nonempty system is a system that has the property. Examples of such properties are given in Corollaries 1 and 2. We are now ready to state our result.

Proposition 12 *Consider a family of dynamical systems for which products are computable and emptiness is undecidable. Then, any dynamical property*

that is not affected by product and that is satisfied by at least one member of the family, but that is not satisfied by the empty system, is undecidable.

Proof. The proof proceeds by reduction to the emptiness problem and is as simple as the proof of Rice's theorem. Let $(S_n)_{n\in\mathbb{N}}$ be a family satisfying the conditions of the proposition. Suppose we have a suitable property P, that is verified by the system S_i, and assume there is an algorithm that decides that property. Then the following algorithm decides emptiness of the system S_n:

```
Input n;
Compute an index for S_i × S_n;
If S_i × S_n has property P
 then print 'S_n is not empty'
 else print 'S_n is empty'.
```

□

We now apply this result to tilings and multi-tape Turing machines.

Corollary 1 *The following questions about tilings and Turing machines restricted to immortal configurations are undecidable: Given a system of the family,*

1. *are there at least two disjoint invariant subspaces?*
2. *is there an infinite number of pairwise disjoint invariant subspaces?*

For any fixed nonempty system S of the family, the problem of determining, for a given system, if S is a factor of the system, is undecidable.

Factors of a system are defined in Section 2.3.

Proof. Tilings and Turing machines restricted to immortal configurations have computable cartesian products, and their emptiness problem is undecidable. The latter fact was proved by Berger [Ber66] and Hooper [Hoo66] respectively.

Properties (1) and (2) easily verify conditions of Proposition 12. In particular, a Turing machine shifting the tapes to the left, and a set of tilings conjugated to $2^{\mathbb{Z}^2}$ have both properties.

For a proof of the second statement, remark that S is not a factor of $\emptyset$, but it is a factor of S and if it is a factor of S' then it is a factor of any product of S' by a nonempty system. □

Note that the presence of two disjoint invariant subsets in a set of tilings exactly means that we can find two tilings of the plane and an integer N such that they have no pattern of size $N \times N$ in common.

3.4.3 Topological entropy

We apply Proposition 12 to derive results on the topological entropy of dynamical systems. Entropy was proved to be uncomputable for cellular automata [HKČ92] and for piecewise affine maps [Koi01], we show that it is not computable for Turing machines restricted to immortal configurations and for tilings. Let us first recall some definitions and basic properties (the material presented here is essentially taken from [Jen01, Bro76, Kůr04]).

The *topological entropy* of a subshift is given by

$$\lim_{n\to\infty} \frac{\log_2 N_n}{n},$$

where N_n is the number of patterns (subwords) of length n in the subshift. Using the fact that $N_{n+m} \leq N_n N_m$, we can show that the limit exists; see Section 2.10 in [Kůr04], for instance. Given the system of immortal configurations of a Turing machine, every clopen partition induces a subshift. The *topological entropy* of the system is the supremum of the topological entropy of induced subshifts. This definition makes sense for any symbolic discrete-time system (as defined in Section 2.6) and can be extended to any discrete-time system, but we only need it for Turing machines restricted to immortal configurations.

Now consider the set of tilings generated by some tile set. Call N_n the number of $n \times n$ patterns that may be found in at least one of the tilings. Then the *topological entropy* of the tile set is given by

$$\lim_{n\to\infty} \frac{\log_2 N_n}{n^2},$$

which always exists.

These two definitions of entropy can be generalized to any system with a finitely generated monoid (i.e., a monoid containing a finite set that generates all the monoid by making all possible products): see [Biś04], for instance.

One can prove in a straightforward manner that in both cases the entropy of the product of two nonempty systems is the sum of the individual entropies, and the entropy of the sum of two nonempty systems is the maximum of the individual entropies. We define the entropy of the empty system to be equal to zero.

Corollary 2 *For any fixed $\alpha > 0$, the following questions about tilings and Turing machines restricted to immortal configurations are undecidable: Given a system of the family,*

1. *Is the topological entropy of the system greater than or equal to α?*
2. *Is the topological entropy of the system greater than α?*

Proof. The topological entropy of the product of two systems is the sum of the individual entropies, and is consequently greater than or equal to the individual entropies. Moreover, for any $\alpha \geq 0$, it is easy to build a tiling or a Turing machine, whose entropy is greater than α (just think to the Turing machine that does nothing but shifts $\lceil \alpha \rceil$ tapes to the left, with an alphabet of size two or more). From Proposition 12, this ensures the undecidability of both questions.□

The reader might feel skeptical about the entropy argument: since entropy of the empty set is actually not defined except by convention, we could not blame an algorithm that correctly decides whether the entropy is greater than α *except for the empty set*, for which it would not halt. But we can easily modify the proof of Proposition 12 in a more satisfactory way: instead of testing the entropy of $S_i \times S_n$ we test $S_i \times S_n + S_k$, where '+' is the disjoint union of dynamical systems, defined similarly to cartesian product, and S_k is a system of the family with entropy zero. For tilings, we can make the disjoint sum of tilings sets by making the disjoint sum of tile sets: this is an effective operation. For two Turing machines with the same number of tapes and the same alphabets, we can make the disjoint sum of immortal systems by making the disjoint sum of internal states: this is an effective operation. Thus in the latter case, we choose for S_k a Turing machine that has the same number of tapes and the same alphabets as $S_i \times S_n$, and whose entropy is zero.

Let us mention a consequence of the uncomputability of entropy for tilings on data storage. One can see a configuration of the plane as the encoding of some data on a two-dimensional surface. If we impose that the encoding must avoid certain patterns (because, for instance, they would be misinterpreted by the lecture head) then the encoding results in a (piece of) tiling. In this case, the topological entropy is interpreted as the average 'capacity' of the storage, i.e., the number of bits that can be encoded in one cell. This approach is undertaken for instance in [HCR+04] (see also references within). Corollary 2 says that the capacity of a two-dimensional data storage cannot be computed in general, but only in particular cases.

3.5 Conclusions

In this chapter we introduce a common method to formulate and solve questions about various objects such as tilings, Turing machines and infinite words. The method is to study them as dynamical systems rather than combinatorial objects.

We give a general result about the existence and cardinality of almost periodic points for dynamical systems. Then we introduce an almost-periodicity function for dynamical systems that generalizes similar functions introduced

in the literature for the specific case of tilings and infinite words. Finally, we study decidability of dynamical systems properties. A number of properties are shown to be undecidable for tilings and Turing machines. In particular topological entropy is shown to be uncomputable.

Many questions remain open. As explained in Section 3.3, the form of the almost periodicity functions of a dynamical system can be connected to the properties of the system. For example, what kinds of systems verify the properties of this function mentioned from [Cas01, CD00, Dur97]? Also, it would desirable to characterize more completely what properties of dynamical systems are undecidable. A natural improvement would be to lift or weaken the hypothesis made in Proposition 12 that the property to be proved undecidable is not affected by product. Along this way, an ambitious goal would be to find general conditions for emptiness of a system to be undecidable. This could lead in particular to a unified proof of Berger's and Hooper's theorems, stating the undecidability of the existence of a tiling and an immortal configuration of a Turing machine respectively.

Another direction of research would be to study which of our results apply to a broader set of dynamical systems, such as cellular automata, piecewise affine maps of $\mathbb{R}^n$ and ordinary differential equations.

4

Decidability and Universality in Symbolic Discrete-time Dynamical Systems

4.1 Introduction

The basic aim of this chapter is to define and characterize computational universality in dynamical systems. We propose an approach for the case of symbolic discrete-time dynamical systems. A presentation of the motivations as well as pointers to the existing literature are provided in Section 1.2.

We recall that a symbolic discrete-time dynamical system is a continuous transformation of a symbolic space, which can be seen as a closed subset of the Cantor space $\{0,1\}^{\mathbb{N}}$ endowed with the product topology; see Section 2.6. Symbolic systems are a particular case of the dynamical systems considered in Chapter 3. The main motivating examples are subshifts, Turing machines (with or without blank symbol) and cellular automata. All these objects are defined in Chapter 2.

In Sections 4.2, 4.3 we enrich symbolic systems with a structure of effectiveness. Clopen sets are assumed to be numbered by $\mathbb{N}$, in such a way that the boolean operations ($\cap$, $\cup$, $.^c$) and the preimage ($f^{-1}(.)$, if f is the map) are computable.

If a clopen partition is given on a symbolic system, the system can be observed by a Büchi automaton (defined in Section 2.5). A system is said to be 'decidable' if given the partition and the Büchi automaton, one can decide whether there is an initial configuration of the system that induces an infinite word accepted by the automaton. In other words, a system is decidable if the emptiness of a given induced subshift with a given ω-regular language is decidable. Some elements of the theory of languages are reviewed in Section 2.5.

If we observe the symbolic system with a finite automaton (also defined in Section 2.5), we can ask whether the language induced by the partition has an intersection with the regular language defined by the finite automaton. If this problem is r.e.-complete, we call the system '(computationally) universal'.

Following Turing's classic argumentation, we can interpret the finite automaton as a human observer with finitely many possible mental states, taking advantage of his/her observations on the symbolic system to perform computations; as the observer cannot distinguish configurations that are too close, the space is partitioned into clopen sets of undistinguishable points. Decidability and universality are discussed in Section 4.5 and a discussion of our definition of universality with respect to other definitions proposed in the literature is the subject of Section 4.8.

What are the dynamical properties of a universal symbolic system? In Section 4.6, necessary conditions for a system to be universal are given, related to minimality, equicontinuity and the shadowing property. For instance, minimal systems are proved to be decidable, hence a universal system has at least one proper nonempty subsystem. In Section 4.7 we build a chaotic system that is computationally universal. As conclusions, some ideas for future work are sketched in Section 4.9.

4.2 Effective symbolic spaces

To define computational universality, we need effective symbolic spaces, in which we can perform boolean combinations on clopen sets effectively.

An *effective symbolic space* is formally a pair (X, P), where X is a symbolic space and $P : \mathbb{N} \to 2^X$ is an injective function whose range is the set of all clopen sets of X, such that the intersection and complementation of clopen sets are computable operations. This means that there exist total computable functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that $P_n^c = P_{f(n)}$ and $P_n \cap P_m = P_{g(n,m)}$.

Of course, union of clopen sets is then also computable. When no confusion is possible, we allow ourselves to denote an effective symbolic space by X rather than (X, P). In the Cantor space $\{0,1\}^{\mathbb{N}}$, the lexicographic ordering yields a standard enumeration

$$\emptyset, [], [0], [1], [00], [01], [10], [11], [00]\cup[11], [01]\cup[10], [00]\cup[01]\cup[10], [00]\cup[01]\cup[11], \ldots$$

Other widely used symbolic spaces like $\{0,1\}^* \cup \{0,1\}^{\mathbb{N}}$, $A^{\mathbb{N}^d}$, $A^{\mathbb{Z}^d}$, $Q \times A^{\mathbb{Z}}$, have also their standard enumerations. Note that we could require intersections and complements to be primitive recursive rather than computable, without altering the examples and results of the chapter.

Let (X, P) and (Y, Q) be two effective symbolic spaces. An *effective continuous map* is a continuous map $h : X \to Y$ such that $h^{-1}(Q_n) = P_{k(n)}$, for some computable map $k : \mathbb{N} \to \mathbb{N}$. If h is bijective then it is an *effective homeomorphism*, and (X, P) is said to be *effectively homeomorphic* to (Y, Q).

Note that the composition of effective continuous maps is an effective continuous map, the identity is an effective continuous map and the inverse map of

an effective homeomorphism is also an effective homeomorphism. In particular, being effectively homeomorphic is an equivalence relation for effective symbolic spaces.

Given an effective symbolic space (X, P), a closed subset Y is said to be *effective*, if the family of clopen sets intersecting Y is decidable. In particular any clopen set is effective. An effective closed subset Y can be endowed with the subset topology, whose clopen sets are all intersections of clopen sets of X with Y. Thus, the enumeration $P_0, P_1, P_2, \ldots$ of clopen sets of X yields an enumeration of clopen sets of Y: $Y \cap P_0$, $Y \cap P_1$, $Y \cap P_2$, ... This enumeration may contain empty sets and repetitions, but we can detect them in an effective way and renumber the sequence accordingly. Hence we get an effective topology for the effective closed set Y. Equivalently, the inclusion $i : Y \hookrightarrow X$ is an effective continuous map. Note that the image of an effective closed set by an effective continuous map is effective closed.

The following is a straightforward generalization of Proposition 3.

Proposition 13 *Every effective symbolic space is effectively homeomorphic to an effective closed subset of the Cantor space. Every perfect effective symbolic space is effectively homeomorphic to the Cantor space.*

Proof. Let (X, P) be an effective symbolic space. The injective map $g : X \to \{0,1\}^{\mathbb{N}}$ constructed in the proof of Proposition 3 is effective, since every step of the construction is effective. The range of an effective map is an effective closed set. If X is perfect, then the homeomorphism $h : X \to \{0,1\}^{\mathbb{N}}$ is effective as well. □

We see that there is no loss of generality in supposing that in any effective symbolic space, for any rational ϵ there exists a finite number of balls of radius ϵ and that we can compute all of them. Indeed, this is the case for all effective subsets of the Cantor space.

4.3 Effective symbolic systems

An *effective symbolic discrete-time dynamical system* is an effective continuous map from an effective symbolic space to itself. In other words, an effective symbolic system is a symbolic space with a continuous self-map in which intersections, complements, and inverse images of clopen sets are computable. This definition of effective function in a Cantor space is equivalent to classical definitions in computable analysis; see for instance [Wei00]. We denote an effective symbolic system by a map $f : X \to X$ or simply f, when the enumeration P of X is implicit. Extending the definition of effective homeomorphism in the preceding section, we define a relation of equivalence between effective systems.

The effective symbolic systems $f : X \to X$ and $g : Y \to Y$ are *effectively conjugated* if there exists an effective homeomorphism $h : X \to Y$ such that $h \circ f = g \circ h$. If $h : X \to Y$ is an effective surjective map (and not necessarily bijective), then the system $g : Y \to Y$ is said to be an *effective factor* of $f : X \to X$. The factor g can be seen as a 'simplification' of f.

An *effective subsystem* of an effective symbolic system is an effective closed subset that is invariant under the map. With the subset topology, it is itself an effective symbolic discrete-time system.

The identity on any symbolic space is the simplest example of an effective symbolic system. A cellular automaton is an effective symbolic system on the space $A^{\mathbb{Z}^d}$, where A is the finite alphabet and d is the dimension. Turing machines, with or without blank symbol, are also effective symbolic discrete-time dynamical systems (see Section 2.7 for details about Turing machines interpreted as dynamical systems).

4.3.1 Shifts and subshifts

A (one-sided or two-sided) shift is an effective symbolic system. A subshift that is an effective closed subset of $A^{\mathbb{N}}$ (for some alphabet A) is again an effective symbolic discrete-time system. All subshifts we consider in this chapter are one-sided subshifts. It is easy to see that a subshift is effective if and only if its language is decidable. In particular every sofic subshift is effective, since a regular language is decidable.

Recall from Section 2.6 that any clopen partition of a symbolic discrete-time system induces a factor subshift. If the system is effective than the subshift is effective. We can actually characterize effective discrete-time symbolic systems in terms of their induced subshifts.

Proposition 14 *A symbolic discrete-time dynamical system is effective if and only if there is an algorithm deciding whether a given word belongs to the language of the subshift induced by a given partition.*

Proof. Let $\mathcal{A} = \{A_1, \ldots, A_N\}$ be a clopen partition. Then a word $a_0 a_1 \ldots a_{l-1} \in \mathcal{A}^*$ is in the language of the subshift induced by the partition if and only if $a_0 \cap f^{-1}(a_1) \cap \cdots \cap f^{-(l-1)}(a_{l-1})$ is not empty. This can be checked algorithmically.

Conversely, suppose that all induced subshifts have decidable languages, and that given the partition we can effectively find a decision algorithm for the corresponding language. Let P_n be a clopen set of X. There exists a clopen partition $\mathcal{A} = \{A_1, \ldots, A_N\}$ such that

- for every i, either $A_i \subseteq P_n$ or $A_i \subseteq P_n^c$;
- if $A_i A_j$ and $A_i A_k$ belong to the language of the induced subshift, then A_j and A_k are either both parts of P_n or both parts of P_n^c.

The first condition says that the partition is finer then $\{P_n, P_n^c\}$, the second condition says that the partition is finer than $\{f^{-1}(P_n), f^{-1}(P_n^c)\}$. It can be checked algorithmically whether a clopen partition has these two properties. Thus a partition with these properties can be found algorithmically. Then we can compute $f^{-1}(P_n)$ as the union of all A_i such that there exists a word A_iA_j in the language of the induced subshift and that $A_j \subseteq P_n$. □

If the subshifts have decidable languages, but decision algorithms are not computable with respect to the clopen partition, then the system may fail to be effective. This happens in the following example.

Example 2 *Assume $k : \mathbb{N} \to \mathbb{N}$ is a noncomputable strictly increasing total function. We define a function f on the Cantor space $\{0,1\}^{\mathbb{N}}$ by $f(x) = f_0(x)f_1(x)f_2(x)\ldots$, where $f_i(x_0x_1x_2\ldots) = \max\{x_0, x_1, x_2, \ldots, x_{k(i)}\}$. There are two fixed points, 0^ω and 1^ω, and the image of a point is of the form 0^*1^ω. Then it is easy to see that for any point x either $f(x) = 0^\omega$ or $f^n(x) = 1^\omega$ for some $n \geq 0$. For any partition $\mathcal{A} = \{A_1, \ldots, A_N\}$, if $0^\omega \in A_1$ and $1^\omega \in A_2$ (say), then every point in $A_2 \cup \cdots \cup A_N$ reaches A_2 in bounded time (not depending on the point) and does not leave it any more. Then the language of the subshift induced by the partition is of the form $A_1^*SA_2^*$, where S is some finite language. This is certainly a decidable language. However f is not effective, for otherwise we could compute k.*

In the rest of the chapter, we use the terms 'symbolic system' or even 'system' to denote an effective symbolic discrete-time dynamical system.

4.3.2 Products

Let $(f_n : X_n \to X_n)_{n\in\mathbb{N}}$ be a family of *uniformly effective* systems on the effective symbolic spaces (X_n, P_n); we mean that there exists an algorithm that, given n and two clopen sets of X_n, can compute their intersection, complements and preimages. Then the *effective product* of $(f_n)_{n\in\mathbb{N}}$ is the system $f : X \to X$ on the effective symbolic space (X, P) such that

- the set X is the cartesian product of all sets X_n;
- the clopen sets of X are all products of clopen sets $\prod_{n\in\mathbb{N}} A_n$ such that only finitely many $A_n \subseteq X_n$ are different from X_n (this is the product topology);
- the clopen sets are indexed by finite sets of integers in a straightforward manner, and f is defined componentwise.

We see that this is indeed an effective symbolic dynamical system. The projections $\pi_n : X \to X_n$ are effective maps as well. Products are useful to build examples of systems with particular properties, as illustrated in the forthcoming Propositions 25 and 30.

4.4 Automata and the halting problem

Consider an effective system $f : X \to X$ and two clopen sets $U, V \subseteq X$. We would like to know if there is a point of U that eventually reaches V, that is, if there exists an x such that

$$x \in U \text{ and } \exists n \in \mathbb{N} : f^n(x) \in V. \tag{4.1}$$

We call *halting problem of* f, the problem of answering this question given U and V. We will see later that this is indeed a generalization of the halting problem traditionally defined for Turing machines or counter machines.

Consider now another formulation of the halting problem. Suppose that the system f is only partially observable. All we know about f is whether the system is currently in U, in V or in $W = (U \cup V)^c$ (we suppose for simplicity that U and V are disjoint). The system is observed by a finite automaton as illustrated in Figure 4.1. At every time step, the automaton jumps to a new state, according to which set U, V or W the system is currently in. The halting problem amounts to deciding whether it is possible, for some initial point of the space X, that the automaton eventually reaches the final state from the initial state.

We would also like to consider variants of the halting problem. For instance, given three disjoint clopen sets U, V and W, we want to check whether the following formula is satisfied for some x:

$$x \in U \text{ and } \exists n : f^n(x) \in V \text{ and } \forall m < n : f^m(x) \notin W, \tag{4.2}$$

where n and m are nonnegative integers. A finite automaton that accepts exactly the points with this property is constructed in Figure 4.2.

We can also ask whether the formula

$$\forall n : f^n(x) \in U \tag{4.3}$$

is satisfied for some $x \in X$. This is the case if and only if the automaton in Figure 4.3, starting from the initial state and observing the system f, reaches infinitely often the final state from the initial state.

In general we are interested in all properties that can be observed by finite automata and by Muller or Büchi automata; see Section 2.5 for the definitions of these terms.

Given a clopen partition $\mathcal{A} = \{A_1, \ldots, A_N\}$ and a finite automaton over $\mathcal{A}$, we would like to know whether there is a nonempty intersection between the language induced by the partition and the regular language accepted by the automaton. In other words, the problem is to know whether there exists a point of the symbolic system whose trajectory, observed through the partition, can make the automaton reach a final state form the initial state. We call this problem the *model-checking problem for finite automata*.

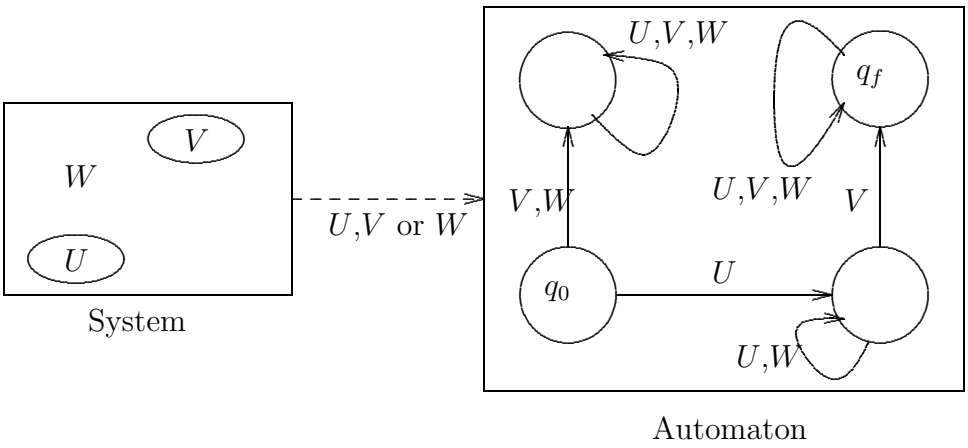

Fig. 4.1. The symbolic space X of the system is partitioned into U, V and $W = (U \cup V)^c$. At every time step, the finite automaton is fed with the symbol U, V or W and jumps to a new state. It is possible to reach the final state q_f from the initial state q_0 iff it is possible that q_f (and only q_f) is reached infinitely often from the initial state q_0 iff there is a point of U that eventually reaches V. Checking whether this is true given U and V, is the *halting problem* of f. The automaton can be considered as a finite automaton (the final state is q_f), as a Büchi automaton (the final state is q_f) or as a Muller automaton (for the family $\{\{q_f\}\}$). In agreement with Kleene's theorem (see Section 2.5), the finite automaton accepts the language $U(V^c)^*V$ and the Büchi or Muller automata accepts the ω-language $U(V^c)^*VX^\omega$.)

The same question can be asked for a Büchi automaton instead of a finite automaton. Given a clopen partition and a Büchi automaton, we want to know whether there is a nonempty intersection between the ω-language induced by the partition and the ω-regular language accepted by the Büchi automaton. In other words, the problem is to know whether there exists a point of the symbolic system whose trajectory, when observed through the partition, is accepted by the Büchi automaton. We call this problem the *model-checking problem for Büchi automata*.

In both cases the automaton may be interpreted as *observing* the system. This formalism includes all three properties described above, including the halting problem. Model-checking aims at finding decision algorithms to check whether the trajectories of a dynamical system satisfy a given property. Systems considered in the literature of model-checking are often nondeterministic and finite or countable, whereas we deal with deterministic systems with a possibly uncountable configuration space.

We mentioned already that Büchi and Muller automata are powerful to express properties on infinite words and are equivalent to several logical formalisms, including the so-called μ-calculus and monadic second-order formulae.

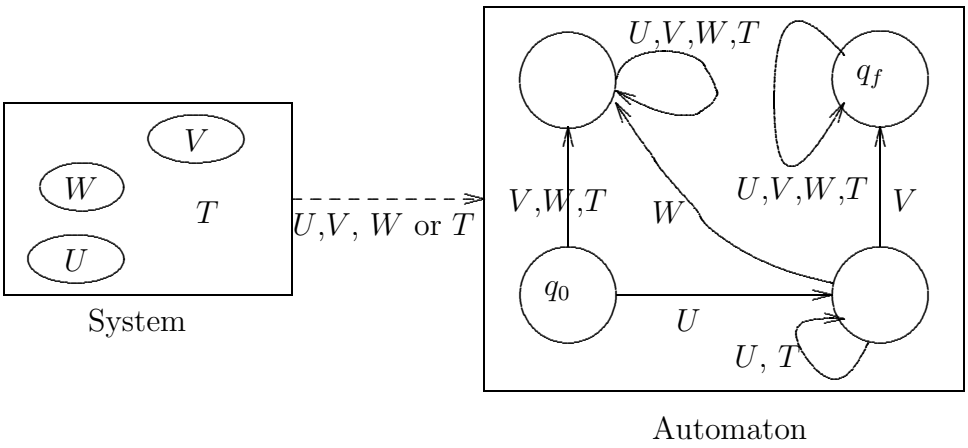

Fig. 4.2. The symbolic system is partitioned into U, V, W and $T = (U \cup V \cup W)^c$. There is a point of U that stays in W^c until it eventually reaches V, iff it is possible that q_f (and only q_f) is reached from the initial state q_0, iff it is possible that q_f (and only q_f) is reached infinitely often from the initial state q_0. The finite automaton accepts the language $U(U \cup V \cup T)^*V$ and, when interpreted as a Büchi or Muller automaton, the ω-language $U(U \cup V \cup T)^*VX^\omega$

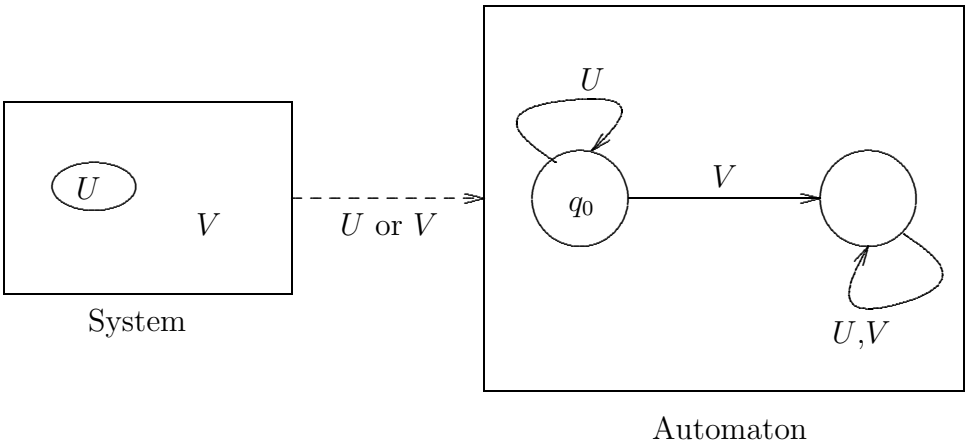

Fig. 4.3. The system is partitioned into U and V. There is a point that never leaves U iff it is possible that q_0 (and only q_0) is reached infinitely often from the initial state q_0. The ω-language accepted is U^ω.

First-order formulae, including (4.1), (4.2), (4.3), are equivalent to the so-called linear temporal logic and strictly weaker than Muller automata. For precise definitions of all these formalisms, see for instance [PP04, Kai96, JGP99].

4.5 Decidable and universal systems

An effective symbolic system is *decidable* if the model-checking problem for Büchi automata associated to the symbolic system is decidable. As we have seen in the preceding section and in Section 2.5, Büchi automata is a simply-defined and well-established framework to express a wide range of properties of a system, such as 'Is there a trajectory that reaches the set A infinitely often', for instance. Büchi automata are nothing else than finite automaton that accept infinite words instead of finite words. Informally, they observe all properties that can be observed with a finite memory of the past. Many interesting properties of trajectories in dynamical systems appear to be of that kind.

Clearly, decidability is preserved by effective conjugacies. The factor of a decidable system is decidable. Indeed, if $g : Y \to Y$ is a factor of $f : X \to X$, then a given partition of Y can be lifted through the factor map to a clopen partition on X. So given a partition of Y and a Büchi automaton, it suffices to solve the problem with the lifted partition on X and the same Büchi automaton.

The identity map on any effective symbolic space is decidable. Indeed, for a partition $\{A_1, A_2, \ldots, A_N\}$, the only words induced by the partition are A_1^ω, A_2^ω, ... and A_N^ω. Given a Büchi automaton, it is enough to check whether one of these paths starting from an initial state of the automaton passes infinitely often through a final state. Alternatively it is a consequence of the forthcoming Proposition 23. The map $x \mapsto 0x$ on $\{0,1\}^{\mathbb{N}}$ with a unique attracting fixed point 0^ω is decidable. This follows from Proposition 21. The full shift on any finite alphabet is a decidable system by a corollary to Proposition 27.

If a system is not decidable, how undecidable can it be? We show that the problem of model-checking is at most Σ_1^1-complete, which is rather highly undecidable. A Σ_1^1 *set* is the set of integers m satisfying a formula of the kind

$$\exists k,\, Q_1 n_1,\, \cdots,\, Q_i n_i : R(k, m, n_1, \ldots, n_i),$$

where k runs over all total functions from $\mathbb{N}$ to $\mathbb{N}$, Q_1, ..., Q_i are quantifiers, $n_1, \ldots, n_i$ run over $\mathbb{N}$, and R is a recursive relation. By 'recursive' we mean that there is a Turing machine with k as oracle and $m, n_1, \ldots, n_i$ as data that decides in finite time whether $R(k, m, n_1, \ldots, n_i)$ holds or not. Oracles are defined in Section 2.7. A Σ_1^1 set is Σ_1^1-*complete* if every Σ_1^1 is many-one reducible to it. The class of Σ_1^1 problems belongs to the so-called analytical hierarchy; see [Rog87] for more details.

Proposition 15 *The problem of model-checking for Büchi automata on an effective symbolic system is at most Σ_1^1-complete.*

Proof. Let $f : X \to X$ be an effective symbolic system. First we show the the model-checking problem is in Σ_1^1. Then we construct a system simulating a

universal Turing machine with oracle for which the problem of model-checking is Σ_1^1-complete. The proof, although rigorous, is not completely formalized.

We can suppose that the space X of the system is an effective closed subset of the Cantor space $2^{\mathbb{N}}$. Let x be a sequence taking values in $\mathbb{N}$. Then the assertion '$x \in X$' is equivalent to the formula '$\forall t \in \mathbb{N} : R_X(x,t)$', where the relation $R_X(x,t)$ is the recursive relation '$x_0, x_1, \ldots, x_t \in \{0,1\}$ and $[x_0 x_1 \ldots x_t] \cap X \neq \emptyset$'. Let m be a natural integer encoding a Büchi automaton whose alphabet is a partition of X. Here Büchi automata are of easier use than Muller automata. Recall that a Büchi automaton is given by a finite set of states, an alphabet and a transition relation, a set of initial states and a set of final states. For any $x \in X$, call $R_f(x,m,t)$ the relation 'For the initial condition x, the Büchi automaton m observing the system can be in a final state at time t'. We say 'can be' to take into account the possible nondeterminism of the automaton. It is a recursive relation; the configuration x can be seen as a function from $\mathbb{N}$ to $\mathbb{N}$. Then the problem of model-checking can be expressed by the logical formula '$\exists x : x \in X$ and $\forall t, \exists t' > t : R_f(x,m,t')$', with m as free variable; hence model-checking for Büchi automata is in Σ_1^1.

The set of natural integers n such that there exists a sequence of integers $k : \mathbb{N} \to \mathbb{N}$ for which the universal Turing machine with initial data n and oracle k does not halt is well known to be Σ_1^1-complete; see [Rog87]. Consider an oracle universal Turing machine of the following form. The Turing machine has two tapes, one of which encodes the oracle function k at the right side of the initial position of the head under the form $10^{k(0)}10^{k(1)}10^{k(2)}1\ldots$. We can suppose that tape alphabet is $\{0,1\}$, without blank symbol. The head has access to both tapes. Not every possible content of the oracle tape is a valid oracle; indeed the word 0^ω cannot appear on the tape. We can suppose without loss of generality that when the head wants to query $k(i)$, it first checks that $k(i)$ is properly encoded by scanning the tape in some state q_{search} until it discovers a 1 and then jumps to the state q_{found}. This two-tape Turing machine is an effective dynamical system, as described in Sections 4.3 and 2.7.

Call Q the states of the head, q_0 the initial state and q_h the halting state. It can be supposed that it is impossible to leave q_h once we reach it. We want to know whether there is an initial configuration of this system, composed of a state of Q and the contents of both tapes, such that we start in the clopen set $\{q_0\} \times [n] \times [1]$ (i.e., the head is in state q_0, the initial data n is encoded at the right of the head on the first tape and a symbol 1 is read by the head on the second tape) and the head reaches infinitely often $Q \setminus \{q_{\text{search}}, q_h\}$. For if an initial configuration is such that the head does not reach infinitely often $Q \setminus \{q_{\text{search}}, q_h\}$, then it either reaches the halting state or gets stuck in a query on an invalid oracle.

This property can be observed by a Büchi automaton in a straightforward manner. Indeed call U the clopen set $\{q_0\} \times [n] \times [1]$, call V the clopen set $(Q \setminus \{q_{\text{search}}, q_h\}) \times \{0,1\}^{\mathbb{Z}} \times \{0,1\}^{\mathbb{Z}}$ of configurations with the head in $Q \setminus$

$\{q_{\text{search}}, q_h\}$ and let W. Then we want to accept the configurations that induce the ω-language $U(X^*V)^\omega$, in the partition of configuration space X into $U \setminus V$, $V \setminus U$, $U \cap V$ and $(U \cup V)^c$. We use Kleene's theorem for ω-regular languages to find a Büchi automaton. Alternatively, it is easy to directly design a suitable Büchi or Muller automaton.

Putting all together, we have constructed a reduction from a Σ_1^1-complete problem to a model-checking problem for Büchi automata of some fixed symbolic system; the latter is therefore Σ_1^1-complete as well. □

We are now ready to state the main definition of computational universality. We define a universal symbolic system as a special kind of undecidable system. The universality of Turing machines is a particular example of this definition, as we show below.

An effective dynamical system is *universal* if the model-checking problem for finite automata is r.e.-complete.

An *r.e.-complete* problem, or *Σ_1-complete* problem, is a recursively enumerable problem, to which any recursively enumerable problem is many-one reducible.

Recall that the model-checking problem for finite automata is checking whether the the language induced by a given partition intersects a given regular language. This problem is always recursively enumerable, because the language induced by a clopen partition is recursively enumerable and the regular language accepted by a finite automaton is recursive; the intersection can be recursively enumerated and if it is nonempty then we can know this after a finite time. Universality is obviously preserved by effective conjugacies, and a symbolic system with a universal factor is also universal.

Proposition 16 *A universal system is not decidable.*

Proof. If the model-checking for Büchi automata is decidable then so is the model-checking problem for finite automata. Indeed, the latter is reducible to the former in the following way. Given a finite automaton, modify it in a such a way that the final states are fixed points of the transition function, whatever the input is; the resulting automaton is interpreted as a Büchi automaton, with the same set of final states. □

Note that a nondeterministic scheme of computation underlies the definition of universality. The computation succeeds if and only if at least one trajectory exhibits a given behavior. For example, recall from Section 4.4 that the halting problem consists in determining, given the clopen sets U and V, whether there is a configuration in U that eventually reaches V. We may think of V as the halting set and of U as an initial configuration of which we know only the

first digits. The unspecified digits of the initial configuration may be seen as encoding the nondeterministic choices occurring during the computation.

4.5.1 Examples

Turing machines with blank symbol.

A Turing machine with blank symbol that is universal in the sense of Turing, is also universal according to our definition, because the halting problem 'Can we go from a clopen set U to a clopen set V?' is r.e.-complete. Indeed the halting problem restricted to clopen sets that are isolated points is already r.e.-complete. Recall that isolated points are exactly finite configurations. Incidentally, we have shown that what we have called 'halting problem' for a general symbolic system is indeed a generalization of the usual halting problem for Turing machines.

Turing machines without blank symbol.

It is only slightly more complicated to build a universal Turing machine without blank symbol. In such a Turing machine, there is no obvious notion of 'finite configuration'. The trick is basically to encode the initial data in a self-delimiting way. Take a Turing machine that is universal in the sense given by Turing. Then add two new symbols L and R to the tape alphabet. On an initial configuration, put an L on the left end and an R on the right end of the encoded data. When the head encounters an L, it pushes it one cell to the left, leaving some more space available for computation. It acts similarly for an R symbol. The working space is always delimited by an L and an R; the symbols situated outside this zone are considered as noise, and do not influence the computation. For this modified universal Turing machine, the (clopen-set-to-clopen-set) halting problem is again undecidable.

Cellular automata.

Let us take a universal Turing machine with a blank symbol. We suppose that when the halting state is reached, then the head comes back to the cell of index 0. We can simulate it in a classic way with a one-dimensional cellular automaton. The alphabet of the automaton is $A \cup (A \times Q) \cup \{L, R, Error\}$, where A is the tape alphabet (including the blank symbol) and Q the set of states. Let us take a point in the cylinder $[L, \text{initial data of the Turing machine}, R]$, and observe its trajectory. The symbol L moves to the left at the speed of one cell per time step, leaving behind blank symbols. The symbol R moves to the right in a similar way. Meanwhile, the space between L and R is used to simulate the Turing machine and is composed of symbols from A and exactly

one symbol from $A \times Q$, which denotes the position of the head. When L or R symbols meet each other, then a spreading *Error* symbol is produced, that erases everything.

This cellular automaton is again universal, because the (clopen-set-to-clopen-set) halting problem is r.e.-complete. Indeed, there is an orbit from the cylinder $[L, \text{initial data of the Turing machine}, R]$ to the cylinder $[A \times \{\text{halting state}\}]$ (both cylinders centered at cell of index zero) if and only if the universal Turing machine halts on the initial data.

Tag systems.

Tag systems were introduced by Post in 1920. A *tag system* is a transformation rule acting on finite binary words. At every step, a fixed number of bits is removed from the beginning of the word and, depending on the values of these bits, a finite word is appended at the end of the word. Minsky [Min61] proved that there is a so-called universal tag system, for which checking whether a given word will eventually produce the empty word when repeating the transformation is an r.e.-complete problem.

To infinite words, just remove the fixed number of bits. This extends the rule of tag systems from finite to infinite words in a continuous way. Thus we have a dynamical system on the compact space $\{0,1\}^* \cup \{0,1\}^{\mathbb{N}}$, in which finite words are clopen sets. Again, if the tag system is universal for the word-to-word definition, then it is universal for our definition with the halting problem on clopen sets of $\{0,1\}^* \cup \{0,1\}^{\mathbb{N}}$.

Counter machines.

A k-counter machine is composed of k counters, each containing a nonnegative integer, and a head that can test which counters are at zero and can increment or decrement every counter (with the convention $0 - 1 = 0$). Thus a counter machine is a map $f : Q \times \mathbb{N}^k \to Q \times \mathbb{N}^k$, where Q is the finite set of states of the head. There exists such a machine f for which given two configurations $x, y \in Q \times \mathbb{N}^k$, the problem to check whether the trajectory of x reaches y is r.e.-complete; see Minsky [Min61].

Let $\mathbb{N} \cup \{\infty\}$ be the topological space with the metric $d(n,m) = |2^{-n} - 2^{-m}|$. This is effectively homeomorphic to the set $\{1^n 0^\omega | n \in \mathbb{N}\} \cup \{1^\omega\}$. The map f is easily extended to the compact space $Q \times (\mathbb{N} \cup \{\infty\})^k$, with the convention $\infty \pm 1 = \infty$. The points of $Q \times \mathbb{N}^k$ are clopen sets of $Q \times (\mathbb{N} \cup \{\infty\})^k$, hence f is universal for the halting problem, even if $k = 2$.

Collatz functions.

We can also apply our definition to functions on integers. Some functions on $\mathbb{N}$ may be extended to $\mathbb{N} \cup \{\infty\}$. For instance, the famous $3n + 1$ function sends

even ns to $n/2$, odd ns to $3n+1$ and ∞ to ∞. Whether this map is decidable is unsettled.

Collatz functions generalizes the $3n+1$ function. For a fixed k, find the remainder of n divided by k. Apply a linear map depending on the remainder, such that the result is always a natural integer. Conway [Con72] proved that there exist universal Collatz functions, by showing that such maps can simulate counter machines.

More examples.

In Section 4.7 we give an example of a universal system that is chaotic, and for which the halting problem is decidable, but not the variant expressed by logical formula (4.2). In Section 4.6.4 we build a system which is neither decidable nor universal. In the setting of point-to-point properties, it was proved by Sutner [Sut03] that there exist cellular automata with a halting problem of an intermediate degree between decidability and r.e.-completeness. The same kind of examples for Turing machines are known for long time (Friedberg-Muchnik theorem, see for instance [Rog87]). However we have not been able to build a system for which finite-automata properties of trajectories are undecidable, but not r.e.-complete.

4.6 Sufficient conditions for decidability

The purpose of this section is to link computational capabilities of a system to its dynamical properties. Most results proved in this section are in fact sufficient conditions of decidability and can thus be interpreted as necessary conditions for universality. For instance, we prove that minimal systems are decidable, hence universal systems are not minimal.

The following constructions and propositions are useful in several proofs. Given an effective system $f : X \to X$, a clopen partition $\mathcal{A} = \{A_1, \dots, A_N\}$ of X and a transition function $\Delta : Q \times \mathcal{A} \to Q$ of a deterministic automaton, we construct the *observation system* $f_\Delta : X \times Q \to X \times Q$ by

$$f_\Delta(x, q) = (f(x), \Delta(q, A_i)), \text{ where } x \in A_i.$$

Clearly f_Δ is an effective symbolic system, and the projection $\pi_X : X \times Q \to X$ is an effective factor map of f_Δ to f.

We say that a dynamical system $f : X \to X$ has *clopen basins*, if for every clopen set $V \subseteq X$, its basin $\mathcal{B}(V) = \bigcup_{n \geq 0} f^{-n}(V)$ is a clopen set.

Proposition 17 *If $f : X \to X$ is an effective system with clopen basins, then the operation $V \mapsto \mathcal{B}(V)$ is computable.*

Proof. If V and $\mathcal{B}(V)$ are clopen sets, then by compactness there exists $m > 0$ such that

$$\mathcal{B}(V) = \bigcup_{n<m} f^{-n}(V) = \bigcup_{n<m+1} f^{-n}(V).$$

Given V we can determine m effectively so the operation $\mathcal{B}(V)$ is effective too. Consequently, there exists a computable total function $k : \mathbb{N} \to \mathbb{N}$ such that $\mathcal{B}(P_n) = P_{k(n)}$, where P_n is the clopen set of index n. □

Proposition 18 *If an effective system is such that for any transition function, the resulting observation system has clopen basins, then the system is decidable.*

Proof.

For every clopen partition $\mathcal{A}$, for every finite set Q and for every transition function $\Delta : Q \times \mathcal{A} \to Q$, the system $f_\Delta : X \times Q \to X \times Q$ has clopen basins.

Assume now that $V \subseteq X \times Q$ is clopen, so that $\mathcal{B}(V)$ is clopen and the index of $\mathcal{B}(V)$ can be computed from the index of V. Moreover $I(V) \triangleq \mathcal{B}(\mathcal{B}(V)^c)^c$ is a clopen set too and its index can be again computed from that of V. A point (x, q) belongs to $I(V)$ iff the trajectory of (x, q) passes through V infinitely often. Given $q_0, q_1 \in Q$, then $(X \times \{q_0\}) \cap I(X \times \{q_1\})$ is again a computable clopen set, so the set $\{x \in X : \forall n, \exists m > n, f_\Delta^m(x, q_0) = q_1\}$ is computable as well. It follows that for a family $\mathcal{F}$ of subsets of Q, the set

$$\{x \in X : \{q \in Q : \forall n, \exists m > n, f_\Delta^m(x, q_0) = q\} \in \mathcal{F}\}$$

is computable too. In particular, whether this set is empty can be decided algorithmically. Hence the model-checking problem for Muller automata is decidable. □

4.6.1 Minimality

We first prove that an undecidable system must have a 'thin' subsystem.

Proposition 19 *A symbolic system such that all nonempty subsystems have a nonempty interior is decidable.*

Proof. Consider an effective symbolic dynamical system $f : X \to X$ whose subsystems have a nonempty interior, a partition $\mathcal{A} = \{A_1, A_2, \ldots, A_N\}$ of X and consider a Muller automaton whose set of states is Q, alphabet is $\{A_1, A_2, \ldots, A_N\}$ and transition function is $\Delta : Q \times \mathcal{A} \to Q$.

We check that the observation system has no subsystem with an empty interior (except the empty set). Indeed let $Y \subseteq X \times Q$ a closed subset such that $f_\Delta(Y) \subseteq Y$. For every $q \in Q$, call $Y_q \subseteq X$ the projection on X of $Y \cap X \times \{q\}$. Then the projection of Y is the union of all Y_q and is a subsystem of X. By hypothesis, it has a nonempty interior; thus one of the Y_q has a nonempty interior (this follows from Baire's theorem). This implies that one of the sets $Y \cap X \times \{q\}$ has a nonempty interior. Hence Y itself has a nonempty interior in $X \times Q$.

We want to show that the observation system f_Δ has clopen basins.

Given a clopen set $V \subseteq X \times Q$, we show that the border $\partial\mathcal{B}(V)$ is f_Δ-invariant. Indeed $f_\Delta(\mathcal{B}(V) \setminus V) \subseteq \mathcal{B}(V)$ and, since V is clopen, $\partial\mathcal{B}(V) = \partial(\mathcal{B}(V) \setminus V)$. If $y \in \partial\mathcal{B}(V)$ and U is a neighborhood of $f_\Delta(y)$, then $f_\Delta^{-1}(U)$ is a neighborhood of y that contains a point $z \in \mathcal{B}(V) \setminus V$. Thus $f_\Delta(z) \in U \cap \mathcal{B}(V)$. Since U is arbitrary, this proves that $f_\Delta(y) \in \partial\mathcal{B}(V)$. But $f_\Delta(y) \notin \mathcal{B}(V)$, because $y \notin \mathcal{B}(V) \setminus V$. This proves the invariance of $\partial\mathcal{B}(V)$.

Since $\partial\mathcal{B}(V)$ is a subsystem with empty interior, it must be empty, so $\mathcal{B}(V)$ is clopen. So the observation system has clopen basins. By Proposition 18, this proves the system f to be decidable. □

Recall that a *minimal* dynamical system is a system with no subsystem (except the empty set and itself). In minimal system, all orbits are dense and the basin of any clopen set is the full set. A minimal system trivially satisfies the hypothesis of the preceding proposition, and so we have the following.

Proposition 20 *A minimal symbolic system is decidable.*

This is in a way not surprising since in some way all trajectories of a minimal system have the same behavior. For instance, consider the *Fibonacci subshift*. The Fibonacci is the smallest nonempty subshift that is invariant under the map $\vartheta : \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}} : x_0x_1x_2\ldots \mapsto y_0y_1y_2\ldots$, where $y_i = 10$ if $x_i = 1$ and $y_i = 1$ if $x_i = 0$. It is a minimal effective subshift (see [Kůr04]), hence it is decidable.

Recall that any dynamical system has a minimal subsystem, thanks to Zorn's lemma and compactness. In particular, any point comes arbitrarily close to a minimal system, since the closed orbit of the point is itself a dynamical system. Suppose that the symbolic system is not minimal but consists of one minimal subsystem attracting the whole space of configurations. In other words, the limit set is minimal. The limit set of a dynamical system $f : X \to X$ is the set $\bigcap_{n\geq 0} f^n(X)$. Then such a system is again decidable. The following proposition is more general.

Proposition 21 *A symbolic system whose limit set is the union of finitely many minimal systems is decidable.*

Proof. Given a symbolic system $f : X \to X$ and a Muller automaton whose set of states is Q, we build the observation system $f_\Delta : X \times Q \to X \times Q$.

First we prove that the observation system f_Δ contains finitely many minimal sets. Let $X_1, \dots, X_k$ be the minimal subsystems of $f : X \to X$. For every $i = 1, \dots, k$ choose an arbitrary point $x_i \in X_i$. A minimal subsystem of f_Δ, when projected on X, is exactly a minimal subsystem of f, as easily seen. Thus any minimal subsystem of f_Δ must contain at least one point of the form (x_i, q), for some $q \in Q$. Since any two different minimal subsystems are disjoint, this means that there are at most $k|Q|$ minimal subsystems in f_Δ.

Then we show that the limit set of f_Δ is exactly the union of all minimal subsystems.

It is clear that the minimal subsystems are in the limit set of f_Δ. Now we prove that each minimal subsystem Z of f_Δ has a nonempty interior in the limit set of f_Δ (for the subset topology). The projection of the limit set of f_Δ on X is the limit set of f. The projection of Z on X is a minimal subsystem of f, which has a nonempty interior in the limit set of f, and the projection of Z is $\bigcup_{q \in Q} Z_q$, where $Z = \bigcup_q Z_q \times \{q\}$. From Baire's theorem, one of these Z_q has a nonempty interior in the limit set of f, and Z itself has a nonempty interior in the limit set of f_Δ.

Let Y_i be a set included in Z_i that is open in the limit set of f_Δ, where $Z_1, \dots, Z_m$ are the minimal subsystems of f_Δ. All sets $\bigcup_{n \in \mathbb{N}} (f_\Delta^{-n}(Y_i))$ are disjoint sets, are open in the limit set and cover the limit set, since the closed orbit of every point in the limit set of f_Δ must include a minimal subsystem. From compactness, all points of the limit set of f_Δ fall in a minimal subsystem in bounded time. We conclude that the union of all minimal subsystems is the exactly the limit set of f_Δ.

So the limit set of the observation system f_Δ is a finite union of minimal subsystems. We get from the lemma below that f_Δ has clopen basins. From Proposition 18 we deduce that f is decidable. □

For instance, the system $f : \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}} : x \mapsto 0x$ is decidable, as expected. The following lemma finishes the proof.

Lemma 4.1. *A symbolic system whose limit set is the finite union of minimal systems has clopen basins.*

Proof. Suppose that the limit set is $Y_1 \cup \dots \cup Y_k$, where Y_i are minimal subsystems, so that $Y_i \cap Y_j = \emptyset$ for $i \neq j$. Let $V \subseteq X$ be a clopen set. If $V \cap Y_i = \emptyset$, then $\mathcal{B}(V) \cap Y_i = \emptyset$. If $V \cap Y_i \neq \emptyset$, then for some $m > 0$, $Y_i \subseteq V_m = \bigcup_{n<m} f^{-n}(V)$. Thus there exists $m > 0$ such that for all i either $Y_i \subseteq V_m$ or $Y_i \cap V_m = \emptyset$. Then $W_m = f^{-m}(V) \setminus V_m$ is a clopen set disjoint from the limit set. From compactness there exists $k > 0$ such that $f^{-k}(W_m) = \emptyset$, so $\mathcal{B}(W_m)$ is a clopen

set. It follows that $\mathcal{B}(V) = V_m \cup \mathcal{B}(W_m)$ is a clopen set too. □

The above proposition is in fact more general than Proposition 19:

Proposition 22 *A symbolic system such that all nonempty subsystems have a nonempty interior has a limit set composed of finitely many minimal subsystems.*

Proof. Let f be a system such that all nonempty subsystems have a nonempty interior. In the interior of every minimal subsystem choose a clopen set U_i. The basin of the open set $\bigcup_i U_i$ is the full space, because every point of the system must come arbitrarily close to some minimal subsystem, thus must fall in some U_i. By compactness, there is a finite set of is and a natural integer m such that $\bigcup_{i\in I} \bigcup_{n<m} f^{-n}(U_i)$ is the full space. So there are finitely many minimal subsystems, and every point falls in a finite time into a minimal subsystems. The union of the minimal subsystems is therefore the limit set. □

A stronger statement than Proposition 21 is suggested by the intuition that an undecidable system (and especially a universal system) is likely to be able to 'simulate' many other systems, for example all Turing machines. One could expect that every simulated system would be simulated in a different subset of the universal system. We can also assume that the simulation preserves some of the dynamical properties. For instance, it seems reasonable to think that a subsystem simulating a Turing machine with periodic points will have periodic points. We therefore state the following conjecture.

Conjecture 1 *A universal symbolic system has infinitely many minimal subsystems.*

Of course, nothing proves that universal systems must 'simulate', say, Turing machines, and we have not given a precise meaning to 'simulation'.

4.6.2 Equicontinuity

A system $f : X \to X$ is *equicontinuous* if for every $\epsilon > 0$ there exists a $\delta > 0$ such that $d(x, y) < \delta$ implies $d(f^t(x), f^t(y)) < \epsilon$, for any points x, y and $t \in \mathbb{N}$. In other words, close points have close trajectories. Note that equicontinuity in symbolic systems is a topological property not just a metric one. Instead of 'For every $\epsilon > 0$, there is a δ ...' we could say 'For every clopen partition, there is a finer clopen partition such that if two points are in the same subset of the finer partition, then they generate the same infinite word in the coarser partition.'

Proposition 23 *An equicontinuous symbolic system is decidable.*

Proof. First we prove that an equicontinuous system has clopen basins. Let V be a clopen set. Then from equicontinuity there exists a δ such that any two points distant of less than δ either both eventually reach V or both never reach V. Hence $\mathcal{B}(V)$ is the union of balls of radius δ, and is a clopen set.

Then we show that the observation system of an equicontinuous symbolic system $f : X \to X$ is equicontinuous. The space $X \times Q$ (for a finite set Q) can be endowed the metric $d((x, q), (x', q')) = d(x, x')$ if $q = q'$ and 1 otherwise. For a partition $\mathcal{A} = \{A_1, \ldots, A_n\}$ of X, let ζ such that every A_i is a union of open balls of radius ζ. Then for any ϵ, choose a δ such that any two points of X distant of less than δ have orbits distant of less than $\min(\epsilon, \zeta)$. Then two points of the observation system distant of less than $\delta < 1$ have the same second component, will always be in the same partition subset and will always have the same second component. Hence their distance will always be less than ϵ. Consequently, the observation is equicontinuous.

It follows by Proposition 18 that the system is decidable. □

We say that a point x of a dynamical system f is *sensitive* if there is an $\epsilon > 0$ such that for every $\delta > 0$ there is a point y with $d(x, y) < \delta$ and a natural integer t such that $d(f^t(x), f^t(y)) > \epsilon$. It is easy to show by compactness that an equicontinuous discrete-time dynamical system is exactly a system with no sensitive point. Hence, Proposition 23 implies that an undecidable symbolic system must have a sensitive point. Equicontinuity in the case of cellular automata has been given a combinatorial characterization in [Kůr97a], where it is also proved that equicontinuous cellular automata are eventually periodic, thus confirming in this particular case that equicontinuity is incompatible with computational universality.

4.6.3 Regular Systems

A subshift is called *sofic* if its language is regular. A symbolic system is called *regular*, if all its induced subshifts are sofic; see [Kůr93, Kůr99]. Can a regular system be universal? We first consider a closely related question. We say that an effective system is *effectively regular* if it is regular and there is an algorithm that builds from a given clopen partition the finite automaton recognizing the regular language induced by the partition.

Proposition 24 *An effectively regular system is decidable.*

Proof. The intersection of two ω-regular languages is well known to be an ω-regular language. Indeed, the union of ω-regular languages is ω-regular, because

ω-regular languages can be represented by ω-regular expressions. And the complement of an ω-regular language is ω-regular: just complement the family of subsets in a Muller automaton accepting the language.

Moreover, whether the language accepted by a given Muller automaton is empty is a decidable problem too. A sofic subshift is an ω-regular set: the finite automaton accepting the language, interpreted as a Büchi automaton with the same set of final states, accepts the sofic subshift.

Suppose that we are given an effectively regular system, a clopen partition $\mathcal{A}$ of the space and a Büchi or Muller automaton over the alphabet $\mathcal{A}$. Then we construct another Muller automaton that accepts exactly the subshift induced by $\mathcal{A}$ and verify whether the languages accepted by these two Muller automata has a nonempty intersection. Hence the system is decidable. □

If the system is regular but not effectively regular, then the argument of the proof fails.

Proposition 25 *There exists a symbolic system that is regular and universal.*

Proof. Let X_n be the subshift of $\{0,1\}^{\mathbb{N}}$ whose forbidden words are words of the form 10^t1, where t is less than the (possibly infinite) halting time of the universal Turing machine launched on data n. If the Turing machine does not halt, then X_n is the sofic subshift $\{0^*10^\omega, 0^\omega\}$. If the Turing machine halts in k steps, then X_n is the subshift of finite type with forbidden words $11, 101, 1001, \ldots, 10^{k-1}1$ (see Section 2.6 for a definition of subshift of finite type). So all subshifts are sofic, but we cannot effectively build the automaton recognizing the language, for it would allow to solve the halting problem.

Now consider the product of all X_n. This product is again an effective symbolic system X, and all its induced subshifts are sofic, due to the fact that the finite product of sofic subshifts is a sofic subshift and the induced subshift of a sofic subshift is again sofic; see [Kůr04]. Thus the system is regular, but not effectively regular. Finally, it is r.e.-complete to check whether there is a trajectory starting from $\pi_n^{-1}([1])$ which eventually reaches $\pi_n^{-1}([01])$. Here $\pi_n : X \to X_n$ is the canonical projection. □

4.6.4 Shadowing property

Let $f : X \to X$ be a symbolic dynamical system. A *δ-pseudo-orbit* is a (finite or infinite) sequence of points $(x_n)_{n\geq 0}$ such that $d(f(x_n), x_{n+1}) < \delta$ for all n. A point x *ϵ-shadows* a (finite or infinite) sequence $(x_n)_{n\geq 0}$ if $d(f^n(x), x_n) < \epsilon$ for all n. See Figure 4.4 for an illustration. A dynamical system is said to have the *shadowing property* if for every $\epsilon > 0$ there is a $\delta > 0$ such that any δ-pseudo-orbit is ϵ-shadowed by some point. If moreover such a rational δ can

be effectively computed from a rational ϵ then we say that the system has the *effective shadowing property.*

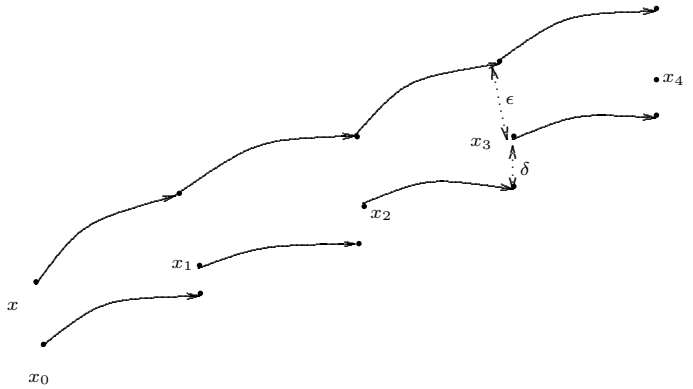

Fig. 4.4. The sequence $x_0, \ldots, x_4$ is a δ-pseudo-orbit which is ϵ-shadowed by x. Every arrow links a point to its image.

For example, the one-sided and two-sided shifts have the shadowing property for $\delta = 2\epsilon$. For instance, the 2^{-2}-pseudo-orbit $00000\ldots$, $00010\ldots$, $00120\ldots$, $01230\ldots$, $12320\ldots$ is 2^{-3}-shadowed by the configuration $00001232\ldots$. By a theorem of Walters, a subshift of finite type has the shadowing property; see [Kůr04] for a proof.

It is easy to see that the effective shadowing property is invariant under effective conjugacies. We can give the following interpretation to the effective shadowing property. Suppose that we want to compute numerically the trajectory of x such that at every step numerical errors are bounded by δ. The resulting sequence of points is a δ-pseudo-orbit, and the shadowing property ensures that this pseudo-orbit is ϵ-close to an actual trajectory of the system, ensuring that the result of the numerical computation is not meaningless.

Proposition 26 *A symbolic system (effective or not) with the shadowing property is regular. An effective symbolic system with the effective shadowing property is effectively regular.*

Proof. The proof generalizes Proposition 5.69 of [Kůr04] about cellular automata. Consider a symbolic system $f : X \to X$ with the shadowing property and a clopen partition $\mathcal{A} = \{A_1, \ldots, A_N\}$. There exists an ϵ such that all clopen sets of the partition are finite unions of balls of radius ϵ. By the shadowing property, there exists δ such that every δ-pseudo-orbit is ϵ-shadowed. We may suppose without loss of generality that $\delta \leq \epsilon$. Let $\mathcal{B} = \{B_1, \ldots, B_M\}$

the clopen partition where each B_i is a ball of radius δ. Then the set of of all infinite words induced by all δ-pseudo-orbits through $\mathcal{B}$ is a subshift of finite type: the word B_iB_j is forbidden iff $B_i \cap f^{-1}(B_j) = \emptyset$, i.e., we cannot go from B_i to B_j in one step. But the partition $\mathcal{A}$ is coarser than $\mathcal{B}$, so the subshift induced by $\mathcal{A}$ is a factor of a subshift of finite type, hence is sofic. If the system has the effective shadowing property, then we can effectively find δ, effectively describe the subshift of finite type and effectively build the sofic subshift. □

Proposition 27 *A symbolic system that has the effective shadowing property is decidable.*

Proof. By Propositions 26 and 24. □

In particular, the full shift and any subshift of finite type are decidable. Proposition 27 is stronger than Proposition 23, as we now show.

Proposition 28 *An equicontinuous effective symbolic system has the effective shadowing property.*

Proof. Let $f : X \to X$ be an equicontinuous system. Then for every $\epsilon > 0$, there is a δ such that any two points distant of less than δ have ϵ-close trajectories. We show that any δ-pseudo-orbit is ϵ-shadowed by some point.

Let $x_0, x_1, x_2, \ldots$ be a δ-pseudo-orbit. We show by induction on m that $d(f^n(x_m), f^{n+m}(x_0)) < \epsilon$ for every m and n. The case $m = 0$ is obvious. If it is true for m then in particular $d(f^{n+1}(x_m), f^{n+m+1}(x_0)) < \epsilon$. But $d(x_{m+1}, f(x_m)) < \delta$ implies $d(f^n(x_{m+1}), f^{n+1}(x_m)) < \epsilon$. From the ultrametric inequality we have $d(f^n(x_{m+1}), f^{n+m+1}(x_0)) < \epsilon$.

It is now enough to prove that a suitable δ is computable from ϵ, i.e., an equicontinuous symbolic system is always 'effectively' equicontinous. Take the partition $\mathcal{B}_0$ of all balls of radius ϵ. For $n = 0, 1, 2, \ldots$, let $\mathcal{B}_{n+1}$ be the coarsest partition finer than $\mathcal{B}_n$ and $f^{-1}(\mathcal{B}_n)$. From equicontinuity, this sequence of finer and finer partitions must stabilize to some $\mathcal{B}_n = \mathcal{B}_{n+1} = \mathcal{B}_{n+2} = \cdots$. To check that we have reached this point it is enough to check that $\mathcal{B}_n = \mathcal{B}_{n+1}$. We choose δ so that the clopen sets of $\mathcal{B}_n$ can be expressed as balls of radius δ.□

We also have the following result.

Proposition 29 *A symbolic system that has the shadowing property is not universal.*

Proof. Let $f : X \to X$ be a symbolic system with the shadowing property.

Given a deterministic finite automaton observing the system through a given clopen partition, the problem is to check whether there exists a finite word induced by the clopen partition that is accepted by the automaton. As we have noticed just above Proposition 16, this problem is recursively enumerable. We show that it is also co-recursively enumerable. This will prove that the problem is decidable and that f is not universal.

Let $\mathcal{A} = \{A_1, \ldots, A_N\}$ be a clopen partition and $\Delta : Q \times \mathcal{A} \to Q$ the transition function of a deterministic finite automaton. We must essentially prove that the halting problem is decidable for the observation system $f_\Delta : X \times Q \to X \times Q$.

But f_Δ is an effective symbolic system with the shadowing property, as we now show. We can suppose that the distance between (x, q) and (x', q') is 1 if $q \neq q'$ and $d(x, x')$ otherwise. For an $\epsilon > 0$, choose an $\epsilon' \leq \epsilon$ such that any A_i can be written as a union of open balls of radius ϵ'. Then the shadowing property for f yields a corresponding δ'. Choose a $\delta \leq \delta'$ such that $\delta < 1$. Then any δ-pseudo-orbit of f_Δ is ϵ-shadowed by some point of $X \times Q$: such a pseudo-orbit is projected onto a δ-pseudo-orbit of f, which is ϵ-shadowed by some point, and this point can be lifted to a point that ϵ-shadows the pseudo-orbit of f_Δ.

Take two clopen sets $U, V \subseteq X \times Q$. There exists an orbit from U to V iff for every $\delta > 0$ there exists a δ-pseudo-orbit from U to V. The 'only if' part is trivial. For the 'if' part, take a δ corresponding to an ϵ such that U and V are unions of open balls of radius ϵ: the δ-pseudo-orbit from U to V is ϵ-shadowed to an actual orbit, which also goes from U to V; alternatively, see Proposition 2.15 of [Kůr04].

If there is no orbit starting in U that reaches V, then there exists a δ such that no δ-pseudo-orbit goes from U to V, and we can algorithmically check it by the following method. For a fixed δ, define V' as the union of balls of radius δ whose center is in $f_\Delta^{-1}(V)$. Then compute V'', V''', and so on. As there are only finitely many balls of radius δ, $V^{(t)} = V^{(t+1)}$ for some t. Then check whether $V^{(t)} \cap U$ is empty; it is the case if and only if there is no δ-pseudo-orbit from U to V. Start again with smaller and smaller δ.

Thus the halting problem for f_Δ is decidable. In particular if $U = X \times \{q_0\}$ (where q_0 is the initial state of the automaton) and $V = X \times F$ (where $F \subseteq Q$ is the set of final states of the automaton), then we can algorithmically check whether there exists a point of X which induces through the clopen partition a word that is accepted by the automaton. □

The following proposition shows that the effective shadowing property is stronger than the shadowing property.

Proposition 30 *There exists an undecidable effective symbolic system that has the shadowing property, but not the effective shadowing property.*

Proof. Let X_n be the subshift with forbidden words 0^t, where the universal Turing machine stops on data n in at most t steps. If the Turing machine does not halt on n, then X_n is the full shift; if it stops in k steps, then the forbidden word is 0^k. All these subshifts are effective, but we cannot compute their set of forbidden words.

The product X of all X_n is an effective system. Whether there is a point that remains for ever in $\pi_n^{-1}[0]$ is co-r.e.-complete (where $\pi_n : X \to X_n$ is the canonical projection). This property has been shown in Figure 4.3 to be expressible in terms of Muller automata. Hence the system is undecidable.

Recall that a subshift of finite type has the shadowing property. We show that the countable product of subshifts that have the shadowing property also has the shadowing property. A ball of radius ϵ in the product system may be expressed as the finite union of products of balls of radius ϵ' in a finite number of constituent subshifts. We choose the smallest of the corresponding δ' given by shadowing property in the subshifts. The product of balls of radius δ' may be expressed as union of balls of radius δ; this is the δ corresponding to ϵ.

Hence the system X has the shadowing property but not the effective shadowing property, since it is undecidable. □

As the shadowing property implies nonuniversality, it also proves that universality is stronger than undecidability.

Corollary 3 *There exists a symbolic system that is neither decidable nor universal.*

Note also that Turing machines that satisfy the shadowing property have been given a combinatorial characterization in [Kůr97b]; in particular, the proof shows that the link between ϵ and δ (of the definition of shadowing property) is linear. Hence the effective shadowing property is not stronger than the shadowing property in the case of Turing machines.

4.7 A universal chaotic system

According to Devaney [Dev89], a discrete-time dynamical system is *chaotic* if it is infinite, topologically transitive and has a dense set of periodic points. A discrete-time dynamical system is *topologically transitive* if for any nonempty open sets U and V, there is a point in U that reaches V. One can prove that a chaotic system is sensitive [BBC+92], i.e., there is an ϵ such that for any point x, there exist points arbitrarily close to x whose trajectory diverges from the trajectory of x by at least ϵ. For instance the (one-sided or two-sided) full shift is chaotic.

It is not difficult to construct a universal subshift. Indeed, in $\{0,1\}^{\mathbb{N}}$ consider all forbidden words of the form 01^n00^t1, where the universal Turing machine launched on data n does not halt in less than t steps. Then the subshift of all configurations avoiding this set of words is effective and universal: the problem 'Can we reach the clopen set [001] from $[01^n0]$?' is r.e.-complete. Modifying this construction, we get the following result:

Proposition 31 *There exists an effective system on the Cantor space that is chaotic and universal.*

Proof. Consider a subshift $X \subseteq \{0,1,\S\}^{\mathbb{N}}$ whose forbidden words are all 01^n00^t1, where the universal Turing machine launched on data n does not halt in less than t steps. Denote by $L \subset \{0,1\}^*$ the language of binary words with no forbidden subword. Then the language of X consists of words $w_1\S w_2\S \dots \S w_n$, where $w_i \in L$. We show that X is a universal chaotic system.

First note that X is a perfect subshift, so it is effectively conjugated to a system on the Cantor space. Then X has dense periodic points: if w is in the language of X, then $(w\S)^\omega$ is in X. Finally X is topologically transitive: for any two finite words v, w of the language of X we can go from $[v]$ to $[w]$ with the point $v\S w\dots$. Thus X is chaotic.

Moreover, given n it is undecidable whether there is a point of $[01^n0]$ that eventually reaches [001] without passing through [§]. This property can be expressed by the finite automaton constructed in Figure 4.2. Thus X is universal. □

Note that the system built in the proof is a one-sided subshift, hence it is expansive: there is an ϵ such that any two points are eventually separated by at least ϵ. Note also that the halting problem for a chaotic symbolic system is trivially decidable, because of the topological transitivity.

The central idea of the 'edge of chaos' is that a system that has a complex behavior should be neither too simple nor chaotic. There are several ways to understand that. Here we interpret 'complex system' by 'universal symbolic system'. Then 'too simple' could refer to the situation treated in Proposition 21: one or several attracting minimal subsystems. This includes of course the case of a globally attracting fixed point. If we take 'chaotic' as meaning 'Devaney-chaotic', then computational universality need not be on the 'edge of chaos', since we have just constructed a chaotic system that is universal.

However, many examples of chaotic discrete-time dynamical systems (whatever the exact meaning given to 'chaotic', and for symbolic systems as well as for analog ones), have the shadowing property. For instance the shift and the so-called Smale's horseshoe (present in some physical systems), as well as the class of so-called hyperbolic systems, satisfy the shadowing property.

Thus we suggest that the term 'edge of shadowing property' would be more appropriate (at least for symbolic systems), although not as thrilling.

Note nevertheless that the 'edge of chaos' has been much studied in cellular automata, and we don't know whether an example of a chaotic universal cellular automaton exists. Also, Devaney's definition of chaos is one of the earliest. Several other definitions have been proposed since then, for instance Li-Yorke's. In physics, positivity of Lyapunov exponents is often taken as a criterion of chaos. For more on these definitions, see for instance [NB99].

4.8 Discussion of universality

Turing [Tur36] justified the form of his machine along the following lines. A human operator applying an algorithmic procedure can be supposed to be at every step of time in a unique mental state. He can be supposed to have finitely many possible mental states, and to have at his disposal a pencil and as much paper as needed, on which he may write out letters or digits. In a finite time he may read or write only finitely many symbols on the paper. Paper is modelled by the tape and the human by a kind of finite automaton that is able to read, write or shift the tape.

Now suppose that the human operator has no paper or pencil, but can observe a (physical realization of) a symbolic dynamical system, without having any control on it. The system can serve as a 'universal computer' if with its help, the human operator is able to solve all problems he could also solve with paper and pencil. As the human operator has finitely many possible mental states, at every step he can distinguish only finitely many configurations of the system. If we group together all points that are undistinguishable between them, we obtain a partition of the system configuration space. We suppose that this partition is clopen, because clopen partitions express in a natural way that finitely many symbols are observed from the system at every step of time, analogously to Turing's assumption. Clopen partitions also express the fact that close enough points are undistinguishable for the observer.

A symbolic system can be used as a 'computer', if it is computationally universal. When we observe the system using a given finite automaton acting on a clopen partition, then deciding whether the finite automaton can reach a final state from an initial state is at least as difficult as deciding the halting problem for a universal Turing machine. This means that when we look for the answer to a recursively enumerable problem, then we can obtain the answer by observing the system, provided that we are 'lucky' and wait long enough.

Our definition of universality perhaps differs in several ways from what we could expect at first glance from a generalization of Turing machine universality. We give now various arguments to support the present definition against seemingly more obvious attempts. In particular, we justify the use of *set-to-set*

properties, observed by *finite automata*, on systems defined by a *computable* map.

4.8.1 Set-to-set properties

Davis [Dav56] proposed the following definition: a Turing machine is universal if the relation 'x_n is in the orbit of x_m' is r.e.-complete, where x_m and x_n are arbitrary finite configurations. This definition has the advantage to bypass the need for a description of a way to encode the input and decode the output of a computation. Many definitions of universality for particular systems (cellular automata, for instance) propose to observe point-to-point properties.

Hemmerling [Hem02] proposes a definition for an effective metric space; the basic idea is to endow a metric space with a countable dense set of points. Examples include the reals with rational points, the Cantor space with ultimately constant configurations, the Cantor space with ultimately periodic configurations. This seems to provide a suitable framework to generalize Davis' definition. Let us say that a metric space endowed with a dense set of points $(x_n)_{n\in\mathbb{N}}$ is universal if the property 'x_n is in the trajectory of x_m' is r.e.-complete.

However, as remarked in [DR99], this leads to conclude that the shift is universal; a consequence that is counter-intuitive. Indeed, consider the set of all configurations with primitive recursive digits. This set is countable and dense. Then we take as an initial configuration the sequence of pairs (state of the head, currently read symbol) of a universal Turing machine during a computation. And we only have to shift it to know whether the halting state will ever appear. It sounds unreasonable to classify the shift among universal systems, because it does not compute anything but just reads the memory.

The definition presented in Section 4.5 overcomes this problem in a simple manner: the user needs only to specify a finite number of bits as an initial condition. Instead of initial *configurations* we should rather talk about initial *sets*, which may be seen as 'fuzzy points', points defined with finite accuracy. This solution is also more satisfactory from the point of view of physical realizability. Indeed, we expect the set of configurations of a physical system to be uncountable in general, and specifying an initial point for the computation means *a priori* that we must give an infinite amount of information. Preparing a physical system to be in a very particular configuration is likely to be impossible, because of the noise or finite precision inherent to every measure.

4.8.2 Finite automata

What kind of property are we going to test on clopen sets (or, equivalently, on induced subshifts)? Here again, we must avoid trivialities. Suppose that we look at identity on the Cantor space. We now choose to observe the following property: a clopen set satisfies the property if and only if its index (i.e., the

integer describing the clopen set) satisfies some r.e.-complete property on $\mathbb{N}$. Then we find that the identity is computationally universal, which is a result not to be desired. The complexity of computation is artificially hidden in the decoding.

On the other hand, we see no reason to restrict ourselves to the sole halting property: 'there is a trajectory from this clopen set to that clopen set'. Any observable property could *a priori* be used as a basis for computation. For instance, the chaotic system built in Section 4.7 is universal but the halting problem is decidable. So we must precisely define a class of observable properties of clopen sets, not too large and not too restricted. Finite automata used to express properties of finite words have been extensively studied in the literature. They also agree with Turing's idea of modelling a human operator as having finitely many possible mental states. We do not use the powerful setting of Büchi or Muller automata in the definition of computational universality, because it may need an infinite time to check that a trajectory has the required property, which goes against the idea that a successful computation should end in a finite time. Whether a given observer Büchi automaton can accept a trajectory of the system is actually a more general question, which is dealt with in our definition of 'decidable system'. Properties observed by Büchi automata are interesting for themselves and are often considered in the theory of model-checking.

4.8.3 Effectiveness

Finally, we show by an example that it is useful to add an effectiveness structure on dynamical systems. Fix an r.e.-complete set $H \subset \mathbb{N}$ of integers and consider the symbolic system $f : \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$ such that $f(1^\omega) = 1^\omega$ and $f(1^n 0 x_0 x_1 x_2 \ldots) = 1^m 0 x_0 x_1 x_2 \ldots$, where m depends on n. If $n \in H$, then m is the largest integer strictly smaller than n such that $m \in H$ or 0 if no such number exists. If $n \notin H$, then $m = n$. Suppose now that $13 \in H$. Then the relation 'the clopen set $[1^n 0]$ will eventually reach $[1^{13}0]$' is r.e.-complete, because H is.

On the other hand, if we were provided with an actual implementation of $f : \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$, we could decide an undecidable problem (namely, H) by observing the trajectories. So there is a discrepancy between the computational complexity of properties of clopen sets and the actual possibilities of the machine. This is because we cannot compute even a single step of f: it is a 'nonsimulable' system. We therefore restrict ourselves to systems such that the inverse image of a clopen set is computable. Note that for instance in [Sie99] Siegelmann allows neural networks with non-recursive weights, leading to a non-computable maps and to super-Turing capabilities.

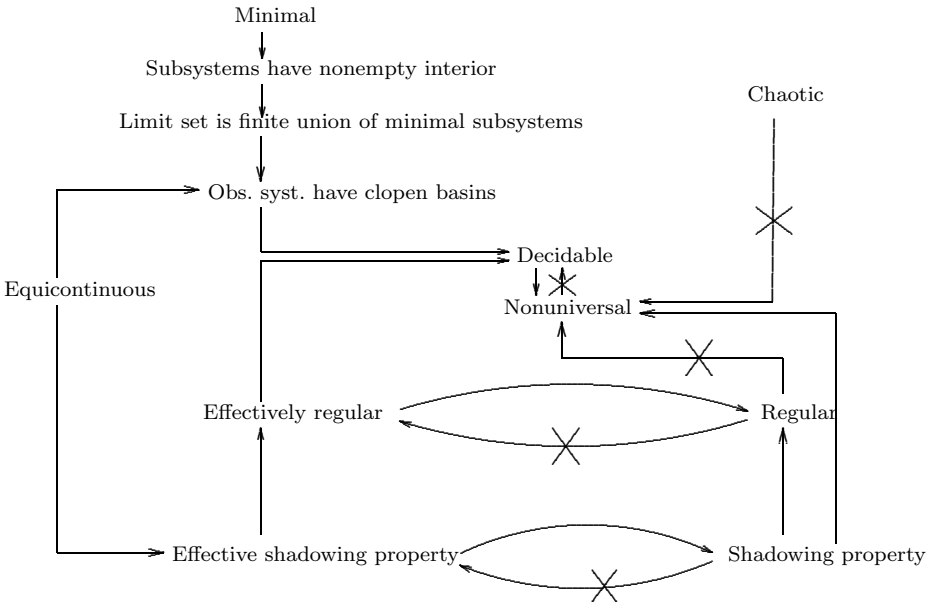

Fig. 4.5. Summary of the results. Arrows read 'implies', crossed arrows read 'does not imply'.

4.9 Conclusions

We provided a definition of decidability and universality for a symbolic systems, and established some links between decidability and the dynamical properties of the system. We also constructed a chaotic system that is universal. These results are summed up in Figure 4.5.

We have already formulated some open problems. Is there a cellular automaton that is chaotic and universal? Do undecidable system have infinitely many disjoint subsystems? And many more questions are yet to be solved. For instance, can we find sufficient conditions of universality? Among the simplicity criteria proposed in [Kůr93], which of those are sufficient conditions for decidability? Are the Game of Life and the automaton 110 universal for our definition? Can a linear cellular automaton be universal?

It also remains to extend the definitions and results to systems in $\mathbb{R}^n$ in discrete time or even continuous time. The resulting definition of universality could then be compared to existing definitions, for instance [BC96, Moo98a, Orp97, SF98]. Then, results such as those of Section 4.6 could hopefully be adapted. For instance, are minimal systems capable of universal computation? Such results could then be applied to physical systems. What systems that can be found in Nature are able to compute? For instance, hyperbolic dynamical

systems are known to have the effective shadowing property. This would suggest that hyperbolic systems are not universal.

A theory a computational complexity could also be investigated. What problems can be solved in polynomial time with a discrete-time dynamical system? Can we formulate a 'P≠NP' conjecture? See [BHSF02, SF98] for theories of complexity in analog computation.

5

Complexity of Control on Finite Automata

5.1 Introduction

In this chapter we consider a control-theoretic problem for input-output automata. Input-output automata, also called transducers, are quite similar to the deterministic finite automata used in Chapter 4. While deterministic finite automata accept or reject a word, input-output automata transform an input word into an output word. The motivation and intuitive description of the content of this chapter is given in Section 1.3, along with a literature background.

Given an input-output automaton in some initial state, we want to make it reach a given target state. In order to achieve this goal, we can feed the automaton directly with a suitable input sequence, computed in advance: this is the open-loop strategy. The input sequence can also be produced by another input-output automaton, whose input alphabet is a trivial one-letter alphabet. Or we can provide a first input, and at every step produce an input that is function of the past outputs of the automaton: this is the feedback strategy. We suppose that the feedback control is also performed by an input-output automaton.

In either case, open-loop or feedback, we quantify the complexity of the problem as the minimal number of states of the controller automaton needed to fulfill the objective.

Then we propose to evaluate the average complexity of controlling a random n-state automaton from a random initial state to a random target state. The random distribution is uniform.

This average complexity is proved to be in $\Theta(\ln n)$ in the case of open-loop control and, in the case of feedback, between $\Omega(\ln n / \ln \ln n)$ and $\mathcal{O}(\ln n)$. This proves that the feedback strategy has no or little advantage over open-loop in terms of complexity when it comes to control a random automaton. Recall that $f(n) = \mathcal{O}(g(n))$, or $g(n) = \Omega(f(n))$, means that $|f(n)| \leq c|g(n)|$ for some $c > 0$

and all large enough n. If $f(n) = \mathcal{O}(g(n))$ and $f(n) = \Omega(g(n))$ then we write $f(n) = \Theta(g(n))$.

The main proposition is stated in Section 5.2, along with preliminary definitions, and proved in Sections 5.3, 5.4 and 5.5. Conclusions and ideas for future work are presented in the last section.

5.2 Formulation and results

In this section we give precise definitions of automaton, control, and complexity of control. Then we describe our main result.

In this chapter, we use the term 'automata' to denote input-output automata, as defined in Section 2.4. We recall the definition.

An (input-output) *automaton* is given by

- a finite set Q, the elements of which are called the *states*;
- a finite set U, called the *input alphabet*;
- an *initial state* $q \in Q$;
- a finite set Y, called the *output alphabet*;
- a *transition function* $\Delta : Q \times X \to Q$;
- an *output function* $\Gamma \in Q \times X \to Q$.

An automaton is a special case of input-output dynamical system, defined in Section 2.4 . To maintain the analogy with deterministic finite automata described in Section 2.5, we include the initial state in the definition. Given an input word $u_0 u_1 \dots u_n$, the automaton starts from the initial state q, moves to the state $q' = \Delta(q, u_0)$ and emits the output letter $y_0 = \Gamma(q, u_0)$. Then it reads the input letter u_1, moves to state $q'' = \Delta(q', u_1)$ and emits the output letter $y_1 = \Delta(q', u_1)$, etc. Finally, the output word corresponding to $u_0 u_1 \dots u_n$ is given by $y_0 y_1 \dots y_n$ (see Figure 5.1 for an example).

We set $\Delta(q, u_0 u_1 \dots u_k) = \Delta(\Delta(q, u_0 u_1 \dots u_{k-1}), u_k)$, for any $u_0 u_1 \dots u_k$ in U^*. Recall that U^* denotes the set of finite words over the alphabet U.

If U has only one symbol then we say that the automaton is *inputless*, because the input contains no information. The length of the sequence is simply the time variable.

Note that we only consider deterministic automata. This implies that the dynamics of the system is perfectly known, as well as the initial state and the output function. We emphasize that our aim is to compare feedback and open-loop in terms of complexity, not in terms of robustness with respect to noise or uncertainty. We leave the case when noise, lack of information or nondeterminism is present for future research.

Suppose we have an automaton and we would like to drive it from one state to another state, by feeding the automaton with an appropriate word. This is what we call the *reachability problem*. An instance is given by an automaton,

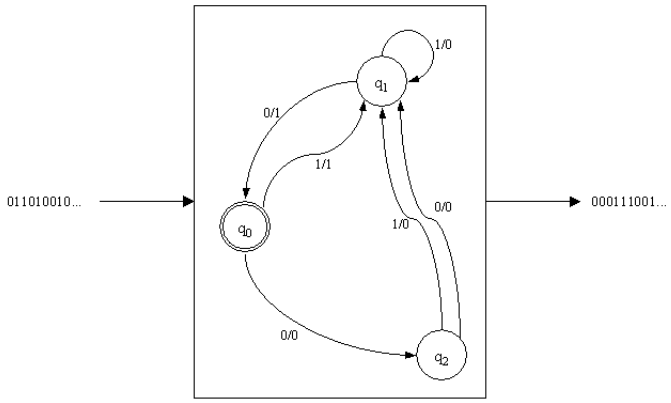

Fig. 5.1. An example of automaton. The vertices are the states, and the edges have labels of the form u/y, where u is an input symbol and y an output symbol. The initial state is q_0. Upon the input of the word 011010010, the automaton passes through the successive states q_0, q_2, q_1, q_1, q_0, q_1, q_0, q_2, q_1, q_0 and outputs the word 000111001.

called the *system automaton*, and a state (the *target state*). The problem is to find a word (the *control word*) of U^* that makes the automaton go from the initial state through the target state at least once.

The control word can be chosen to be the output of another automaton. We compare two kinds of strategies in order to solve a reachability problem: open-loop and feedback.

- *The open-loop strategy:* An inputless automaton O (the *control* automaton) is connected to the input of the target automaton, as indicated on the top of Figure 5.2.
- *The feedback strategy:* An automaton F (the *control* automaton) is connected in feedback (bottom of Figure 5.2). We can see it as a game played between the control automaton and the system automaton; the goal of the former is to force the latter to enter the target state. Let us mention a detail: for the process to be completely specified, the first input feeding the system automaton must be specified, too.

Thus an *open-loop solution* for an instance of the reachability problem is an inputless automaton that makes the system automaton evolve and reach the target state, when connected in open-loop as described above. A *feedback solution* is composed of a control automaton and a symbol of its output alphabet, that makes the system automaton reach the target state when the control automaton is connected in feedback with the system automaton and the specified symbol is given as initial input to the system automaton.

Our goal is to find for both strategies an estimate of the number of states of the control automaton needed to steer a given system automaton. We would also like to know whether open-loop control is more complex than feedback control. In other words, we ask the following question: does the possibility to make a measurement on the system lead to less complex controllers?

Given a solvable instance of the reachability problem, we naturally define the *open-loop complexity* of the instance as the number of states of the smallest open-loop solution, and the *feedback complexity* as the number of states of the smallest feedback solution.

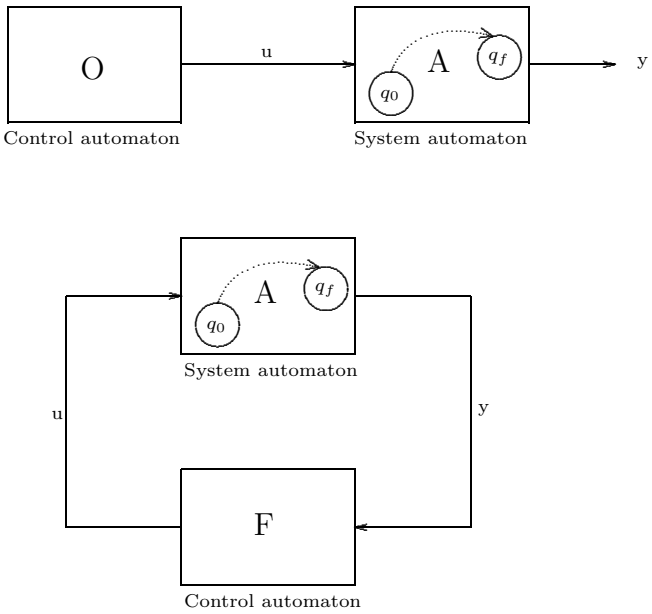

Fig. 5.2. We want to drive the system automaton A from q_0 to q_f. Above: The open-loop strategy. Note that no input has been drawn towards O, since only one input word is possible, up to the length (which may be seen as the time variable). Below: The feedback strategy. The control automaton F is supposed 'to play first': the first value of $u_0 \in U$ must be given as input to A, leading to an output $y_0 \in Y$, leading to a new output u_1 of F, etc. The control and system automata update their states alternatively. Hence there is a delay between the time at which y is output and the time at which the next value of u is given as input to the system.

For the sake of simplicity, we will focus on automata with binary input and binary output alphabets. However our results have immediate extension to alphabets of arbitrary fixed cardinality.

The following example shows that feedback can be much less complex than open-loop.

Example 3 *We want to solve the reachability problem for the n-state automaton represented on Figure 5.3 and the target state q_{n-1}. We assume that the word $u_0u_1 \dots u_{n-2}$ is not eventually periodic, except trivially. An inputless automaton producing this word as output must therefore have at least $n-1$ states. And the complexity of the open-loop strategy for this instance is $n-1$.*

But the particular form of the output function allows a one-state automaton put in feedback to solve the problem. Actually, even a zero-state automaton suffices! It is enough indeed to connect the output of the automaton directly to the input.

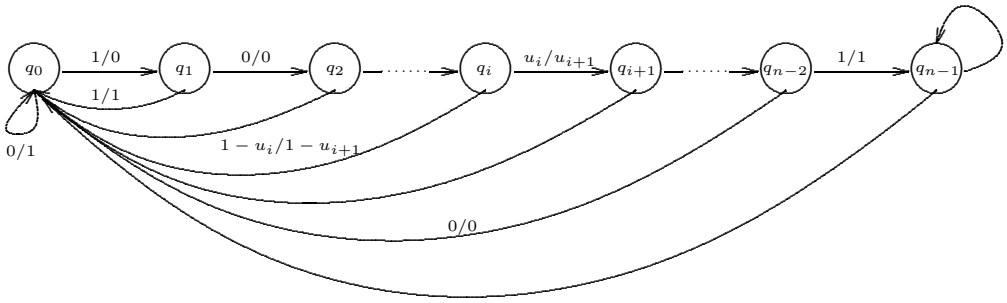

Fig. 5.3. An example of automaton for which the feedback strategy is much less complex than open-loop. The initial state is q_0, and we want to reach state q_{n-1}. The word $u_0u_1u_2 \dots u_{n-2}$ is not eventually periodic. In the open-loop strategy, an $(n-1)$-state control automaton is needed. In the feedback strategy, a zero-state automaton is enough.

As announced in the introduction, the gap of complexity between open-loop and feedback is quite small on average.

To make this statement precise, we must endow the set of n-state automata with a probability measure.

Let us compute the number of different n-state automata with input and output alphabets of cardinalities $|U|$ and $|Y|$ respectively. From every state $|U|$ edges must be drawn and for each of these there are n possible end states and $|Y|$ possible outputs. This leads to $(|Y|n)^{|U|n}$ possible graphs. For any of them there is a choice of n possible initial states, and so there are finally $n(|Y|n)^{|U|n}$ different automata. For simplicity, we suppose that the set of states is always $\{0, 1, \dots, n-1\}$.

Of course, many different automata are *isomorphic*, i.e., they are identical up to a permutation of the states. And even more automata define the same input-output function on words.

However, keeping the formalism of [TB73], we shall do averages on the class of $n(|Y|n)^{|U|n}$ different automata. For that purpose we endow this set with uniform probability measure. A random n-state instance of the reachability problem is composed of a random n-state automaton and a random state chosen uniformly among $\{0, 1, \ldots, n-1\}$. From now we will also restrict ourselves to the case $|U| = |Y| = 2$, as already mentioned.

Why the uniform distribution? This distribution reflects the fact that we have made no assumption on the origin or properties of the system automaton. The choice of the uniform distribution is consistent with the so-called 'maximum entropy principle', stating that an a priori probability distribution on a family of objects is best chosen as the less informative distribution, given the constraints that we know to be satisfied. Here 'less informative' is technically defined as 'maximizing the Shannon entropy'; see [Ros83]. As we have no a priori knowledge on the system automaton, the uniform distribution is certainly the less informative. Choosing a simple distribution also makes the problem tractable. The uniform distribution is the one considered in [TB73].

Of course, we might argue that 'real-life' systems are not all equally frequent or equally interesting, and most real-life situations would lead to a non-uniform distribution on system automata. For instance, the automaton might result from the quantization of a linear system. Depending on the method of quantization and the distribution of probability chosen for linear systems, we are likely to get a non-uniform distribution on finite automata. More generally, the quantization of any continuous discrete-time system should reflect the fact that 'close' states have 'close' images for the same input. Here again we expect that not all automata are equally represented. Similarly, if we quantize a continuous-time system with a small step of time, then the image of a state of the automaton is likely to be a 'close' state.

When we say that a statement is true for 'almost all n-state automata', we mean that it is true for 'a proportion of n-state automata that tends towards 1 as n increases'.

A *solvable* n-state instance of the reachability problem is made of an n-state system automaton and a target state that is reachable from the initial state. A random solvable instance is picked up by choosing at random an n-state automaton, and choosing at random a reachable state of the automaton. We now state the main result of the chapter.

Proposition 32 *Consider the uniform probability measure on n-state automata.*

There are constants c_1, c_2 such that for any large enough n, the expected number of states of the smallest open-loop control automaton that solves a solvable n-state instance reachability problem is between $c_1 \ln n$ and $c_2 \ln n$.

There are constants c_1, c_2 such that for any large enough n, the expected number of states of the smallest feedback control automaton that solves a solvable n-state instance reachability problem is between $c_1 \ln n / \ln \ln n$ and $c_2 \ln n$.

In other words, the expected complexity of open-loop is in $\Theta(\ln n)$, while the expected complexity of feedback is in $\Omega(\ln n / \ln \ln n)$ and $\mathcal{O}(\ln n)$.
Proof. Corollary 4 of Section 5.4 proves that the complexity for both strategies is in $\mathcal{O}(\ln n)$. Proposition 36 of Section 5.5 proves that the open-loop complexity is in $\Omega(\ln n)$. Proposition 38 of Section 5.5 proves that the complexity of feedback is in $\Omega(\ln n / \ln \ln n)$. □

Sections 5.3, 5.4 and 5.5 are devoted to filling the details of the proof of Proposition 32, and may be skipped in a first reading.

5.3 Solvable instances of the reachability problem

It is clear that the expected complexity can be computed only over solvable instances. The following proposition essentially says that more than one third of the instances of the reachability problem are solvable. Recall that we consider only automata with binary input and output alphabets.

Proposition 33 *For any $\alpha < 1/e = 0.36...$ and any large enough n, a random n-state instance of the reachability problem is solvable with probability at least α. More precisely, for almost all n-state automata at least $\lceil \alpha n \rceil$ states are reachable from the initial state.*

Proof. Let $Q = \{0, 1, \ldots, n-1\}$ the set of states of a random automaton.

A subset $R \subseteq Q$ is *invariant* if, starting from a state of R, it is impossible to leave R. In other words, no edge leaves R.

We are going to prove that in almost all n-state automata, there is no invariant subset of $\lfloor \alpha n \rfloor$ states or less, for $0 < \alpha < 1/e$. As the set of states that are reachable from the initial state is an invariant set, this is enough to prove the proposition.

The probability for any set R of r states to be invariant is $(\frac{r}{n})^{2r}$, since $2r$ edges start from R, and each of them arrive in R with probability r/n.

For any $R \subseteq Q$, consider the random variable I_R that is equal to 1 if the set R is invariant and 0 otherwise. The expectation of I_R is the probability of the event $I_R = 1$, which is $(r/n)^{2r}$. The probability that there is a invariant set of size at most αn is equal to

$$\mathbb{P}\Big(\sum_{\substack{R\subseteq Q\\ 0<\mathrm{card}(R)\leq \alpha n}} I_R > 0\Big),$$

where $\mathbb{P}$ denotes the probability of an event.

For any nonnegative integer-valued random variable X, Markov's inequality

$$\mathbb{P}(X > 0) \leq \mathbb{E}X$$

holds, where $\mathbb{E}$ is the expectation. Indeed, $\mathbb{E}X = \sum_{i>0} i\mathbb{P}(X = i) \geq \sum_{i>0} \mathbb{P}(X = i) = \mathbb{P}(X > 0)$.

Here we take $X = \sum_{0<\mathrm{card}(R)\leq \alpha n} I_R$. The probability that there is an invariant set of size at most αn is

$$\begin{aligned}\mathbb{P}(X > 0) &\leq \mathbb{E}X\\ &= \sum_{0<\mathrm{card}(R)\leq \alpha n} \mathbb{E}I_R\\ &= \sum_{r=1}^{\lfloor \alpha n\rfloor} \binom{n}{r}(r/n)^{2r}.\end{aligned}$$

If we prove that this last quantity converges to 0 as $n \to \infty$, then we conclude from Markov's inequality that almost all automata of size n have have no invariant set of states of size at most αn. This argument is an example of the so-called *first moment method*.

It remains to prove that $\lim_{n\to\infty} \sum_{r=1}^{\lfloor \alpha n\rfloor} \binom{n}{r}(\frac{r}{n})^{2r} = 0$. For every n and for every term of the series, the following relations hold:

$$\begin{aligned}\binom{n}{r}(r/n)^{2r} &\leq \frac{r^{2r}}{r!n^r}\\ &\leq \frac{e^r r^r}{n^r} \leq (e\alpha)^r.\end{aligned}$$

We used the inequalities $\binom{n}{r} \leq \frac{n^r}{r!}$ and $r^r/r! \leq \sum_{i\geq 0} r^i/i! = e^r$, and the fact that $r \leq \alpha n$.

Let us consider for every n the function

$$f_n : r \mapsto \begin{cases} \binom{n}{r}(\frac{r}{n})^{2r} & \text{if } 0 < r \leq \alpha n;\\ 0 & \text{otherwise.}\end{cases}$$

We want to prove that $\lim_{n\to\infty} \sum_r f_n(r) = 0$. But the following properties are verified:

- The sequence $(f_n)_n$ converges pointwise to the constant function 0, since $f_n(r) \leq \frac{e^r r^r}{n^r}$.

- Every f_n is bounded by the function $g : r \mapsto (e\alpha)^r$, and g is summable: $\sum_{r\geq 0}(e\alpha)^r = \frac{1}{1-e\alpha}$.

From Lebesgue's dominated convergence theorem (see Section 2.2), this is enough to conclude that the sequence $(\sum_r f_n(r))_n$ converges to $\sum_r 0 = 0$. □

We don't know how much the ratio $1/e$ can be improved. It might be the case that all states are reachable for almost all automata. As far as we know this is an open problem.

5.4 An upper bound on complexity

If a state of an n-state automaton is reachable from the initial state, then it is so with an input word of length at most n; hence the complexity of feedback and open loop are both bounded by n. On average much more can be said.

Proposition 34 *There is a constant c such that for almost all n-state automata, if a state is reachable from another state, then it is reachable with an input word of length at most $c \ln n$.*

We do not prove this involved result, due to Trakhtenbrot and Barzdin; the interested reader is referred to [TB73]. The idea is to prove that the graph of the automaton around the initial state looks very much like a directed tree, that contains most of the reachable states. This proposition may be seen as stating a 'small world property' for automata. Small world properties roughly say that any two points of a graph are always linked by a path of short length, and are proved to be almost always true for large classes of random graphs. They are a mathematical expression of the 'six degrees of separation' theory, stating that any two humans on earth are very likely to be connected to each other through a chain of no more than five intermediate acquaintances. Small world phenomena are observed in numerous social networks; see [AB02] for a survey on related questions.

Corollary 4 *There is a constant c such that for almost all n-state instances of the reachability problem, there are feedback and open-loop solutions of $c \ln n$ states or less. In particular, the expected complexity of a solvable instance is in $\mathcal{O}(\ln n)$ for both strategies.*

5.5 Lower bounds on complexity

Let us a take a random instance of the reachability problem of size n, i.e., a random n-state automaton (with binary input and output alphabets) with a

random target state. Take a random p-state inputless automaton (with binary output alphabet). We will denote by $\mathcal{P}^{ol}(n,p)$ the probability that the random p-state control automaton solves the random n-state instance of the reachability problem in open-loop.

Now take a random p-state automaton (with binary input and output alphabets) and a random initial bit. We will denote by $\mathcal{P}^{f}(n,p)$ the probability that the random p-state control automaton together with the random initial bit solves the random n-state instance in feedback.

In this section, we compute estimates of these quantities that allow us to give lower bounds on complexity. First of all, a technical lemma:

Lemma 1 *There is a constant $c > 0$ such that for any large enough p, any $n \geq p^3$ and $k = 2p\sqrt{n \ln n}$ we have*

$$\frac{n^p e^{-k}}{(1 - \frac{k+p}{pn})^{pn-k}} \leq c\frac{1}{n}.$$

Proof. Taking the logarithm:

$$\begin{aligned}
\ln \frac{n^p e^{-k}}{(1 - \frac{k+p}{pn})^{pn-k}} &= -k + p \ln n - \big(pn - k\big) \ln(1 - \frac{k+p}{pn}) \\
&= -k + p \ln n + \big(pn - k\big) \frac{k+p}{pn} + \frac{1}{2} \big(pn - k\big) \big(\frac{k+p}{pn}\big)^2 \\
&\quad + \mathcal{O}\Big(\big(pn - k\big) \big(\frac{k+p}{pn}\big)^3\Big)
\end{aligned}$$

We used the expansion $\ln(1+x) = x - x^2/2 + \mathcal{O}(x^3)$. A little calculation yields:

$$\begin{aligned}
\ln \frac{n^p e^{-k}}{(1-\frac{k+p}{pn})^{pn-k}} &= -k + p\ln n + k + p - \frac{k(k+p)}{pn} + \frac{1}{2}\frac{(k+p)^2}{pn} \\
&\quad -\frac{1}{2}\frac{k(k+p)^2}{(pn)^2} + \mathcal{O}\big(pn\frac{k^3}{(pn)^3}\big) \\
&= p(\ln n + 1) + \frac{(p-k)(k+p)}{2pn} - \frac{1}{2}\frac{k(k+p)^2}{(pn)^2} + \mathcal{O}\big(\frac{k^3}{(pn)^2}\big) \\
&= p(\ln n + 1) + \frac{p^2}{2pn} - \frac{k^2}{2pn} - \frac{1}{2}\frac{k(k+p)^2}{(pn)^2} + \mathcal{O}\big(\frac{p\ln^{3/2} n}{\sqrt{n}}\big) \\
&= -p(\ln n - 1) + \frac{p}{2n} - \frac{1}{2}\frac{k(k+p)^2}{(pn)^2} + \mathcal{O}\big(\frac{p\ln^{3/2} n}{\sqrt{n}}\big) \\
&\le -p(\ln n - 1) + \frac{p}{2n} + \mathcal{O}\big(\frac{p\ln^{3/2} n}{\sqrt{n}}\big)
\end{aligned}$$

We used the fact that $\ln(1+x) = x - x^2/2 + \mathcal{O}(x^3)$. The last two terms of the last line are bounded by a constant, provided that $n \ge p^3$. We can also suppose $p - 1 \ge 1$. Thus

$$\frac{e^{-k}}{(1-\frac{k+p}{pn})^{pn-k}} \le c\frac{1}{n},$$

for some $c > 0$. □

Finally, a Stirling-like formula taken from [DK03].

Lemma 2 *For all $n \ge 1$,*

$$e(\frac{n}{e})^n \le n! \le en(\frac{n}{e})^n$$

Proof. From $\lfloor x \rfloor \le x \le \lceil x \rceil$, we have $\ln\lfloor x \rfloor \le \ln x \le \ln\lceil x \rceil$. Integrating between 1 and n, we find:

$$\sum_{i=1}^{n-1} \ln i \le [x\ln x - x]_1^n \le \sum_{i=2}^{n} \ln i,$$

which is equivalent to

$$\ln(n-1)! \le n\ln n - n + 1 \le \ln n!.$$

The result follows almost immediately by exponentiation. □

5.5.1 A lower bound for open-loop complexity

We may obtain the following bound on $\mathcal{P}^{ol}(n,p)$:

Proposition 35 *For any large enough p and any $n \geq p^3$, the probability $\mathcal{P}^{ol}(n,p)$ for a random inputless p-state automaton to solve a random n-state instance of the reachability problem in open-loop is bounded by*

$$\mathcal{P}^{ol}(n,p) \leq 3p\sqrt{\frac{\ln n}{n}}.$$

Proof. We fix an arbitrary inputless control automaton of p states, and plug the output to the input of a random n-state automaton.

In a first step we give an upper bound on the probability for the path described in the random target automaton to pass through at least $k+1$ of the n states. In a second step we use Lemma 1 to compute a bound on $\mathcal{P}^{ol}(n,p)$.

Step 1 To explore at least $k+1$ different states, we must explore at least k different edges. We may express the probability to explore at least k edges as $\prod_{i=1}^{k} P_i$, where P_i is the probability that at least i edges are discovered, knowing that at least $i-1$ edges are discovered.

Each time a new edge is being explored, the end state of the edge is randomly chosen among the n states. If this new edge occurs after l steps (i.e., when the path already drawn has length l), and if less than k edges are explored at that moment, then at most $n - \lfloor l/p \rfloor$ are allowed for the end state of the edge (in order not to loop before having seen k states). This is because if the pair (state of the control automaton, state of the system automaton) is the same at two steps, then the system enters a loop, and because the succession of states of an inputless automaton is eventually periodic with period at most p. Thus at least $\lfloor l/p \rfloor$ states are 'forbidden'. Moreover, the ith new edge is discovered after a length $l \geq i$. This shows that $P_i \leq \frac{n-\lfloor i/p \rfloor}{n}$.

Thus the probability to explore at least k edges is less than the following product of k factors:

$$\overbrace{\frac{n}{n}\cdots\frac{n}{n}}^{p}\overbrace{\frac{n-1}{n}\cdots\frac{n-1}{n}}^{p}\cdots\cdots\overbrace{\frac{n-\lfloor k/p \rfloor}{n}\cdots\frac{n-\lfloor k/p \rfloor}{n}}^{\leq p}, \tag{5.1}$$

where each factor —except maybe the last— is repeated p times. Dropping the last factor, this product is smaller than:

$$\begin{aligned}\Big(\frac{n!}{(n-\lfloor k/p \rfloor)!n^{\lfloor k/p \rfloor}}\Big)^p &\leq \Big(n\frac{e^{-\lfloor k/p \rfloor}}{(1-\frac{1}{n}\lfloor\frac{k}{p}\rfloor)^{n-\lfloor k/p \rfloor}}\Big)^p \\ &\leq n^p\frac{e^{-k'}}{(1-\frac{k'+p}{pn})^{pn-k'}},\end{aligned}$$

where $k' = k - p$ (implying $k'/p < \lfloor k/p \rfloor \leq k'/p + 1$). Lemma 2 has been used to derive the first inequality.

Step 2 Let k' be equal to $2p\sqrt{n \ln n}$. Then Lemma 1 ensures that

$$\frac{e^{-k'}}{(1 - \frac{k'+p}{pn})^{pn-k'}} \leq c\frac{1}{n},$$

for some $c > 0$, if p is large enough and $n \geq p^3$. Thus the probability to explore more than $2p\sqrt{n \ln n} + p + 1$ states is at most c/n. The expected number of states visited by the control automaton in the system automaton is therefore at most

$$\left(1 - c\frac{1}{n}\right)\left(2p\sqrt{n \ln n} + p + 1\right) + c\frac{1}{n}n \leq 3p\sqrt{n \ln n}$$

for p large enough and $n \geq p^3$. As any of the n states may be chosen as the target state with equal probability, we conclude that $\mathcal{P}^{ol}(n, p) \leq 3p\sqrt{\ln n/n}$ for p large enough and $n \geq p^3$. □

In fact, the proof shows that the result holds not only for a random control automaton but for any fixed control automaton.

We now derive a lower bound on open-loop complexity.

Proposition 36 *There is a c such that for any large enough n the expected complexity of the open-loop strategy for an n-state solvable instance of the reachability problem is at least $c \ln n$.*

Proof. The behavior of an inputless automaton is completely characterized by the infinite word it produces as output. This output word is eventually periodic.

The number of inputless p-state automata generating different output sequences is $p2^p$. Indeed, p is the sum of lengths of the nonperiodic part and the period, and there are p possibilities for the length of the period. This also includes the control automata of less than p states. Let $\{O_1, O_2, \ldots O_{p2^p}\}$ be a family of p-state inputless automata covering all possible output sequences of all inputless automata of at most p states. Then the probability that an n-state instance of the reachability problem is solved by at least one p-state open-loop automaton is the probability that an n-state instance of the reachability problem is solved by at least one automaton among $\{O_1, O_2, \ldots O_{p2^p}\}$, which is bounded by the sum over i of probabilities that an n-state instance of the reachability problem is solved by O_i.

So, using Proposition 35, the probability that an n-state instance of the reachability problem is solved by at least one p-state open-loop automaton is (provided that $n \geq p^3$ and p is large enough) bounded above by

$$p2^p 3p\sqrt{\frac{\ln}{n}}.$$

Assume that this quantity is lower than 1/6:

$$2^p 3p\sqrt{\frac{\ln n}{n}} \leq \frac{1}{6},$$

which may be written as

$$3p^2 2^p \leq \frac{1}{6}\sqrt{\frac{n}{\ln n}}.$$

This is satisfied if

$$p \leq c_0 \ln n,$$

for some $c_0 > 0$. This condition is stronger than $n \geq p^3$, for p large enough.

Thanks to Proposition 33, we know that for n large enough, at least one third of the n-state instances of the reachability problem are solvable. If $p \leq c_0 \ln n$, at most one sixth of the n-state instances are solved by open-loop control automata of p states or less.

Hence at least one half of the n-state solvable instances are not solved by any open-loop control automaton of at most p states. Consequently, the average complexity of an n-state solvable instance is at least $\frac{c_0}{2} \ln n$. So the result follows with $c = c_0/2$. □

5.5.2 A lower bound for feedback complexity

We now adapt the arguments of the preceding subsection to the case of feedback.

Proposition 37 *For any large enough p and for any $n \geq (2p)^3$, the probability $\mathcal{P}^f(n,p)$ for a random p-state automaton and a random first bit to solve in feedback a random n-state instance of the reachability problem is bounded by*

$$\mathcal{P}^f(n,p) \leq 6p\sqrt{\frac{\ln}{n}}.$$

Proof. First we fix an arbitrary feedback control automaton F.

The main idea of the proof is the same as in the proof of Proposition 35. There we used the fact that if the system automaton is twice in the same state while the open-loop control automaton is also in the same state, then the whole system falls into a loop and will not discover new states. The case of the feedback strategy is slightly more complicated. Indeed, knowing the states of the control and system automata is not sufficient to determine the evolution of the system.

Suppose that the system automaton A 'has just played' (remember the game-theoretic terminology) and is now in state q, emitting a symbol y. The

control automaton F is about to play and is in state s. Now we have enough information to determine the evolution of the system. If during two different steps, A is in state q, outputs the symbol y and F is in state s, then the whole system enters a loop. Since y can take two values, the system automaton and the control automaton can only be twice in the same states if we still hope to discover new states of A. Heuristically, we thus expect the same result as in Proposition 35, except that that p becomes $2p$. The rest of the proof rigorously establishes this fact.

Let the states of F be denoted by $s_1, \ldots, s_p$. At every step of time, when F has just played, and for every state s_i, we define

- a_i as the number of states of A that have been reached exactly once while F was in state s_i;
- b_i as the number of states of A that have been reached exactly twice while F was in state s_i;
- r_i as $n - a_i - b_i$; as long as the whole system has not entered a loop, r_i is the number of states of A that have never been reached while F was in s_i.

At the beginning of the process, $r_i = n$ and $a_i = b_i = 0$ for all states s_i. At every step of time, if F is in s_i, then a_i, b_i, r_i must be updated. If a state of A is reached for the first time while F is in s_i, then a_i is increased ('$a_i := a_i + 1$') and r_i is decreased. If a state of A is reached for the second time while F is in s_i, then a_i is decreased and b_i is increased. If a state of A is reached for the third time or more, then we will not discover any new state of A anymore.

In the first two cases, the quantity $\frac{a_i}{2} + r_i$ decreases by $\frac{1}{2}$.

We now want to estimate the probability that at least $k+1$ states of the system automaton are visited when F is connected in feedback to it. To explore at least $k+1$ different states of the system automaton, we must explore at least k different edges. Every time a new edge is being explored, the end state of the edge is chosen randomly among the n states. Suppose that after l steps (i.e., when the path already drawn has length l), less than k edges have been reached, the system automaton is in state $\tilde{q}$, F has just sent the output u and the edge starting from $\tilde{q}$ labelled by u has never been used: we must choose an end state $q = \Delta_A(\tilde{q}, u)$ for this new edge, as well as an output value $y = \Gamma_A(\tilde{q}, u)$ for this edge.

Suppose in addition that the current state of F is s_i. The end state of the new edge is to be chosen among three kinds of states:

- Those that have been reached once while F was in state s_i. There are a_i of them. If one of these states is chosen, then the probability for the trajectory to fall into a loop is at least $1/2$, because one of the two possible values for $\Gamma_A(\tilde{q}, u)$ is "forbidden" (remember we do not want to loop before having explored k states).

- Those that have been reached (exactly) twice while F was in state s_i. There are b_i of them. They may not be chosen as end state of the edge, since the whole system would loop.
- Those that have not been explored yet when F was in state s_i. There are r_i of them. They may be chosen without restriction.

So the probability to choose a 'good' end state for a new edge (i.e., not to fall into a loop) is at most $\frac{1}{2}\frac{a_i}{n} + \frac{r_i}{n}$. Remember that $\frac{1}{2}a_i + r_i$ is equal to $n - \frac{1}{2}$(number of times that F has been in state s_i).

Call t_i the number of occurrences of s_i while exploring a new edge, i.e., the number of times that F was in s_i when we took an edge of the path for the first time. If k edges at least have been explored, then $t_1 + \cdots + t_p \geq k$. Knowing that k edges at least have been explored, we can encode with a natural integer every possibility of repartition of the number of edges into $t_1, \ldots, t_p$.

The probability of discovering at least k new edges, conditional to the number encoding the repartition into $t_1, \ldots, t_p$, is thus bounded above by

$$\prod_{1 \leq i \leq p} \frac{n}{n} \frac{n - \frac{1}{2}}{n} \frac{n - \frac{2}{2}}{n} \cdots \frac{n - \frac{t_i}{2}}{n}. \tag{5.2}$$

The probability of discovering at least k new edges is this product, averaged over all possible repartitions into $t_1, \ldots, t_p$.

On the other side, a product of this form, under the constraint $t_1 + \cdots + t_p \geq k$ takes its maximum value when $t_1, \ldots, t_p$ are all equal to $\lfloor k/p \rfloor$ or $\lceil k/p \rceil$, as some elementary calculus can show.

Hence the probability of discovering at least k states in A is at most

$$\Big(\frac{n}{n} \frac{n - \frac{1}{2}}{n} \frac{n - \frac{2}{2}}{n} \cdots \frac{n - \lceil \frac{k}{p} \rceil \frac{1}{2}}{n}\Big)^p,$$

itself bounded by

$$\Big(\frac{n}{n}\frac{n}{n} \frac{n-1}{n} \frac{n-1}{n} \cdots \frac{n - \lceil \frac{k}{2p} \rceil}{n} \frac{n - \lceil \frac{k}{2p} \rceil}{n}\Big)^p,$$

where each factor is repeated twice.

Comparing with Equation 5.1, we see that nothing has changed, except that p has become $2p$. So the end of the proof remains unchanged in its principle: the conclusion is that the probability that $\mathcal{P}^f(n,p)$ is less than $6p\sqrt{\frac{\ln n}{n}}$. □

Again, the proof holds not only for a random feedback control automaton but for any feedback control automaton. Thus:

Proposition 38 *The expected number of states of the smallest feedback control automaton that solves an n-state solvable instance of the reachability problem is, for some constant c and any large enough n, at least $c\frac{\ln n}{\ln \ln n}$.*

Proof. The number of different p-state automata is $p(2p)^{2p}$ and there are two possible initial bits. So, using the preceding proposition, the probability that a random n-state instance of the reachability problem is solved by at least one p-state feedback automaton is, for p large enough and $n \geq (2p)^3$, bounded above by:

$$2p(2p)^{2p}6p\sqrt{\frac{\ln n}{n}}.$$

We want this quantity to be lower than $1/6$:

$$2p(2p)^{2p}6p\sqrt{\frac{\ln n}{n}} \leq \frac{1}{6}.$$

This is equivalent to

$$3(2p)^{2p+2} \leq \frac{1}{6}\sqrt{\frac{n}{\ln n}},$$

or

$$(2p+2)\ln(2p) \leq \frac{1}{2}(\ln n - \ln\ln n) - \ln 18.$$

This is satisfied if

$$p \leq c_0 \frac{\ln n}{\ln\ln n},$$

for some $c_0 > 0$ and n large enough. Indeed, this condition implies

$$\begin{aligned}
(2p+2)\ln(2p) &\leq 3p\ln(p/c_0) \\
&\leq 3c_0\frac{\ln n}{\ln\ln n}(\ln\ln n - \ln\ln\ln n) \\
&= 3c_0(\ln n - \frac{\ln n \ln\ln\ln n}{\ln\ln n}) \\
&\leq 3c_0(\ln n - \ln\ln n) \\
&\leq \frac{1}{2}(\ln n - \ln\ln n) - \ln 18,
\end{aligned}$$

for any large enough p. This condition is stronger than $n \geq (2p)^3$.

Thanks to Proposition 33, we know that for n large enough, at least one third of the n-state instances of the reachability problem are solvable. If $\ln n/\ln\ln n$, it means that at least half of the solvable n-state instances are not solved by any p-state open-loop control automaton. Hence the expected complexity of an n-state instance is at least $\frac{c_0}{2}\ln n$. So the result follows with $c = c_0/2$. □

5.6 Conclusions

We have studied finite automata as input-output systems, and have analyzed the complexity needed to control them. It follows from our results that making measurements on the output of a system is not very useful for automata with a completely random structure. Of course, feedback cannot be more complex than open-loop, and we gave an example where it is actually much less complex.

Thus, we are led to the conclusion that feedback is useful to gain complexity only in connection with a particular structure of the automaton. Future work could be devoted to finding classes of particular systems for which feedback is particularly interesting. For instance, it would be natural to see what happens if the automaton results from the quantization of a continuous linear system. If we dare extrapolate this to a very general situation by considering that any input-output system can be approximated by an automaton, we could conclude that a system is controllable in feedback with a relatively small controller only if the system is highly structured; for instance our eyes do help us to navigate in an environment only if there is a great amount of order in it.

The introduction of some noise or nondeterminism in the model is also a logical direction of research. The need for some robustness with respect to noise is expected to give feedback the advantage over open-loop.

Finally, it can be argued that we did not really address the introductory example of Section 1.3 (describing a path in a city by a set of instructions in a concise way), since in that case the size of the output alphabet (number of streets, for instance) is not fixed, but depends on the size of the automaton (number of crossroads). For what kind of dependence between sizes of input alphabet, output alphabet, and number of states does Proposition 32 remain valid?

It can also be argued that in control theory one often need to remain in the target state once we reach it, which we do not require in what we called the reachability problem. Are the results of this chapter robust to modifications of the objective?

6

An Optimal Quantized Feedback Strategy for Scalar Linear Systems

6.1 Introduction

In Section 1.4 we explained the problem investigated by Fagnani and Zampieri [FZ03, FZ04a, FZ05], namely, how to control the system $x := ax + u$ (with $a > 1$) with a memoryless feedback map that has a finite range, in order to contract the interval $[-1, 1]$ to $[-1/C, 1/C]$ (for $C > 1$). Input-output systems and feedback maps are rigorously defined in Section Motivation and background are also given in Section 1.4. In Section 6.2 we give a brief overview of the results of [FZ03, FZ04a, FZ05].

In Section 6.3 we study a different, but related, problem. Finitely many maps act on a space endowed with a probability measure. All maps preserve the measure. We show that when it is possible to concentrate all the space to a given subset, it is possible to do so with a memoryless feedback. An information-theoretic lower bound on the time needed to reach this objective is derived. A link with Maxwell's demon and the second law of thermodynamics is also pointed out.

When a is an integer, the map $f : [-1, 1] \to [-1, 1] : x \mapsto ax + u \mod 2$ preserves measure. By this way we can use the bound derived in Section 6.3 to get a bound on the system $x \mapsto ax + u$. Thus the results of Section 6.3 apply.

We exhibit in Section 6.4 a family of feedback maps for $f(x, u) = ax + u \mod 2$ and a integer. The idea is to write the expansion of x in base a, and choose a feedback map that eventually transforms the first digits of the expansion to 0s. As it nearly achieves the information-theoretic bound, it is close to be optimal. These feedback maps can be converted into feedback maps for the original system $x \mapsto ax + u$. The subsets on which the feedback map is constant are not intervals. In light of this, the assumption of 'intervalness' made in [FZ03, FZ04a, FZ05] is discussed. The final section draws some conclusions for future work.

6.2 Control on scalar linear systems

To get familiar with the problem, we review some of the results of [FZ03, FZ04a, FZ05] on the problem of quantized memoryless feedback on scalar linear maps. Given numbers N, C and T, we wish to find a partition $[-1,1]$ into N intervals $X_1,\dots,X_N$ and a feedback map $u=\Gamma(x)$ that is constant on every X_i such that every point of $[-1,1]$ is driven to $[-1/C,1/C]$ in average time T (for the uniform distribution on $[-1,1]$). Basics of probability and measure theory are covered in Section 2.2.

The *deadbeat* feedback map is $u=-\frac{(2j+1)}{C}$ if $x\in]\frac{2j}{aC},\frac{2(j+1)}{aC}]$. It is a piecewise constant approximation of $u=-ax$ and sends every quantization interval $]\frac{2j}{aC},\frac{2(j+1)}{aC}]$ into $[-1/C,1/C]$. Thus every point of $[-1,1]$ is driven to $[-1/C,1/C]$ in one step of time. We can check that $N=2\lceil|a|(C-1)/2\rceil$ and $T=1-1/C$.

If $|a|(C-1)/2$ is an integer (say n), then after one step with deadbeat feedback map, the state is uniformly distributed in $[-1/C,1/C]$. In the same time, we can apply the same feedback map, scaled by a factor $1/C$, that drives $[-1/C,1/C]$ to $[-1/C^2,1/C^2]$. We can repeat this operation several times: this is the *nested* feedback map. Thus for any integers n and k, we can construct a feedback map with parameters $C=(2n/|a|+1)^k$, $N=2kn$ and $T=k\frac{1}{1+|a|/2n}$. Here k is the number of nestings, and for $k=1$ we get back the deadbeat feedback map.

If we fix k, then we get a family of feedback maps for which $T\simeq k=\Theta(1)$ and $N\simeq|a|TC^{1/T}$. If we fix n we get a family of feedback maps for which $T=\Theta(\log C)$ and $N=\Theta(\log C)$.

The *chaotic* feedback map is such that the state x is sent to $ax \mod 2$ if $|x|\in[1/C,1]$ and to $ax \mod 2/C$ if $x\in[-1/C,1/C]$. By '$ax \mod 2$' we mean that we add or subtract 2 to ax if necessary, in order to stay in $[-1,1]$. Similarly, '$ax \mod 2/C$' ensures that the interval $[-1/C,1/C]$ is invariant. The map $x\to ax \mod 2$ in $[-1,1]$ is such that almost every point visits any interval. Thus the chaotic feedback map drives almost every state of $[-1,1]$ to $[-1/C,1/C]$, where it stays forever.

If $|a|$ is integral then we can apply the chaotic feedback map to $[-1/C,1/C]$ itself, in order to contract it to $[-1/C^2,1/C^2]$. Repeating this operation to contract to $[-1/C^k,1/C^k]$, we get the *nested* chaotic feedback. One can prove that for any integers n and k, we have the parameters $C=|a|^{nk}$, $N\simeq k|a|$ and $T\simeq k|a|^n$. Thus k is the number of nestings and n is such that the contraction ratio of one degree of nesting is $|a|^n$. If we fix k, then we have a family of feedback maps such that $N\simeq k|a|=\Theta(1)$ and $T\simeq\frac{N}{|a|}C^{N/|a|}$. If we fix n, then $N=\Theta(\log C)$ and $T=\Theta(C)$.

These feedback maps provide examples of realizable triples of parameters N,C,T. Lower bounds are also derived, proving that some triples are nonrealizable. The main of them are listed below.

- *For any $r > 0$, there exists $s > 0$ such that $N \leq r \log_2 C$ implies $\lceil T \rceil \geq s \log_2 C$;*
- *For any $r > 0$, there exists $s > 0$ such that $\lceil T \rceil \leq r \log_2 C$ implies $N \geq s \log_2 C$;*
- *There exist $K_1 > 0, \beta_1 > 0, C_1 > 1$ such that $C \geq C_1$ and $N \leq \beta_1 \log_2 C$ imply $T \geq K_1 N C^{1/N}$;*
- *There exist $K_2 > 0, \beta_2 > 0, C_2 > 1$ such that $C \geq C_2$ and $\lceil T \rceil \leq \beta_2 \log_2 C$ imply $N \geq K_1 \lceil T \rceil C^{1/\lceil T \rceil}$.*

The proofs are quite involved and use tools of symbolic dynamics: given a feedback map and a partition of the space, the combinatorics of language induced by the partition (in a sense similar to Section 2.6) is studied.

These bounds prove that the particular families of feedback maps (for k or n fixed) are optimal: we cannot significantly simultaneously improve all three parameters N, C, T.

6.3 Control on measure-preserving maps

We now study a slightly different situation than the problem studied by Fagnani and Zampieri, and see how it relates to the problem of memoryless quantized feedback on scalar linear systems. We consider an input-output system

$$x_{k+1} = f(x_k, u_{k+1}),$$

where for any $k \geq 0$, x_k is supposed to lie in a probability space X, endowed with the probability measure μ. The control input u_{k+1} takes values in a set U. We suppose that for any u, the map $f(., u) : X \to X$ is *measure-preserving*, i.e., the preimage of a measurable set is a measurable set of same measure.

If we control the system in open-loop, i.e., we apply a sequence $u_1 u_2 \ldots$ that does not depend on the initial state x_0, then the probability distribution of x_i is equal to μ, thus it is not possible to concentrate the state to a small region of the space.

We now define a (memoryless) *quantized feedback map* in this context. We partition the set X into finitely many measurable subsets $X_1, X_2, \ldots, X_N$ called *quantization subsets*, and we choose a map $u_{k+1} = \Gamma(x_k)$ that is constant on every set X_i. In other words, the only information available to choose the next input is whether the present state of the system is in X_1, X_2, ... or X_N. The objective is that almost every initial condition eventually reaches a specified subset of measure $1/C$ and stays in this subset forever. The parameter C is again called the *contraction ratio*.

Someone observing the sequence of quantization subsets $X_{i_0}, X_{i_1}, X_{i_2}, \ldots$ (where $x_k \in X_{i_k}$) should also know when the control objective is fulfilled. From now, when we talk of the random variable t as the *first entry time*,

we mean the first time that we know for sure, given the past observations, that the state lies in the prescribed set of measure $1/C$ and will not leave it. More precisely, observing the sequence $X_{i_0}, X_{i_1}, X_{i_2}, \ldots, X_{i_{t-1}}$ is enough to deduce that the goal is reached (with probability 1), but not the sequence $X_{i_0}, X_{i_1}, X_{i_2}, \ldots, X_{i_{t-2}}$. Thus the set made of all such observable finite sequences $X_{i_0}, X_{i_1}, X_{i_2}, \ldots, X_{i_{t-1}}$ is a *prefix* set, meaning that no finite sequence is a prefix of another.

By T we denote the expected value of t, with respect to measure μ.

The problem of scalar memoryless quantized feedback can be expressed in this formalism for the case when a is integer. Indeed, take the map $f : [-1,1] \times \mathbb{R} \to [-1,1] : x \mapsto ax + u \mod 2$. As -1 is then equivalent to 1 modulo 2, it seems at first sight that we should work with $]-1,1]$ or $[-1,1[$ instead of $[-1,1]$; but since these three intervals only differ by a set of measure zero, it makes no difference at the end.

If a is an integer then it is well known that $f(.,u)$ preserves Lebesgue measure for every u. Indeed, the preimage of a small interval of length δ is composed of $|a|$ intervals of length $\delta/|a|$; as the measurable sets are generated by small intervals, the preimage of any measurable set is a measurable set of same measure. If we fix finitely many values $u_1, \ldots, u_N$, then we must choose at every step which of the N measure-preserving maps $f(.,u_1), \ldots, f(.,u_N)$ to apply.

Moreover, given a feedback map on f with a partition of $[-1,1]$ into N subsets, there is a corresponding feedback map for the map $(x,u) \mapsto ax + u$ with a partition into at most $(|a|+1)N$ subsets leading to the same closed-loop map. Indeed, consider the map "If x is in X_i then apply $x \mapsto ax + u_i \mod 2$", where $i = 1, \ldots, N$. It is equal to the map "If x is in $X_{i,j}$ then apply $x \mapsto ax + u_i + 2j$", where $X_{i,j} = [(-1-u_i-2j)/a, (1-u_i-2j)/a[\cap[-1,1]$ is an interval on which $x \mapsto ax + u_i$ is continuous. The equality between these two closed-loop maps is straightforward. As there are at most $|a|+1$ values of j for which $[(-1-u_i-2j)/a, (1-u_i-2j)/a[\cap[-1,1]$ is nonempty, there are at most $(|a|+1)N$ sets $X_{i,j}$.

In the following, when we write N we mean the number of quantization subsets for $f(x,u) = ax + u \mod 2$.

In [FZ03, FZ05] it is supposed that the target interval $[-1/C, 1/C]$ is itself the union of one or several quantization intervals, and is invariant when the feedback map is applied (i.e., almost all points of $[-1/C, 1/C]$ remain in $[-1/C, 1/C]$). Thus the first time at which we know for sure that the system has reached the target interval and will remain in this interval is exactly the first time at which the target interval is reached.

As already explained, there is a crucial difference between the problem described in Sections 1.4 and 6.2, which is taken from [FZ03, FZ05], and the control on $f(x,u) = ax + u \mod 2$, following the setting of the present section. Indeed, in [FZ03, FZ05] the quantization subsets are required to be intervals,

whereas in the setting of this section all measurable subsets are allowed. As shown below, if we drop the hypothesis of 'intervalness', then the inequalities listed at the end of 6.2 do not hold.

6.3.1 When is memoryless quantized feedback possible?

Let us come back to the general problem. We have N maps $f_1, f_2, \ldots, f_N$ acting on the space X, and the goal of a memoryless quantized feedback map is to choose at every step a map $f_1, f_2, \ldots$ or f_N to apply to the system, in order to confine almost every initial point to a subset Y.

Hence if there exists a memoryless quantized feedback, then for almost every initial state there is a sequence of maps driving the state to Y where it stays forever.

The following proposition says that the converse also holds.

Proposition 39 *Let $f_1, f_2, \ldots, f_N$ be measure-preserving maps of a probability space X, and Y a measurable subset of X. Then the following conditions are equivalent:*

(a) there exists a memoryless quantized feedback map contracting almost all X to Y;
(b) for almost every point x, there is a sequence $f_{i_1}, f_{i_2}, f_{i_3}, \ldots$ such that the sequence of points $f_{i_1}(x), f_{i_2}f_{i_1}(x), f_{i_3}f_{i_2}f_{i_1}(x), \ldots$ eventually reaches Y and stays in Y forever;
(c) there is a measurable set $Z \subseteq Y$ of positive measure with the following properties:
- *$\bigcup_{i=1}^{N} f_i^{-1}(Z) \supseteq Z$ up to a set of measure zero;*
- *Every subset of positive measure that is invariant under f_1, f_2, ... and f_N has an intersection of positive measure with Z.*

A subset is said to be *invariant* under a measure-preserving map if almost all points of Z are sent into Z.

Proof.

$(a) \Rightarrow (b)$ Obvious.

$(b) \Rightarrow (c)$ Let us take Z to be the set of points of Y that are and will remain in Y for a suitable sequence of maps chosen in $\{f_1, \ldots, f_N\}$. If $Y_0 = Y$ and $Y_{j+1} = Y_j \cap \bigcup_i f_i^{-1}(Y_j)$, then Y_j is the set of points of Y such that stay in Y during j consecutive steps, for a suitable sequence of maps.

We prove that $Z = \bigcap_{j\in\mathbb{N}} Y_j$. Indeed if, for every j, a point stays in Y during j steps for the sequence of maps $w_j \in \{1, \ldots, N\}^j$, then from compactness of $\{1, \ldots, N\}^{\mathbb{N}}$ we can extract an infinite sequence $w \in \{1, \ldots, N\}^{\mathbb{N}}$ that has arbitrarily long prefixes of the form w_j, and this sequence is a sequence of maps that makes the point remain in Y forever.

Thus Z is measurable, and the first condition of (c) is satisfied.

Now consider $Z_0 = Z$ and $Z_{j+1} = Z_j \cup \bigcup_i f_i^{-1}(Z_j)$. As it is possible to confine almost every point to Y, then $X = \bigcup_{j\in\mathbb{N}} Z_j$, up to a set of measure zero. Then Z has a positive measure, because the countable union of sets of measure zero is also a set of measure zero. Moreover, the second condition of (c) is satisfied, because a subset invariant under all N maps and disjoint from Z must be included in $X \setminus \bigcup_{j\in\mathbb{N}} Z_j$.

$(c) \Rightarrow (a)$ Given such a Z, we construct a partition of X into N subsets $X_1, X_2, \ldots, X_N$.

First label with a '1' all points x of Z such that $f_1(x) \in Z$. Then label with a '2' all points of Y that have no label and whose image by f_2 is in Z. Do the same for all N functions. From the first condition, almost every point of Z gets a label.

Then go successively through the following steps

- among the yet unlabelled points of X, label with a '1' those that will reach a labelled point via one application of f_1;
- among the yet unlabelled points of X, label with a '2' those that will reach a labelled point via one application of f_2;
- ...
- among the yet unlabelled points of X, label with an 'N' those that will reach a labelled point via one application of f_N;
- among the unlabelled points of X, label with a '1' those that will reach a labelled point via one application of f_1;
- ...

and so on, cycling through $1, 2, \ldots, N$. The set of points that never get a label through this process is invariant under every map $f_1, f_2, \ldots, f_N$ and is disjoint from Z. From the second condition of (c), this set must have measure zero. Thus almost every point gets a label.

Now call X_1 the set of points labelled by '1', X_2 the set of points labelled by '2', etc. By construction, the feedback map 'If the state is in X_i then apply f_i' drives almost every point to $Z \subseteq Y$, where it stays for ever.

Now we refine the partition $\{X_1, \ldots, X_N\}$ with the partition $\{Z, X \setminus Z\}$, i.e., we consider the partition $\{X_i \cap Z, X_i \setminus Z | i = 1, 2, \ldots, N\}$. We finally have $2N$ quantization subsets. The feedback map becomes 'If the current state is in $X_i \cap Z$ or in $X_i \setminus Z$ then apply f_i', and once we reach $X_i \cap Z$, for some i, we know that we are in Z and will not leave it anymore (with probability 1). □

Note that measure-preserving property of maps is not essential in the proof; we could require only that the maps are measurable and that the inverse image of set of measure zero has measure zero.

Roughly speaking, this proposition tells that feedback is equivalent to memoryless quantized feedback. A system can be controlled in feedback if and only if it can be controlled by a memoryless quantized feedback. However it says

nothing on the performance that can be obtained with memoryless quantized feedback.

6.3.2 An information-theoretic bound

The following inequality can be obtained easily.

Proposition 40 *Suppose that with a certain quantized feedback map, almost all the space X is eventually concentrated to a set of measure $1/C$. Then*

$$\log_2 C \leq H(X_{i_0}, X_{i_1}, \ldots, X_{i_{t-1}}), \tag{6.1}$$

where t is the first entry time, $X_1, \ldots, X_N$ are the quantization subsets and $H(X_{i_0}, X_{i_1}, \ldots, X_{i_{t-1}})$ is the Shannon entropy in base 2 of the set of sequences of quantization subsets that can be observed.

See Section 2.2 for the definition of entropy.

Proof. Let Y be the target set of measure $1/C$. Call $X_{i_0 i_1 \ldots i_{t-1}} \subseteq X$ the set of initial conditions that generate the sequence of quantization subsets $X_{i_0}, X_{i_2}, \ldots, X_{i_{t-1}}$ when the feedback map is applied. The set of all $X_{i_0 i_1 \ldots i_{t-1}}$ forms a finite or countable partition of X, and we can write

$$H(X_{i_0}, X_{i_1}, \ldots, X_{i_t}) = -\Sigma \mu(X_{i_0 i_1 \ldots i_t}) \log_2 \mu(X_{i_0 i_1 \ldots i_t}).$$

Recall that μ denotes the probability measure on space X.

Now consider the set $f_{i_0}^{-1} f_{i_1}^{-1} \ldots f_{i_{t-1}}^{-1}(Y)$, where f_i is the map applied to quantization subset X_i. From the measure-preserving property of f_i, the measure of this set is $1/C$. But $X_{i_0 i_1 \ldots i_{t-1}} \subseteq f_{i_0}^{-1} f_{i_1}^{-1} \ldots f_{i_{t-1}}^{-1}(Y)$ (up to a set of measure zero); this follows from the fact that all points of $X_{i_0 i_1 \ldots i_{t-1}}$ go to Y in time t. Thus $\mu(X_{i_0 i_1 \ldots i_{t-1}}) \leq 1/C$. Thus the function $-\log_2 \mu(X_{i_0 i_1 \ldots i_{t-1}})$ is at least $\log_2 C$ and its average $H(X_{i_0}, X_{i_1}, \ldots, X_{i_{t-1}})$ is at least $\log_2 C$. □

Note that the probability measure of the state after time t is no longer μ in general, but is concentrated on the target subset.

The first term of Inequality (6.1) can be seen as the amount of information that we have gained on the present state of the system. The second term is the average information collected on the initial state of the system by observing all the quantization subsets during the control.

In other words, after the process of control we know more about the initial state than about the current state. Informally, we can say that it is easier to observe a system than to control it.

A more concrete bound is the following.

Corollary 5 *With the hypotheses of Proposition 40,*

$$\log C \leq T \log N, \tag{6.2}$$

where C is the contraction ratio, T the average first entry time and N is the number of quantization subsets.

Proof. This follows from Proposition 40, Shannon's noiseless coding theorem (see Section 2.5) and the fact that the set $\{(X_{i_0}, X_{i_1}, \ldots, X_{i_{t-1}})\}$ of observed sequences of quantization subsets before the control objective is known to be completed is a prefix language. □

Proposition 40 and Corollary 5, proved in the general case of measure-preserving maps, thus apply to the case of the map $f(x,u) = ax + u \mod 2$ (where $|a| > 1$ is an integer) and fixed finite set of values $u_1, \ldots, u_N$. These facts however were already proved by Fagnani and Zampieri [FZ04b], under the form of equivalent bounds holding for the case of the map $f(x,u) = ax + u$ with arbitrary real slope a.

6.3.3 Maxwell's demon

The *thermodynamic entropy* of a physical system is identified, according to Boltzmann's formula, to the quantity $k_B \ln \Omega$, where k_B is Boltzmann's constant and Ω is the number of microscopic configurations of the physical system compatible with the macroscopic observations. The second law of thermodynamics states the the entropy of an isolated physical system cannot decrease.

Maxwell's demon is a small but clever being that stands near a physical system; he performs measurements on the system and, according to the result of these measurements, modifies the physical system without any expense of energy. By its measurements, the demon is able to reduce the thermodynamic entropy of the system, in apparent contradiction with the second law of thermodynamics. Bennett proposed a solution to this paradox: at every step the demon, who has a limited memory of the past, erases the result of the preceding measurement, thus generating entropy. Indeed *Landauer's Principle* states that erasing a bit of information is an irreversible process and must therefore increase the thermodynamic entropy of the universe by at least $k_B \ln 2$ (where k_B is Boltzmann's constant); see [Ben82] for a detailed description. See also Bricmont [Bri95] for an account of the second law of thermodynamics and Maxwell's demon with a historic perspective.

The setting presented in this section can be viewed as a formalization of Maxwell's demon paradox in control theory. The demon is the feedback map and the physical system is the system of space X.

If the physical systems manipulated by the demon are at equilibrium, this means that all statistical quantities of the system are preserved in time; this also

means that the probability distribution on the state of the system is invariant. This corresponds to the fact that $f(.,u)$ preserves the measure.

If the feedback map tends to concentrate the state to a subset of measure $1/C$, then the thermodynamic entropy of the system is decreased by (at least) $k_B \ln 2 \log_2 C$. Now recall the bound given by Corollary 5:

$$\log_2 C \leq H(X_{i_0}, X_{i_1}, \ldots, X_{i_{t-1}}).$$

If we want to encode the result of all measurements performed on the system, then the average length of the transcription of these measurements will be greater than $H(X_{i_0}, X_{i_1} \ldots X_{i_{t-1}})$, in virtue of Shannon's noiseless coding theorem (see Section 2.5). So the erasure of information gained during the process of control (up to a factor of $k_B \ln 2$) must increase the thermodynamic entropy of the environment by at least $k_B \ln 2 H(X_{i_0}, X_{i_1}, \ldots, X_{i_{t-1}})$. We conclude that the total thermodynamic entropy of the universe (system and environment) has not decreased. As such, Corollary 5 can be viewed as an application of the second law of thermodynamics in control theory.

6.4 An optimal scalar quantized feedback map

We now show how to construct a memoryless quantized feedback map for scalar linear systems with integral slope that comes close to the bound given by Corollary 5. For the sake of simplicity, we first work on the interval $[0,1]$, then a change of variable will bring us back to $[-1,1]$.

Let us consider the map $f : [0,1] \times \mathbb{R} \to [0,1] : (x,u) \mapsto ax + u \mod 1$. The idea is the following. Suppose, with $a = 10$, that we want to drive the points of $[0,1]$ to $[0, 10^{-6}]$. This means that we need to force the first six decimal digits to be zeros. First remark that the system $x \mapsto 10x \mod 1$ (with $u = 0$) has the effect to shift the decimal expansion to the left, and suppress the first digit. Switching off six digits can be done in six steps if at every step we measure the seventh digit of the state and we choose the input u in order to cancel it. If $x_0 = 0.12345678987654321\ldots$ (in decimal), then the system goes through the states $x_1 = 0.2345608987654321\ldots$, $x_2 = 0.345600987654321\ldots$, $x_3 = 0.45600087654321\ldots$, $x_4 = 0.5600007654321\ldots$, $x_5 = 0.600000654321\ldots$, $x_6 = 0.00000054321\ldots$, $x_7 = 0.0000004321\ldots$, etc.

If $a < 0$, then we write x in base a again: $x = b_1 a^{-1} + b_2 a^{-2} + b_3 a^{-3} + \cdots \mod 1$, where $b_i = 0, 1, \ldots$ or $|a| - 1$. For instance, in base -10, the numbers of the form $0, b_1 b_2 b_3 \ldots$ range from $0.909090\ldots = -9(10^{-1} + 10^{-3} + 10^{-5} + \cdots) = -10/11$ to $0.09090\ldots = 9(10^{-2} + 10^{-4} + 10^{-6} + \cdots) = 1/11$. Taken modulo 1, this covers the whole interval $[0,1]$.

Proposition 41 *If $|a| \geq 2$ is an integer, then for every integer n there is a feedback map for $f : (x,u) \mapsto ax + u \mod 1$ contracting $[0,1]$ to an interval of length $|a|^{-n}$ in time n with $|a|$ quantization subsets.*

Proof. Let us represent $x \in [0,1]$ in the form $x = \Sigma_{i=1,2,3,\ldots} b_i a^{-i} \mod 1$, where $b_i \in \{0, 1, \ldots, |a| - 1\}$. If a is positive then this is the usual representation in base a. In any case, this representation is unique except for a subset of measure zero.

Now call $X_{i,b}$ the set of reals of $[0,1]$ whose ith digit in expansion is b. We use the sets $X_{n+1,0}, \ldots, X_{n+1,|a|-1}$ as quantization subsets, and it is clear that the feedback map 'Apply $x \mapsto ax - ba^{-n}$ if $x \in X_{n+1,b}$' will have for effect to switch progressively all first n digits to 0, as in the example above.

If a is positive, then after n steps we are and we stay in the set $[0, a^{-n}]$.

If $a < 0$, then after n steps we stay in the set $X_{1,0} \cap \cdots \cap X_{n,0} = [-|a|^{-n} \frac{1-|a|^{-1}}{1-|a|^{-2}}, |a|^{-n} \frac{|a|-1}{|a|^2-1}[$ if n is even and in the set $X_{1,0} \cap \cdots \cap X_{n,0} = [-|a|^{-n} \frac{|a|-1}{|a|^2-1}, |a|^{-n} \frac{1-|a|^{-1}}{1-|a|^{-2}}[$ if n is odd. In any case, this is an interval of length a^{-n}. Note that one endpoint is negative. The interval taken modulo 1 is actually composed of two intervals in $[0,1]$. □

On Figure 6.1, the feedback map for $a = 2$ and $n = 3$ is graphically represented.

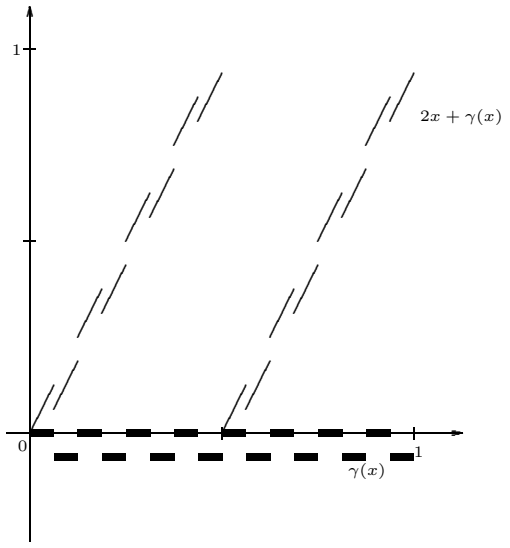

Fig. 6.1. An optimal feedback for the map $x \mapsto 2x+u \mod 1$ on the interval $[0,1]$. In thick line is the feedback map $x \mapsto u = \gamma(x)$, which takes two values: 0 and $-1/8$. In thin line is the closed-loop map $x \mapsto 2x+\gamma(x) \mod 1$. There are $N = 2$ quantization subsets, the contraction ratio is $C = 8$, the time needed to drive every point of $[0,1]$ to the interval $[0, 1/8]$ is $T = 3$.

If at every step we change k digits to 0 in the expansion of the state x (instead of just one), then the time needed is divided by k.

For instance, if $a = 10$, $k = 2$ and we want to switch off the first six digits of the initial state, say, $x_0 = 0.12345678987654321\ldots$, then at every step we measure the fourth and seventh digits and choose the right input. The successive states of the system are $x_1 = 0.2305608987654321\ldots$, $x_2 = 0.300600987654321\ldots$, $x_3 = 0.00000087654321\ldots$, $x_4 = 0.0000007654321\ldots$, $x_5 = 0.000000654321\ldots$, etc.

We thus extend the proposition in the following way.

Proposition 42 *If $|a| \geq 2$ is an integer, then for every integer n and every divisor k of n there is a feedback map for f contracting $[0,1]$ to an interval of size $|a|^{-n}$ in time n/k with $|a|^k$ quantization subsets.*

Proof. We extend the argument used in the preceding proof. The state $x \in [0,1]$ is expressed in the form $x = \Sigma_{i=1,2,3,\ldots} b_i a^{-i} \mod 1$, where $b_i \in \{0, 1, \ldots, |a| - 1\}$.

We measure the digits of position $n/k+1$, $2n/k+1$, $\ldots$, $n+1$s in this expansion and we apply the feedback map $x \mapsto ax - b_{n/k+1}a^{-n/k} - b_{2n/k+1}a^{-2n/k} - \cdots - b_{n+1}a^{-n} \mod 1$.

This strategy reduces $[0,1]$ to an interval of length $|a|^{-n}$ in n/k steps by partitioning the interval into $|a|^k$ subsets. □

Note that it is possible to contract the interval to any subinterval of length $|a|^{-n}$ instead. Indeed, suppose that the feedback map '$x \mapsto ax + u_i$ if $x \in X_i$' reduces $[0,1]$ to $[0,\epsilon]$, and consider the change of variables $y = x + r \mod 1$, $v = u - ar + r$ and $Y_i = X_i + r \mod 1$. Then the feedback map '$y \mapsto ay + v_i$ if $y \in Y_i$' reduces $[0,1]$ to $[r, r+\epsilon]$.

Moreover, by the change of variables $y = 2x - 1$, $v = 2u - 1 + a$ and $Y_i = 2X_i - 1$, we transport the conclusions to the system $f(x,u) = ax + u \mod 2$ on $[-1,1]$.

Proposition 43 *Consider the system $f(x,u) = ax + u \mod 2$ on $[-1,1]$, with $|a| > 1$ an integer. For any $C \geq 1$ and $N \geq |a|$, there is a feedback map taking a constant time $T = \lceil \frac{\log_{|a|} C}{\lfloor \log_{|a|} N \rfloor} \rceil \leq \min\{\frac{\log C}{\log N - \log |a|} + 1, 2\frac{\log C}{\log N} + 1\}$ that reduces the interval $[-1,1]$ into $[-1/C, 1/C]$ with at most N quantization subsets.*

Proof. Take $k = \lfloor \log_{|a|} N \rfloor$ and $n = k\lceil \frac{\log_{|a|} C}{k} \rceil$ in Proposition 42 and then apply appropriate change of variables. The bound $\lceil \frac{\log_{|a|} C}{\lfloor \log_{|a|} N \rfloor} \rceil \leq 2\frac{\log C}{\log N} + 1$ comes from the fact that $\frac{1}{2} \log_{|a|} N \leq \lfloor \log_{|a|} N \rfloor$. □

Note that, provided that we specify how to handle states that have two expansions in base a, this feedback map works for *all* points of the interval $[-1, 1]$, and not only for *almost all* of them.

This family of feedback maps satisfies the bound $\log C \leq T \log N$ given by Corollary 5 with approximate equality, if N, C, T are not too small. Hence this family is optimal, in the sense that if two parameters among T, N, C are fixed, then the third cannot be much improved.

According to Proposition 43 for every integers n and k, where k divides n, we have a feedback map with $T = n/k, N = |a|^k, C = |a|^n$. This is a family with two parameters. If we add a constraint on k and n, then we have a family of one parameter.

In particular if we take $n = k|a|^k$ then we obtain a family of feedback maps for which $T = N = |a|^k = n/k$ and $C = |a|^n = N^N = T^T$. For this family some of the main bounds linking N, C and T proved in [FZ05] and listed below do not hold, as easily checked.

- *For any $r > 0$, there exists $s > 0$ such that $N \leq r \log_2 C$ implies $\lceil T \rceil \geq s \log_2 C$;*
- *For any $r > 0$, there exists $s > 0$ such that $\lceil T \rceil \leq r \log_2 C$ implies $N \geq s \log_2 C$;*
- *There exist $K_1 > 0, \beta_1 > 0, C_1 > 1$ such that $C \geq C_1$ and $N \leq \beta_1 \log_2 C$ imply $T \geq K_1 N C^{1/N}$;*
- *There exist $K_2 > 0, \beta_2 > 0, C_2 > 1$ such that $C \geq C_2$ and $\lceil T \rceil \leq \beta_2 \log_2 C$ imply $N \geq K_1 \lceil T \rceil C^{1/\lceil T \rceil}$.*

Let us remark that this 'counter-example' does not formally contradict the results of Fagnani and Zampieri, according to which N counts the number of intervals on which the feedback map is constant. This essentially amounts to imposing that quantization subsets are intervals, whereas we only require the quantization subsets to be measurable.

Recall also that between the number N of subsets in the case of the map $x \mapsto ax + u \mod 2$ and the map $x \mapsto ax + u$, there might be a factor at most $|a| + 1$, but this does not affect the conclusions. On Figure 6.2, an example of optimal feedback map for for the map $x \mapsto 2x + u$ on the interval $[0, 1]$, with $N = 4$ quantization subsets; compare with Figure 6.1, where for the map $x \mapsto 2x + u \mod 1$, there are only two quantization subsets.

This shows that imposing a topological condition on the quantization subsets, like being an interval, is actually a restriction. This constraint is however not a communication constraint ('How many bits can I transmit?') but rather a measurement constraint ('What kind of information can I acquire on the state of the system?').

We therefore think that if the problem of quantized feedback is motivated by a bounded rate of transmission between the output and the input of the system (because, for instance, the information needed to control the system

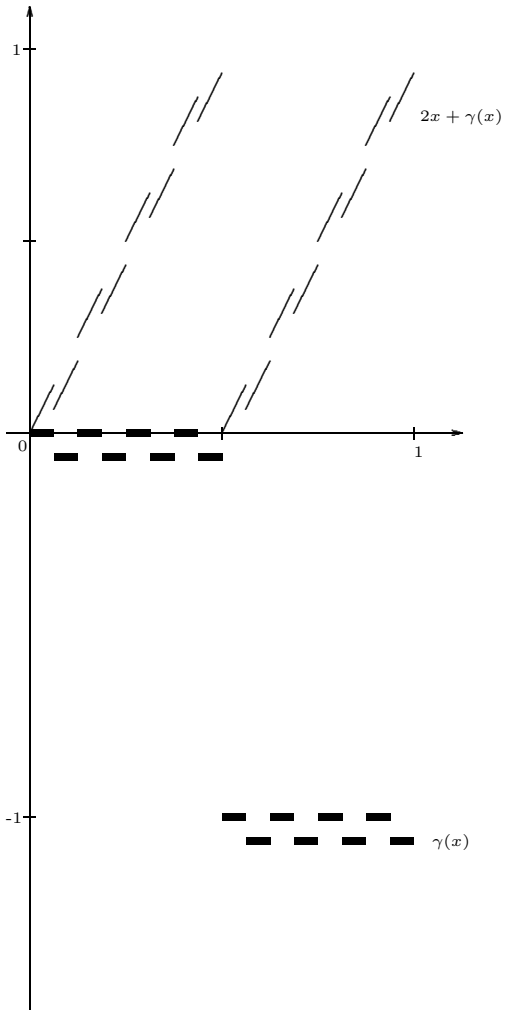

Fig. 6.2. An optimal feedback for the map $x \mapsto 2x + u$ on the interval $[0, 1]$. In thick line is the feedback map $x \mapsto u = \gamma(x)$, which takes four values: 0, $-1/8$, -1 and $-9/8$. In thin line is the closed-loop map $x \mapsto 2x + \gamma(x)$, which is the same as in Figure 6.1. There are $N = 4$ quantization subsets, the contraction ratio is $C = 8$, the time needed to reach the interval $[0, 1/8]$ is $T = 3$.

must be sent via the internet), then we can drop the constraint of intervals without harm.

But if the quantized feedback theory is used to model a situation where the measurement instruments have finite precision, thus quantizing the state space into finitely many subsets, then it is very plausible that the quantization subsets are connected (i.e., are intervals, in the scalar case).

6.5 Conclusions

We studied the framework of memoryless quantized feedback for scalar linear systems proposed by Fagnani and Zampieri, and exhibited a new family of optimal feedback maps, provided that we drop the hypothesis that the quantization subsets are intervals. Our solution works only for integral slopes. Extension to nonintegral slopes and higher dimensions is to be studied.

Future work could also be devoted to developing the general formalism sketched in Section 6.3.

7

Conclusions

I started this research with a feeling and an ambition. I had the feeling that computation, control, dynamical systems and information are all intermingled concepts, the interrelations of which are not fully understood, and my ambition was to think hard on this and build, at least for myself, a coherent vision incorporating all these concepts in their most general meaning.

Even such a vast programme has to start somewhere. A paper of Blondel, Cassaigne and Nichitiu [BCN02] exhibited a Turing machine with no periodic configuration, refuting a conjecture by Kůrka [Kůr97b]. At the same time, an article by Bruno Durand [Dur97] proved that, although not all tile sets that can tile the plane can do so periodically (as known since Berger [Ber66]), it can always do so in an almost periodic way. I therefore attached myself to parallel these two situations, and it seemed to me that several results had some common points. For instance, not only Berger and Hooper theorems both express the undecidability of checking the existence of a configuration that avoids some pattern (respectively, a valid tiling or an immortal configuration of a Turing machine), but their proofs bear some strong resemblance. I pushed this similarity as far as I could, translating some theorems known about tilings to Turing machines and trying to give the most general framework to express the results: dynamical systems, as sets on which acts a monoid. I sowed the seeds of this research in my master thesis and made it grow in my Ph.D. thesis, giving birth to Chapter 3. In particular, I came to prove some properties to be undecidable for classes of dynamical systems, including Turing machines and tilings.

Following this result of undecidability, I shifted my interest towards computational universality of a single dynamical system, rather than considering a whole class of systems. Universality of cellular automata, e.g., automaton 110 or the Game of Life, relies on checking whether a ultimately periodic configuration is in the orbit of another ultimately periodic configuration, while for analog systems, e.g., piecewise polynomial maps, rational points are observed.

The choice of such a particular set of states to work with appeared to me as both physically unrealizable and mathematically arbitrary. I therefore worked out a new definition, which I improved several times to eventually reach the definition of Chapter 4, which can be applied to any symbolic discrete-time system, such as cellular automata or Turing machines. Unfortunately I could not reach a fully satisfactory definition for analog systems. Exploring the properties of this new definition, looking for examples and necessary conditions was the occasion to collaborate fruitfully with Prof. Petr Kůrka.

During my master thesis, my advisor Prof. Vincent Blondel attracted my attention on a paper by Egerstedt and Brockett [EB03]. It proposed an appealing example, how to describe a path in a town in the most concise way, and showed how incorporating some feedback allowed shorter descriptions. The solution proposed by Egerstedt and Brockett was however rather complex, and I wondered how would behave a simpler formalism. I started this research during my master thesis and pursued it in my Ph.D. thesis. The conclusion is that in this simple framework, feedback is not much shorter than open-loop. The final results of this research are exposed in Chapter 5.

Meanwhile, my advisor, Vincent Blondel again, made me aware of the work of Fagnani and Zampieri [FZ01], who proposed a simply-formulated question, yet with highly nontrivial developments. How to control a scalar linear system with a limited number of possible inputs? This raised the question of how much information is needed to control an input-output system, a question also examined by other authors. Dropping a hypothesis made by Fagnani and Zampieri, I derived a solution of unexpected simplicity for integer slopes, which achieves an information-theoretic bound already derived in some form by Fagnani and Zampieri [FZ04b]. I also attached myself to investigate a general formalism of control with a quantized input, inspired by classical ergodic theory and considerations on Maxwell's demon dating back to the 1980's [Ben82]. This general theory is only sketched in Chapter 6.

Many more chapters could be written on the interplay of dynamics and computation. Let us see some possible ways for further research.

For instance, we can consider a Turing machine (say, without blank symbol) as an input-output system, whose state is the content of the tape, that is controlled by an input-output automaton, that can at every step measure one symbol on the tape and apply one of these maps: shift to the left, shift to the right, switch the currently read symbol. That point of view may be seen as an extension of Chapter 4, where the automaton is not only an observer but a controller. If we endow the symbols of the tape with independent and equal probabilities (this is the so-called Bernoulli distribution), then we can check that all maps that can be applied by the controller automaton preserve the measure. Hence we are almost in the framework of Chapter 6; the only difference is that the controller has a finite memory instead of having no memory at all.

Hence we could apply ideas and methods of both Chapters 4 and 6. In particular, analyzing the flow of information in a universal Turing machine may be related to the possibility of reversible computation (because reversibility is intuitively the fact that no information is destroyed), and to the so-called universal measure on integers, which gives the probability for a number to be the output of a universal Turing machine with random input data.

If we believe that the (classical) physical world is a dynamical system on a real manifold rather than on a Cantor set—but not everyone thinks so, see for instance Wheeler's 'it for bit' [Zur89], Toffoli [Zur89] or Wolfram [Wol02]—then we really should find a satisfactory formalism adapted to analog systems, that would extend Chapter 4. The question of which systems of classical physics are capable of computation, if any, could be asked in this framework. One might feel that the gravitational N-body system, for instance, is not universal. A further step would be to consider quantum systems.

One could also dream of formulating and solving in a unified context including dynamical systems and input-output dynamical systems. Universal coalgebra [Rut00], theory advocated as a unifying framework for all formalisms of dynamical systems, could possibly help for this goal. A coalgebra is a map $X \to F(X)$, where $F(X)$ is a set built from the state space X. For discrete-time dynamical systems, $F(X) = X$, while for input-output dynamical systems, $F(X) = (X \times Y)^U$, where U is the input set and Y the output set. Thus a coalgebra is a very general dynamical system.

While the time has come to end this thesis, I remain with the feeling that computation, control, dynamical systems and information are all intricately intermingled concepts, in a way far from being fully understood, and my ambition for future research is to shed some light in the dark.

References

[AB01] E. Asarin and A. Bouajjani. Perturbed turing machines and hybrid systems. In *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science (LICS-01)*, pages 269–278, Los Alamitos, CA, June 16–19 2001. IEEE Computer Society.

[AB02] R. Albert and A. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.

[Abr63] N. Abramson. *Information theory and coding*. McGraw-Hill, New-York, 1963.

[AMP95] E. Asarin, O. Maler, and A. Pnueli. Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138(1):35–65, 1995.

[BBC+92] J. Banks, J. Brooks, G. Cairns, G. Davis, and P. Stacey. On Devaney's definition of chaos. *American Mathematics Monthly*, 99:332–334, 1992.

[BC96] O. Bournez and M. Cosnard. On the computational power of dynamical systems and hybrid systems. *Theoretical Computer Science*, 168:417–459, 1996.

[BCG85] E. R. Berlekamp, J. H. Conway, and R. K. Guy. *Winning ways for your mathematical plays. Volume 2: Games in particular*. Academic Press, New York-San Francisco-London-San Diego, 1985.

[BCN02] V. D. Blondel, J. Cassaigne, and C. M. Nichitiu. On the presence of periodic configurations in turing machines and in counter machines. *Theoretical Computer Science*, 289(1):573–590, 2002.

[Ben82] Ch. H. Bennett. The thermodynamics of computation–a review. *International Journal of Theoretical Physics*, 21(12):905–940, 1982.

[Ber66] R. Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 66:1–72, 1966.

[BHSF02] A. Ben-Hur, H. T. Siegelmann, and S. Fishman. A theory of complexity for continuous time systems. *Journal of Complexity*, 18(1):51–86, 2002.

[Biś04] A. Biś. Entropies of a semigroup of maps. *Discrete and Continuous Dynamical Systems*, 11(2–3):639–648, 2004.

[BL00] R. Brockett and D. Liberzon. Quantized feedback stabilization of linear systems. *IEEE Transactions on Automatic Control*, AC-45:1279–1289, 2000.

[Bri95] J. Bricmont. Science of chaos or chaos in science? *Physicalia Magazine*, 17(3–4):159–208, 1995.

[Bro76] J. Brown. *Ergodic Theory and Topological Dynamics*. Academic Press, New York, 1976.

[Cas01] J. Cassaigne. Recurrence in infinite words (extended abstract). In *STACS: Annual Symposium on Theoretical Aspects of Computer Science*, 2001.

[CD00] J. Cervelle and B. Durand. Tilings: Recursivity and regularity. In *STACS: Annual Symposium on Theoretical Aspects of Computer Science*, 2000.

[Con72] J. H. Conway. Unpredictable iterations. In *Proceedings of the Number Theory Conference (Univ. Colorado, Boulder, Colo., 1972)*, pages 49–52, Boulder, Colo., 1972. Univ. Colorado.

[Cut80] N. J. Cutland. *Computability: An introduction to recursive function theory*. Cambridge University Press, Cambridge, 1980.

[CY89] J. P. Crutchfield and K. Young. Computation at the onset of chaos. In W. Zurek, editor, *Complexity, Entropy and the Physics of Information*, pages 223–269. Addison-Wesley, 1989.

[ČHY90] K. Čulik, L. P. Hurd, and S. Yu. Computation theoretic aspects of cellular automata. *Physica D*, 45:357–378, 1990.

[ČI96] K. Čulik II. An aperiodic set of 13 Wang tiles. *Discrete Math.*, 160(1-3):245–251, 1996.

[ČIK97] K. Čulik II and J. Kari. On aperiodic sets of wang tiles. In Chr. Freksa, M. Jantzen, and R. Valk, editors, *Foundations of Computer Science: Potential - Theory - Cognition*, volume 1337 of *Lecture Notes in Computer Science*, pages 153–162. Springer, 1997.

[Dav56] M. D. Davis. A note on universal Turing machines. In C.E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 167–175. Princeton University Press, 1956.

[DB04a] J.-Ch. Delvenne and V. D. Blondel. Complexity of control on finite automata. In B. De Moor, B. Mortmans, J. Willems, P. Van Dooren, and V. D. Blondel, editors, *Proceedings of the Symposium on Mathematical Theory of Networks and Systems (MTNS 2004)*, Leuven, 2004.

[DB04b] J.-Ch. Delvenne and V. D. Blondel. Quasiperiodic configurations and undecidable dynamics for tilings, infinite words and Turing machines. *Theoretical Computer Science*, 319:127–143, 2004.

[DB06] J.-Ch. Delvenne and V. D. Blondel. Complexity of control on finite automata. *IEEE Transactions on Automatic Control*, To appear in 2006.

[Del90] D. Delchamps. Stabilizing a linear system with quantized state feedback. *IEEE Transactions on Automatic Control*, AC-35:916–924, 1990.

[Del06] J.-Ch. Delvenne. An optimal quantized feedback strategy for scalar linear systems. *IEEE Transactions on Automatic Control*, To appear in 2006.

[Dev89] R. Devaney. *An Introduction to Chaotic Dynamical Systems*. Addison-Wesley, 1989.

[DK03] M. Dietzfelbinger and M. Kunde. A case against using Stirling's formula (unless you really need it). *Bulletin of the European Association for Computer Science*, 80:153–158, 2003.

[DKB05a] J.-Ch. Delvenne, P. Kůrka, and V. D. Blondel. Computational universality in symbolic dynamical systems. In M. Margenstern, editor, *MCU 2004*, number 3354 in Lecture Notes in Computer Science, pages 104–115. Springer-Verlag, 2005.

[DKB05b] J.-Ch. Delvenne, P. Kůrka, and V. D. Blondel. Computational universality in symbolic dynamical systems. *Fundamenta Informaticae*, Accepted in 2005.

[DR99] B. Durand and Z. Róka. The Game of Life: universality revisited. In M. Delorme and J. Mazoyer, editors, *Cellular Automata: a Parallel Model*, volume 460 of *Mathematics and its Applications*, pages 51–74. Kluwer Academic Publishers, 1999.

[Dur97] B. Durand. Tilings and quasiperiodicity. *Lecture Notes in Computer Science*, 1256:65–75, 1997.

[EB03] M. Egerstedt and R. Brockett. Feedback can reduce the specification complexity of motor programs. *IEEE Transactions on Automatic Control*, 48(2):213–223, 2003.

[EEN00] D. B. Ellis, R. Ellis, and M. Nerurkar. The topological dynamics of semigroup actions. *Transactions of the American Mathematical Society*, 353(4):1279–1320, 2000.

[EM01] N. Elia and S. K. Mitter. Stabilization of linear systems with limited information. *IEEE Transactions on Automatic Control*, AC-46:1384–1400, 2001.

[FT82] E. Fredkin and T. Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3-4):219–253, 1981/82.

[FZ01] F. Fagnani and S. Zampieri. Stability analysis and synthesis for scalar linear systems with a quantized feedback. Technical Report 20, Dipartimento di Matematica, Politecnico di Torino, 2001.

[FZ03] F. Fagnani and S. Zampieri. Stability analysis and synthesis for scalar linear systems with a quantized feedback. *IEEE Transactions on automatic control*, 48(9):1569–1584, 2003.

[FZ04a] F. Fagnani and S. Zampieri. Quantized stabilization of linear systems: complexity versus performance. *IEEE Transactions on automatic control*, 49(9):1534–1548, 2004.

[FZ04b] F. Fagnani and S. Zampieri. Steady state and transient performance in memoryless quantized controllers. Unpublished talk given at the *Symposium on Mathematical Theory of Networks and Systems (MTNS 2004)*, Leuven, 2004.

[FZ05] F. Fagnani and S. Zampieri. A symbolic dynamics approach to performance analysis of quantized feedback systems: the scalar case. *SIAM Journal on Control and Optimization*, 44(3):816–866, 2005.

[Gác97] Peter Gács. Reliable cellular automata with self-organization. In *38th Annual Symposium on Foundations of Computer Science*, pages 90–99, Miami Beach, Florida, 20–22 October 1997. IEEE.

[Gar70] M. Gardner. Mathematical games. the fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223:120–123, Oct 1970.

[Gil87] R. H. Gilman. Classes of cellular automata. *Ergodic Theory & Dynamical Systems*, 7:105–118, 1987.

[HADB05] J. M. Hendrickx, B. D.O. Anderson, J.-Ch. Delvenne, and V. D. Blondel. Directed graphs for the analysis of rigidity and persistence in autonomous agent systems. *International Journal of Robust and Non-Linear Control*, Accepted in 2005.

[Hal50] P. R. Halmos. *Measure Theory*. Van Nostrand, Princeton, New Jersey, 1950.

[HCR+04] Sh. Halevy, J. Chen, Ron Roth, P. Siegel, and J. K. Wolf. Improved bit-stuffing bounds on two-dimensional constraints. *IEEE Transactions on Information Theory*, 50(5):824–838, 2004.

[Hed69] G. A. Hedlund. Endomorphisms and automorphisms of the shift dynamical systems. *Mathematical Systems Theory*, 3(4):320–375, 1969.

[Hem02] A. Hemmerling. Effective metric spaces and representations of the reals. *Theoretical Computer Science*, 284(2):347–372, 2002.

[HJ84] K. Hrbacek and Th. Jech. *Introduction to Set Theory*. Marcel Dekker Inc., New York, 1984.

[HKČ92] L. P. Hurd, J. Kari, and K. Čulik. The topological entropy of cellular automata is uncomputable. *Ergodic Theory and Dynamical Systems*, 12(2):255–265, 1992.

[Hoo66] P. Hooper. The undecidability of the Turing machine immortality problem. *Journal of Symbolic Logic*, 31(2):219–234, June 1966.

[Jen01] O. Jenkinson. Strong cocycle triviality for $\mathbb{Z}^2$ subshifts. *Theoretical Computer Science*, 262(1-2):191–213, 2001.

[JGP99] E. M. Clarke Jr., O. Grumberg, and D. A. Peled. *Model checking*. MIT Press, 1999.

[Kai96] R. Kaivola. *Using automata to characterise fixed point temporal logics*. PhD thesis, University of Edinburgh, 1996.

[Kar94] J. Kari. Rice's theorem for the limit sets of cellular automata. *Theoretical Computer Science*, 127(2):229–254, 1994.

[KCG94] P. Koiran, M. Cosnard, and M. Garzon. Computability with low-dimensional dynamical systems. *Theoretical Computer Science*, 132(1-2):113–128, 1994.

[Kel55] J. L. Kelley. *General Topology*. Van Nostrand, Princeton, 1955. Reprinted 1975 by Springer-Verlag as Graduate Texts in Mathematics, volume 27.

[KM99] P. Koiran and Cr. Moore. Closed-form analytic maps in one and two dimensions can simulate universal Turing machines. *Theoretical Computer Science*, 210(1):217–223, 1999.

[Koi96] P. Koiran. A family of universal recurrent networks. *Theoretical Computer Science*, 168(2):473–480, 1996. Universal machines and computations (Paris, 1995).

[Koi01] P. Koiran. The topological entropy of iterated piecewise affine maps is uncomputable. *Discrete Mathematics & Theoretical Computer Science. DMTCS. An Electronic Journal*, 4(2):351–356 (electronic), 2001.

[Kůr93] P. Kůrka. Simplicity criteria for dynamical systems. *Analysis of Dynamical and Cognitive Systems*, pages 189–225, 1993.

[Kůr97a] P. Kůrka. Languages, equicontinuity and attractors in cellular automata. *Ergodic Theory & Dynamical Systems*, 17:417–433, 1997.

[Kůr97b] P. Kůrka. On topological dynamics of Turing machines. *Theoretical Computer Science*, 174(1-2):203–216, 1997.

[Kůr99] P. Kůrka. Zero-dimensional dynamical systems, formal languages, and universality. *Theory of Computing Systems*, 32(4):423–433, 1999.

[Kůr04] P. Kůrka. *Topological and Symbolic Dynamics*, volume 11 of *Cours Spécialisés*. Société Mathématique de France, 2004.

[Lan90] C. G. Langton. Computation at the edge of chaos. *Physica D*, 42:12–37, 1990.

[Lot97] M. Lothaire, editor. *Combinatorics on Words*. Cambridge University Press, second edition, 1997.

[MH36] M. Morse and G. A. Hedlund. Symbolic dynamics. *American Journal of Mathematics*, 3:286–303, 1936.

[MHC94] M. Mitchell, P. T. Hraber, and J. P. Crutchfield. Dynamic computation, and the "edge of chaos": A re-examination. In G. Cowan, D. Pines, and D. Melzner, editors, *Complexity: Metaphors, Models, and Reality*, Santa Fe Institute Proceedings, Volume 19, pages 497–513. Addison-Wesley, 1994. Santa Fe Institute Working Paper 93-06-040.

[Min61] M. L. Minsky. Recursive unsolvability of Post's problem of "tag" and other topics in theory of Turing machines. *Annals of Mathematics (2)*, 74:437–455, 1961.

[MO98] W. Maass and P. Orponen. On the effect of analog noise in discrete-time analog computations. *Neural Computation*, 10(5):1071–1095, 1998.

[Moo56] E. F. Moore. Gedanken-experiments on sequential machines. In C.E. Shannon and J. McCarthy, editors, *Automata Studies*. Princeton University Press, 1956.

[Moo90] Cr. Moore. Unpredictability and undecidability in dynamical systems. *Physical Review Letters*, 64(20):2354–2357, 1990.

[Moo91] Cr. Moore. Generalized shifts: Unpredictability and undecidability in dynamical systems. *Nonlinearity*, 4:199–230, 1991.

[Moo98a] Cr. Moore. Dynamical recognizers: real-time language recognition by analog computers. *Theoretical Computer Science*, 201:99–136, 1998.

[Moo98b] Cr. Moore. Finite-dimensional analog computers: Flows, maps, and recurrent neural networks. In C.S. Calude, J. Casti, and M. J. Dinneen, editors, *Unconventional Models of Computation*. Springer-Verlag, 1998.

[NB99] H. Nagashima and Y. Baba. *Introduction to Chaos: Physics and Mathematics of Chaotic Phenomena*. Institute of Physics Publishing, 1999.

[NEMM04] G. N. Nair, R. J. Evans, I. M. Y. Mareels, and W. Moran. Topological feedback entropy and nonlinear stabilization. *IEEE Transactions on Automatic Control (special issue on Networked Control Systems)*, 49(9):1585–1597, 2004.

[Orp97] P. Orponen. A survey of continuous-time computation theory. In D.-Z. Du and K.-I Ko, editors, *Advances in Algorithms, Languages, and Complexity*, pages 209–224. Kluwer Academic Publishers, 1997.

[PP04] D. Perrin and J.-E. Pin. *Infinite Words (Automata, Semigroups, Logic and Games)*, volume 141 of *Pure and Applied Mathematics*. Elsevier, 2004.

[Ric53] H. Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*, 83, 1953.

[Rob71] R. M. Robinson. Undecidability and non periodicity of tilings of the plane. *Inventiones Mathematicae*, 12:177–209, 1971.

[Rog87] H. Rogers Jr. *Theory of recursive functions and effective computability*. MIT Press, Cambridge, MA, second edition, 1987.

[Ros83] R. D. Rosenkrantz, editor. *E.T. Jaynes: Papers on Probability, Statistics and Statistical Physics*. D. Reidel Publishing, 1983.

[Rut00] J. J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.

[Sch] Christian Schwarzer. *MatLab Random Boolean Network Toolbox*. Freely available on http://www.teuscher-research.ch/rbntoolbox/. *MatLab* is a product of *The MathWorks Inc.*

[SF98] H. T. Siegelmann and S. Fishman. Analog computation with dynamical systems. *Physica D*, 120:214–235, 1998.

[Sie99] H. T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Progress in Theoretical Computer Science. Springer-Verlag, 1999.

[Sut03] K. Sutner. Cellular automata and intermediate degrees. *Theoretical Computer Science*, 296(2):365–375, 2003.

[TB73] B. A. Trakhtenbrot and Y. M. Barzdin. *Finite Automata, Behavior and Synthesis*. North-Holland, 1973.

[TL04] H. Touchette and S. Lloyd. Information-theoretic approach to the study of control systems. *Physica A*, 331:140–172, 2004.

[TM04] S. Tatikonda and S. K. Mitter. Control under communication constraints. *IEEE Transactions on Automatic Control*, 49(7):1056–1068, 2004.

[Tur36] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2):230–265, 1936.

[Ula52] S. Ulam. Random processes and transformations. In *Proceedings of the International Congress of Mathematics (1950)*, volume 2, pages 264–275, 1952.

[vN66] J. von Neumann. *Theory of self-reproducing automata*. University of Illinois Press, Urbana, 1966.

[Wan61] H. Wang. Proving theorems by pattern recognition-II. *Bell Systems Technical Journal*, 40:1–42, Jan 1961.

[Wei00] K. Weihrauch. *Computable Analysis*. Springer-Verlag, 2000.

[Wol02] S. Wolfram. *A new kind of science*. Wolfram Media, Inc., Champaign, IL, 2002.

[Zur89] W. H. Zurek, editor. *Complexity, Entropy and the Physics of Information. Proceedings of the 1988 Workshop on the Complexity, Entropy and the Physics of Information*. Addison-Wesley, Reading MA, 1989.

[Zus69] K. Zuse, editor. *Rechnender Raum*. Vieweg, Braunschweig, 1969.