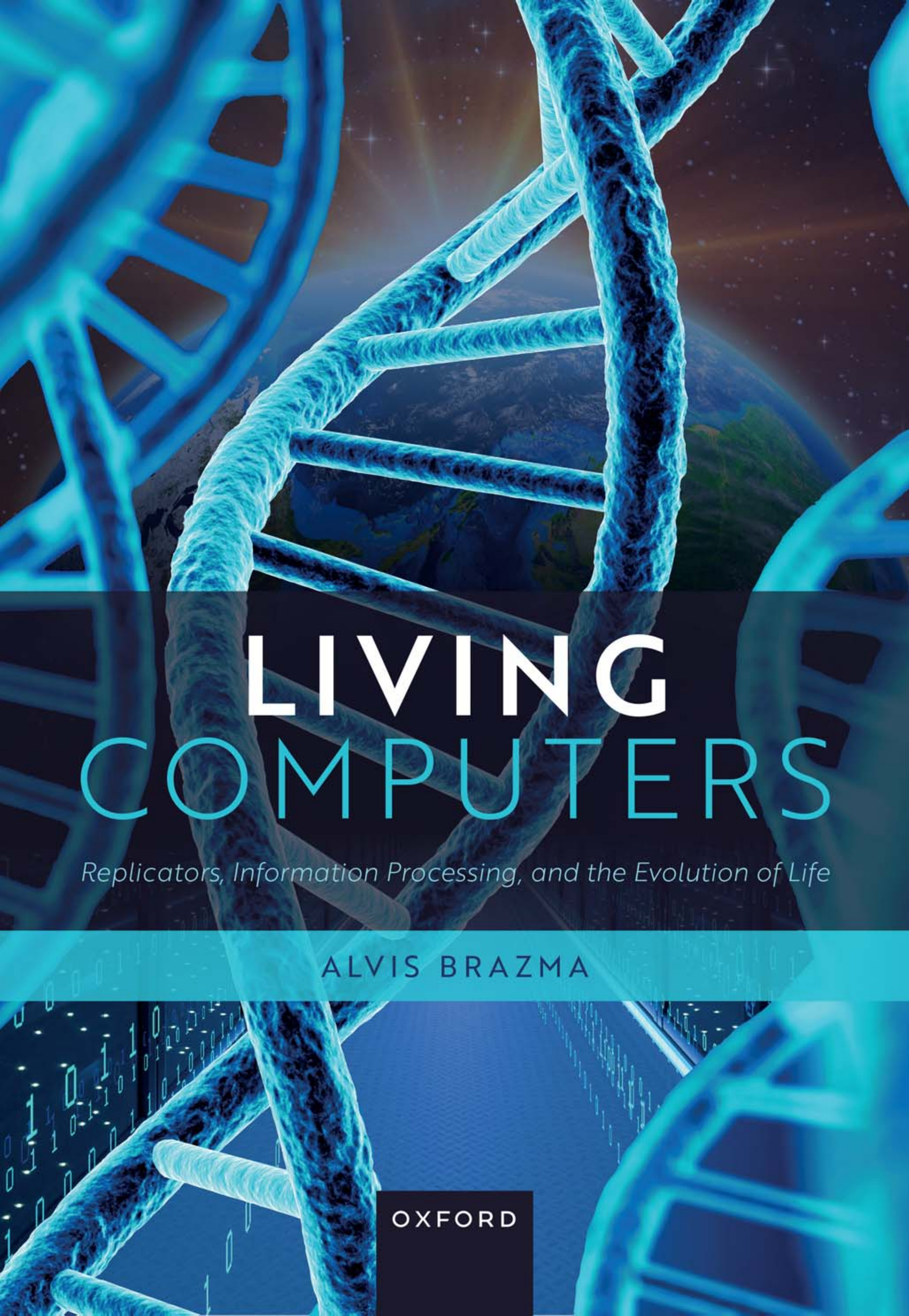
LIVING
COMPUTERS
Replicators, Information Processing, and the Evolution of Life
ALVIS BRAZMA
OXFORD

LIVING COMPUTERS

Living Computers

Replicators, Information Processing, and the Evolution of Life

Alvis Brazma

European Molecular Biology Laboratory
European Bioinformatics Institute
Cambridge, UK

OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

Great Clarendon Street, Oxford, OX2 6DP,
United Kingdom

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide. Oxford is a registered trade mark of
Oxford University Press in the UK and in certain other countries

Published in the United States of America by Oxford University Press
198 Madison Avenue, New York, NY 10016, United States of America

British Library Cataloguing in Publication Data

Data available

Library of Congress Control Number: 2023933858

ISBN 9780192871947

DOI: 10.1093/oso/9780192871947.001.0001

Printed and bound by
CPI Group (UK) Ltd, Croydon, CR0 4YY

For my wife, Diana

Preface

All currently known forms of life rely on three types of molecules: proteins—life's main structural and functional building blocks; DNA—life's information carrier; and RNA—the messenger providing the link between these two. But what was life like when it first emerged on Earth billions of years ago? What will life be like in millions or billions of years if it still exists? Can a living being be made of, say, steel, copper, and silicon? And what is it that distinguishes living systems from those we do not regard as living?

An essential feature of life is its ability to record, communicate, and process information. Every living cell does this. Every animal brain does this. And even more fundamentally, evolution processes the information recorded in the entire collection of DNA on Earth using the algorithm of Darwinian selection of the fittest. Life and the recording of information emerged together; there is no life without information and, arguably, there is no information without life.

We do not know what the first carriers of information were when life had just emerged, but for the last few billions of years, the most important information storage medium on Earth has been DNA. Once life had emerged and started evolving, the information in DNA was accumulating, at least for a while. A few hundred thousand years ago, quite recently on the evolutionary timescale, human language emerged and a major transition began; for the first time, large amounts of information began accumulating outside DNA. Most likely, information in the world's libraries and computer clouds is now expanding faster than in the genomes of all species on Earth taken together. Even more importantly, language triggered evolutionary mechanisms different from and faster than biological evolution, namely cultural evolution. The emergence of human language was a transition as remarkable as the emergence of life itself.

Nevertheless, at least at our current state of evolution, the information in DNA is indispensable; if the DNA existing on Earth were to become too corrupted, all cultural information and life itself would swiftly disappear. But must life be like this? Or can future civilizations, possibly in thousands or millions of years colonizing planets of distant galaxies, be based on entirely different principles? Can there be another major transition in which DNA becomes less central?

This book, in which life and evolution are explored as information processing phenomena, is aimed at everybody interested in science and comfortable with elementary mathematics. It will also be of interest to students and scholars from a wide range of disciplines, from physics, computing, and biology to social sciences and philosophy. This book is for everyone who has ever wondered: how do living systems work? When do assemblies of non-living molecules become living organisms? Where does life's complexity come from? And where will life go after the Solar system ceases to exist? The fascinating idea of viewing living organisms as computers and the evolution of life as information processing will be enriching for everyone interested in our daily and long-term existence.

Acknowledgements

This book could not have been written without the support of the European Molecular Biology Laboratory (EMBL), and in particular, that of the European Bioinformatics Institute (EBI), where I have spent more than half my scientific life, gradually evolving from a mathematician into a biologist, though never really completing the transition. With its multidisciplinary focus, collaborative atmosphere, and striving for scientific excellence, EMBL and EBI provided me with the creative environment where the ideas ending up in this book could develop. But the book would also not have been possible without what I learnt about mathematics in my earlier scientific carrier, mostly at the University of Latvia, where I was fortunate enough to have been taught by some of the pioneers of theoretical computer science, and foremost, without my PhD supervisor and mentor Professor Jānis Bārzdiņš. I can trace the idea of writing this book to the early 1990s, when the Soviet Union was disintegrating and I got my first opportunity to travel outside the Soviet bloc and to work at the New Mexico State University, thanks to the kindness and generosity of Juris and Lauma Reinfelds.

I am enormously indebted to my friends and colleagues who read drafts of this book from cover to cover at various stages, and in particular to Paulis Ķikusts, Gerard Kleywegt, Robert Petryszak, Kārlis Podnieks, Uģis Sarkans, and Roderic Guigo. Without their invaluable advice and help, this book could not have been written. My wife Diana Brazma and my brother Gunars Brazma were also amongst those who read the entire manuscript and gave me important feedback. Others, in particular Nick Goldman, Mar Gonzàlez-Porta, Mark Green, Frank Holstege, Wolfgang Huber, Julio Collado-Vides, and Ainis Ulmanis, read parts of the manuscript, suggesting many important improvements. My thanks also go to anonymous reviewers who commented on the book's proposal and sample chapters. There are a larger number of colleagues and friends with whom I discussed the ideas behind this book; amongst them: Wilhelm Ansorge, Alexander Aulehla, Jurg Bahler, Pedro Beltrao, Ewan Birney, Graham Cameron, Marco Galardini, Cornelius Gross, Edith Heard, Des Higgins, Iain Mattaj, Cedric Notredame, Janet Thornton, Nassos Typas, and Rick Welch. Some of them were not aware that the questions I was asking were about an emerging book. Enormous thanks go to Andrew King, the librarian of the Michael Ashburner's Library of the Wellcome Genome Campus, who helped me chase up many obscure references central to this book. Several of my colleagues went out of their way to provide me with illustrations exactly the way I wanted them, in particular Osman Salih and Julian Hennies at EMBL, and Louis Scheffer at Janelia Research Campus. At the proposal stage I was helped by Caroline Palmer who corrected my English and Oana Stroe who helped me to write the Preface. I also want to thank my commissioning editor at OUP, Ian Sherman, for all his great encouragement and help, to my project editor, Katie Lakina, for dealing with questions as the manuscript was developing, as well as Rajeswari Azayecoche and Janet Walker, who did an admirably professional job in editing the manuscript. Any errors in this book are entirely my responsibility. Over time, my work at EMBL was funded mainly by the EMBL member states but also by many grants from the European Commission, the Wellcome Trust, the National Institute of Health, UK Research and Innovation, and others.

Contents

Introduction

> *The History of every major Galactic Civilization tends to pass through three distinct and recognizable phases, those of Survival, Inquiry and Sophistication, otherwise known as the How, Why, and Where phases. For instance, the first phase is characterized by the question 'How can we eat?', the second by the question 'Why do we eat?' and the third by the question, 'Where shall we have lunch?'*
>
> (Douglas Adams, *The Hitchhiker's Guide to The Galaxy*)

How complex does the simplest living being have to be? How many different parts it needs? Can the complexity of life be measured? As proteins are the most diverse components of all known life forms, one possible way to measure the complexity of an organism is by counting how many different protein molecules it has. Bacteria, arguably the simplest known free-living organisms, have hundreds to thousands of them. Humans have tens of thousands of different proteins, but so do other animals. Are humans unique among animals? If yes, what makes them special? On the one hand, in the number of different molecules, humans are of about the same complexity as many other animals. On the other hand, unlike other animals, humans can read, write, and play chess. But so can computers. Are humans closer to other animals or to computers?

These are not new questions. More than 2000 years ago, the Greek philosopher Aristotle thought that the difference between the living and non-living was in the presence or absence of a 'soul'. For Aristotle, there were different levels of a soul: the nutritive soul was common to all living things, whereas higher levels of soul accounted for the locomotive and perceptive capacities of animals, including humans. At the highest level there was the rational soul, unique to humans. Two-thousand years later in the seventeenth century, with the invention of the microscope, Antoni van Leeuwenhoek discovered another type of life: microbes. For a while, it was thought that microbes could spontaneously emerge from non-living matter, blurring the distinction between living and non-living. This changed when, in the nineteenth century, the French scientist Louis Pasteur demonstrated that microbes could not appear *de novo* but could only grow from the ones already in existence. The distinction between living and non-living appeared to be clear-cut again. But what exactly is the dividing line?

The developments of physics and chemistry in the nineteenth and twentieth centuries provided scientists with new ways of investigating this question, but they also imposed new demands on the acceptable answers. The answers would have to be based on atomic

Living Computers. Alvis Brazma, Oxford University Press. © Alvis Brazma (2023). DOI: 10.1093/oso/9780192871947.003.0001

theory and be consistent with the laws of physics. Towards this goal, in 1944, one of the founders of quantum mechanics, Erwin Schrödinger, wrote a book called *What is Life?*, inspiring a generation of scientists and facilitating the development of a new scientific discipline—*molecular biology*. Schrödinger described a hypothetical 'code script' that carries information passed on by living organisms from generation to generation.

Around the same time that molecular biology was emerging, the first electronic computers were invented, and a different new scientific discipline—*computer science*—was also developing. With it, the scientific concepts of *information* and *information processing* were solidifying. Computers were different from other machines in that they were performing intellectual rather than physical work. The question 'can machines think?' soon became a topic for discussion in science as well as in science fiction. To answer this question, Alan Turing, one of the founders of computer science, proposed a test: if a human talking to 'somebody' behind a curtain was not able to tell whether that 'somebody' was human or not, then by definition, that 'somebody' could think.[1] But does being able to think have to mean being alive? The converse does not seem to be true: not everything living can think.

In the 1950s the molecular structures of increasingly complex biological molecules were deciphered. The discovery of the double-helical structure of DNA in 1953 was a defining moment paving the way to understanding the molecular basis of biological information: how it is coded; how a living cell, the basic unit of all known life, uses it to function; and how this information is passed on from generation to generation. A conceptual link between biology and information/computer science began to emerge.

The initial input from computer science to biology was arguably rather limited, nevertheless computer science entered the realm of biology a few decades later, in the 1980s. This was when DNA sequencing—reading the sequences of the molecular 'letters' of DNA (reading Schrödinger's 'code script') and transcribing them onto paper or, more usefully, into a computer memory—was invented. This ability to read life's information molecule resulted in the emergence of a new scientific discipline—*bioinformatics*. Soon scientists found that analysing DNA as a sequence of letters, rather than a chemical, can reveal many secrets of life. Apparently, for life's processes, the information in DNA is at least as important as the specific chemistry. But to what extent does the particular chemistry matter at all?

All currently known life relies on the same carbon-based chemistry, which seems to suggest that chemistry matters. But is this the only way how life can exist, or was it a historic accident that life emerged on Earth based on this particular chemistry? And even if this specific chemistry was the only possible way how life could have emerged, does life have to remain based on the same chemistry in the future? Can life evolve to inhabit very different physical substances and use very different, non-carbon-based chemistry? Perhaps even no chemistry at all?

In this book I will argue that first and foremost, living means processing information. I will argue that the emergence of life and the first recording of information are intrinsically linked and perhaps are one and the same phenomenon. I will show that biological evolution can be viewed as computing where the information residing in the entire collection of DNA on Earth is processed by the *genetic algorithm* of Darwinian survival of the fittest. At least initially, as life on Earth was evolving and adapting to changing environment, information stored in DNA was growing. In a sense, life was learning from the environment, recording what it had learnt in its DNA.

For billions of years, all durable information present on Earth was almost exclusively recorded in DNA or other polymer molecules. But a few hundred thousand years ago, this changed. Human language emerged, providing an alternative means for recording and transmitting large amounts of information from individual to individual and from generation to generation. Information begun increasingly accumulating in substances different than DNA. Now, there is probably more information stored in the world's libraries and computer clouds than in the genomic DNA of all the species on Earth. Even more importantly, language enabled information processing mechanisms different from and much faster than the ones existing before. Cultural evolution, enabled by language, does not have to rely on the survival of the fittest; the information acquired over one's lifetime can be passed on to others and to future generations.

Nevertheless, at the present stage of evolution, the information in DNA and its Darwinian processing are indispensable: cultural information cannot be maintained or evolve without the information in DNA underpinning it. But can this change in the future? Could future civilizations, which in thousands or millions of years possibly be colonizing planets of distant galaxies, be based on entirely different principles, and made of entirely different molecules? Can life exist without DNA or similar molecules? Can civilization exist without life? If the main defining feature of life is information processing, then perhaps the particular substance in which this information is stored—DNA, RNA, paper, silicon chips, or anything else—becomes secondary. Or will life always need DNA and the Darwinian algorithm working in the background? What will the life eventually escaping the dying solar system be like?

The appreciation of the role of information and information processing in biology can be traced back to the mentioned Erwin Schrödinger's book *What is Life?* and to publications of one of the co-discoverers of the structure of DNA, Francis Crick. The observation of the central role of information in biology is at the basis of the so-called *gene's-eye view* introduced by the American and British biologists George C. Williams and Richard Dawkins, stating that *genes*—pieces of heritable information—are the main object of Darwinian selection of the fittest.[2] But a special place in the development of the idea of information as a central concept in biology belongs to the British biologist John Maynard Smith. In his seminal book *The Major Transitions in Evolution*,[3] written with the Hungarian biologist Eörs Szathmáry and published in 1995, the authors show that the key events in biological evolution can be seen as transitions in how life processes information.

With genome sequencing now a commodity and bioinformatics becoming a mainstream science, the central role of the concept of information in biology is becoming increasingly accepted. In his recent book, *The Demon in the Machine*,[4] the physicist Paul Davies explores the links between the concepts of entropy, information, and life, arguing that information processing is at the very centre of life. Even more recently, the Nobel

Laureate Paul Nurse writes about *life as chemistry* and *life as information* in his popular science book that has the same title as Schrödinger's *What is Life?*.[5]

In this book, *Living Computers*, I discuss how living systems work, how life and information emerged already intrinsically linked, how life and information processing evolved through RNA and DNA to neural networks and the animal brain, and continued through human language to kick-start a new kind of evolution. Throughout the book, I try to be clear where I am describing the consensus with which most scientists would agree, note where I am touching controversies, and state where I am just speculating (and to make sure that my claims can be verified, I provide many references to peer-reviewed literature). Most of the book concerns the first, consensus positions, even though I often try to show the established knowledge in new light, and I sometimes question an odd orthodox assumption. Not everything in the book is a part of the established scientific consensus. The thesis that life and the recording of information emerged jointly seems obvious to me, nevertheless there is no reference to scientific literature that I can find where this had been stated explicitly. I argue that large amounts of information can be recorded and processed only by digital-combinatorial means, which may also sound controversial to some. This then further leads to the conclusion that before human language emerged, the only means to record large amounts of information were in polymer molecules such as DNA. Probably, the most speculative is my thesis that the new means of information recording and processing resulting from the emergence of human language can take evolution in entirely new directions. But about the future of life, clearly, we can only speculate.

1

How to Clone Oneself?

Her Majesty pointed to a clock and said, 'See to it that it produces offspring'.

(Anonymous account of an alleged discussion between René Descartes and Queen Christina of Sweden, 1649–1650)

Is it possible to make a machine that clones itself? A machine which when set in motion, goes on and assembles its own replica? In other words, a machine that reproduces? The ability to reproduce is often seen as one of the principal dividing lines between living and non-living. Allegedly, when in the seventeenth century, the French philosopher and mathematician René Descartes in a conversation with Queen Christina of Sweden suggested that an animal body could be regarded as a machine, the Queen replied that then a clock should be able to produce offspring. Obviously, clocks do not do this, but is it possible to build an automaton or a robot that can clone itself? In other words, an automaton that can *self-reproduce*.

Regardless of whether this conversation really took place, or how accurate the narrative is, it is quite likely that the phenomenon of *self-reproduction* or *self-replication* has been occupying the minds of thinkers and philosophers for centuries. However, the first truly scientific investigation of this question apparently began only with the onset of the computer age. In June 1948, the Hungarian-born American mathematician John von Neumann (Figure 1.1) gave three lectures to a small group of friends at the Institute for Advanced Study at Princeton discussing how to build a mechanical self-reproducing machine.[1]

By that time, von Neumann was already an established scientist, having made major contributions to quantum mechanics, statistical physics, game theory, and the foundations of mathematics, amongst other disciplines. In early 1940s, von Neumann joined the project at the University of Pennsylvania's Moore School of Electrical Engineering, to design the Electronic Discrete Variable Arithmetic Computer—EDVAC.[2] Von Neumann's *First Draft of a Report on the EDVAC*, published in 1945, is now widely regarded as the first description of a stored-program computer—a computer which is given its operating instructions by software, as all computers now are.[3]

In this report, von Neumann abstracted from the specific electronic parts, such as electron vacuum tubes or electric relays, used in computers at the time, and instead talked about idealized *logical switches*. He used an analogy between a computer and brain, comparing his logical switches to neurons. He thought that the specific physical

Living Computers. Alvis Brazma, Oxford University Press. © Alvis Brazma (2023). DOI: 10.1093/oso/9780192871947.003.0002

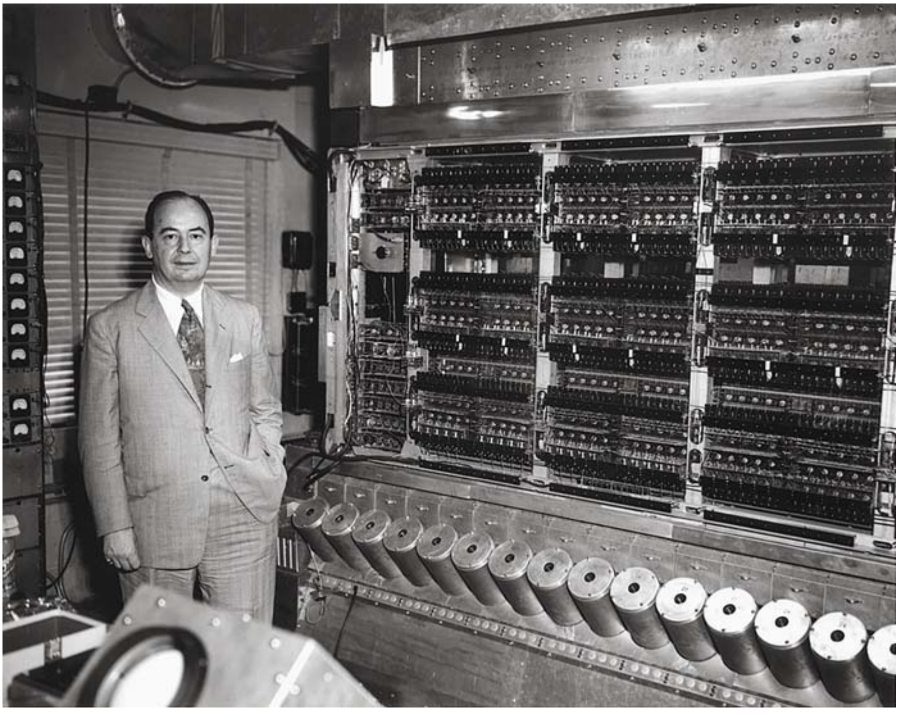

Figure 1.1 *John von Neumann standing in front of computer.*
Photographer, Alan Richards. From the Shelby White and Leon Levy Archives Center, Institute for Advanced Study, Princeton (NJ).

implementation of the switches could soon change, and he was right: it was not long before the transistor was invented and replaced the vacuum tube.[4] For von Neumann, the logical basis of computing was more fundamental than the specific physical implementations. In 1957, less than ten years after giving the mentioned lectures on self-reproduction, at the age of 53, von Neumann died from cancer.

Around the same time when the first electronic computers were being built, in the 1940s and 1950s, geneticists and biochemists were closing in on understanding the principles of the molecular basis of life. Although von Neumann did not play a major part in this effort, he was discussing his ideas with the leading biochemists and geneticists, and when working on the problems of computing, he always kept returning to the computer–brain analogy. It is quite likely that von Neumann's interest in self-reproduction was motivated by what he saw as the analogy between machines and living organisms, and by his desire to understand the 'logical' basis of life.

For von Neumann, it was clear that a self-reproducing device was possible; after all, living organisms reproduce. For him the question was how to make such a device and how complex the simplest possible self-reproducing system would have to be. Von Neumann noticed a paradox. For a device A to be able to build another device B, device A would have to contain, in some sense, a description of B. If so, it appeared that A would have to be more complex than B. Thus, a device could only build devices simpler than itself. Von Neumann called this phenomenon the 'degenerative trend'. But this contradicted the observations that living organisms not only manage to produce descendants as complex as themselves but, through evolution, also increasingly complex ones. How could a device build something as complex or even more complex as itself?

Looking for a solution, von Neumann came up with the concept of the *Universal Constructor*—an abstraction of a general-purpose robot which, given appropriate instructions and the necessary materials, could build whatever it has been instructed to build. The Universal Constructor could be given instructions to build itself, though to complete the reproduction cycle the new device would also have to be equipped with a new copy of the respective instructions. Von Neumann argued that the Universal Constructor would inevitably have to be quite a complex device, and consequently, a self-reproducing machine—a robot that clones itself—could not be simple. He also speculated how random errors in such self-reproducing devices could potentially lead to evolution towards increasing complexity. Thus, von Neumann postulated that there is a complexity dividing line below which only the degenerative trend could occur but above which evolution towards increasing complexity was possible.[5]

A mechanical replicator

In the mentioned lectures at Princeton, and a year later in the Hixon Symposium on *Cerebral Mechanisms in Behavior*,[6] von Neumann discussed the general principles at the basis of such a mechanical self-reproducing device. He called this a *kinematic* replicator. *Kinematics* is a branch of physics that studies motion of bodies and their systems. Von Neumann was interested in self-reproduction in its 'logical form' and wanted to abstract from problems such as supply of energy, the nature of molecular forces, or chemistry. Therefore, the building blocks of the replicator were allowed to be more complex than individual '*molecules, atoms or elementary particles*'. In a lecture at the University of Illinois in 1949 he said:

> *Draw up a list of unambiguously defined elementary parts. Imagine that there is a practically unlimited supply of these parts floating around in a large container. One can imagine an automaton functioning in the following manner: It also is floating around in this medium; its essential activity is to pick up parts and put them together . . . .*[7]

This process should result in a new copy of the automaton being built. Von Neumann suggested a list of eight different *elementary parts*, which he called the *stimulus producer*, *rigid member*, *coincidence organ*, *stimuli organ*, *inhibitory organ*, *fusing organ*, *cutting organ*, and *muscle*.[8] The *stimulus producer* served as a source of discrete pulses of signals to make the other parts act. The *rigid member* was the physical building block from which the frame of the automaton was constructed. The next three 'organs'—the *coincidence*, *stimuli*, and *inhibitory organs* were what we would now call the *logical gates* AND, OR, and NOT—the basic building blocks of a computer.[9] Von Neumann used these elements to design the control module—the brain of the replicator, which processed the signals generated by the stimulus organ to guide the construction. The last three organs were there to follow the instructions of this control module and to perform the actual physical construction. The *fusing organ*, when stimulated by a signal, soldered two parts together; a *cutting organ*, when stimulated, unsoldered a connection; and finally, the *muscle* was used to move the parts. The automaton built from these elements would be 'floating in a container' alongside spare parts and following the instructions recorded on a tape, telling it which parts to pick up and how to assemble them.

On a closer look, there are three main ideas at the basis of von Neumann's self-reproducing automaton. The first comes from a simple observation that if one has to copy a complex three-dimensional object, for instance to make a replica of a historic steam-engine, it is usually easier first to make a blueprint of this object and then build by the blueprint, rather than to try to copy the three-dimensional object directly. This, indeed, is how things are usually manufactured. It may also help to supplement the blueprint with a sequence of instructions describing the steps of the construction process. For instance, flat-pack furniture usually comes with a blueprint and an instruction manual describing the order in which the pieces should be put together. If these instructions are to be executed by a robot, then every step will need to be encoded in a precisely defined way. Nowadays, such instructions would normally be stored on some electronic information carrier, such as a *flash* memory-card, or streamed wirelessly, but in the early days of computing, a punched tape, such as the one shown in Figure 1.2, would be used. Conceptually, such a tape can be regarded as a one-dimensional sequence of symbols encoding the instructions.

The second idea represents another simple observation that it is easier to copy a one- or two-dimensional object, such as a punch-tape or this printed page, than a three-dimensional object. For instance, to copy the punch-tape in Figure 1.2, one could put it on top of an unused (unperforated) tape and punch holes where they already are in the original. It should not be too difficult to build a purely mechanical device automating this process.

Von Neumann described a device that would mechanically copy a chain encoding instructions as a sequence of two different types of links, such as shown in Figure 1.3A. Conceptually, such a chain can be viewed as a sequence of 0s and 1s, and as we will see later, any piece of information can be encoded as such a binary string. In fact, it may be even easier to copy a chain if it is designed as two 'complementary' chains, as shown in Figure 1.3B, each unambiguously defining the complement. We simply need to decouple the complementary chains and use each as a template to make a new complement, as shown in Figure 1.3C. Once the process is finished, we have two copies of the original chain. A reader with some knowledge of molecular biology may notice a similarity between this method and how a DNA molecule is copied in a living cell.

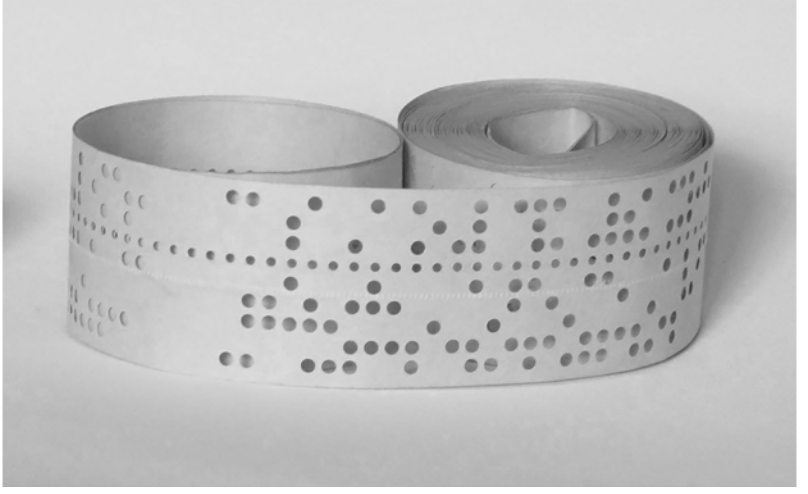

Figure 1.2 *Punched (perforated) tapes were widely used in the 1950s and the 1960s in electronic computers for data input. Although such a tape is essentially a two-dimensional object, abstractly we can view it as a one-dimensional sequence of symbols.*
(Adapted from Wikimedia, author Ted Coles.)

The third idea, the most fundamental and non-trivial of the three, was that of the already mentioned Universal Constructor. Von Neumann assumed that this machine would read a sequence of instructions given to it encoded on a tape or as a chain of elements as shown in Figure 1.3, and by following these instructions, it would build any specified three-dimensional object. This object could be another machine. A robot able to read instructions given to it on a perforated tape had already been invented in the eighteenth century by the French weaver, Basile Bouchon of Lyon. Bouchon built a loom that followed instructions encoded on such a tape, translating them into weaving patterns. Of course, that was not a general-purpose robot but a loom. One can say that von Neumann's Universal Constructor was an extension of Bouchon's loom towards Turing's Universal Machine, discussed later in this chapter.

Let us see how a combination of these three ideas can be used to make a self-reproducing device. I will roughly follow von Neumann's description, which is somewhat abstract, but it will later help us to see the analogy between his self-reproducing automaton and the main principles how a living cell replicates.

Suppose X is an arbitrary object made of von Neumann's eight elementary parts. This object, just like any object, can be manufactured by the Universal Constructor, therefore there exists a sequence of instructions following which the Universal Constructor produces X. Let us denote this sequence of instructions by code(X). Conceptually code(X) is a sequence of 0s and 1s, physically this is encoded ono a perforated tape or any way which this Universal Constructor can read. What matters is that on the input of code(X), the Universal Constructor outputs object X.

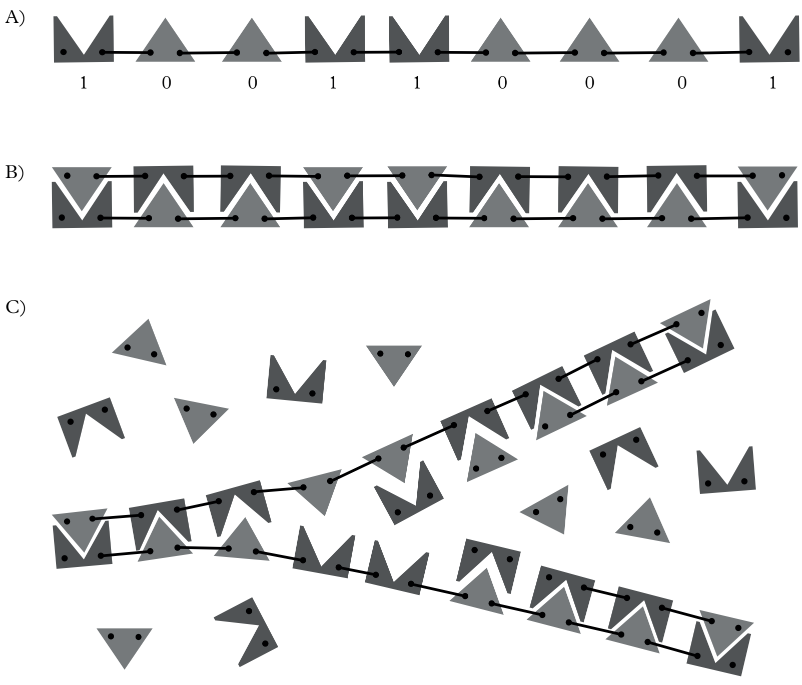

Figure 1.3 A) *A chain of two different types of links encoding a binary string.* B) *Two 'complementary' chains of such links forming a double chain.* C) *A possible mechanism of information copying that exploits complementarity: the complementary chains are separated, and a new compliment is assembled onto each of them from 'free' links 'floating' in the media. Once both chains are fully separated and the complements assembled, the neighbouring elements of the newly assembled complements only need to be 'stapled' together and a new copy of the entire chain is obtained. It is not clear if such complementarity based approach had occurred to von Neumann; he does not mention this possibility. As we will discuss in* *Chapter III**, in DNA replication a similar mechanism is used.*

For brevity, let us denote the Universal Constructor by U. To complete the construction of a self-reproducing machine, in addition to U we need two other, simpler robotic devices, which we denote by R and C. The device R, called *code replicator*, given the tape code(X), makes a new copy of code(X). That is, R makes another copy of the tape containing the same instructions. As already noted, copying a tape is quite a simple procedure.

The third robot C controls both U and R in the following way. Given code(X), C first runs replicator R, thus making a new copy of code(X). Then C runs U on code(X), thus producing object X. Finally, C attaches the new copy of code(X) to the newly created X and releases the new object X+code(X). In von Neumann's own words, it 'cuts the new object loose'. Here, following von Neumann, we use symbol '+' to denote that the two objects are somehow 'tied' together.

Now start with the combined robot U+R+C+code(X). First, C instructs U to run on code(X). The result is:

U+R+C+code(X), X

Next, C instructs R to make another code(X); the result is:

U+R+C+code(X), X, code(X)

Finally, U 'ties' X and code(X) together; the result is:

U+R+C+code(X), X+code(X)

But now recall that X was an arbitrary object. In particular, X can be some sort of a device, for instance, we can take U+R+C for X (that is, X=U+R+C). If we do this, and give code(U+R+C) to our three-part robot as an input, we start with:

U+R+C+code(U+R+C)

and we end with:

U+R+C+code(U+R+C), U+R+C+code(U+R+C)

The robot U+R+C+code(U+R+C) has managed to build its own copy; it has reproduced itself. The 'degenerative trend' has been broken—we have a robot that built another robot as complex as itself. One may wonder how exactly it did this, but it did. We needed two separate devices: the Universal Constructor U and code replicator R. Could the Universal Constructor do both? For von Neumann's idea to work, the process R of replicating the tape had to be carried out separately from process U (a sceptic reader can try to merge both processes into one and see the difficulties). As we will see later, this is also what happens when a living cell divides; the tape there is the genomic DNA, which is a polymer 'chain' of four different elements. DNA replication is a process that is implemented in a living cell by a distinct machinery.

Note that the object U+R+C+code(U+R+C) is an object that contains its own description: code(U+R+C) describes the device U+R+C. This is what has allowed it to beat the 'degenerative trend'. But this also turns out to be at the basis of life. As Sydney Brenner, who shared the Nobel Prize in physiology and medicine with Robert Horvitz and John Sulston in 2002 'for their discoveries concerning genetic regulation of organ development and programmed cell death', wrote in 2012:

> *Arguably the best examples of Turing's and von Neumann's machines are to be found in biology. Nowhere else are there such complicated systems, in which every organism contains an internal description of itself.*[10]

What von Neumann described in his lectures was only an idea; he did not build a physical mechanical self-reproducing machine. The complexity of such a machine would have been an enormous engineering challenge in 1950s and still is today. Some parts of von Neumann's device, as he proposed it, have been built, and suggestions have been made that completing the cycle must be possible. However, as the English say: 'the proof of the pudding is in the eating'. The kinematical replicator is still not there and, as we will discuss later, possibly for a good reason—the number of mechanical steps such a device would have to carry out to complete the reproduction cycle may be prohibitive. To prove that self-reproduction is logically possible, as a mathematician, von Neumann turned to developing a rigorous mathematical model.

Self-reproduction without hand-waving

In experimental sciences, the ultimate proof that an apparatus does what it is claimed to do is an experiment. Will the machine build its replica or not? But even in experimental sciences, non-trivial reasoning may be needed to conclude that the experiment has indeed proven what is claimed. Is the replica authentic? Was it the machine that cloned itself or was it the complex environment around it that made the copy? In mathematics, the proof is achieved by defining the problem and the rules for solving it rigorously, and then finding a solution within the rules. The reasoning that we used to argue how von Neumann's kinematic replicator built its own copy did not quite fit this bill of rigour. For instance, the + sign was not defined rigorously. Obviously, this was not an arithmetic addition, it was a kind of attachment between the objects, not defined precisely. Can we define and solve the problem of self-reproduction mathematically? What is the simplest mathematical model in which the phenomenon of self-reproduction can be described rigorously—without what mathematicians call *hand-waving*?

Those who have studied computer programming may have come across an exercise: write a code that prints its own text. Or to be precise, a code that instructs the computer to print a text identical to this code. Such a program is sometimes referred to as a *quine*.[11] In computer programming it is difficult to cheat—if the program does not do the job, no amount of hand-waving will make it work. Those familiar with computer programming can check that coding a *quine* is a non-trivial task. However, it was proven by one of the founders of computer science, the US mathematician Stephen Kleene, that it is possible to code a quine in any general enough programming language. An in-depth discussion of quines and their relevance to biology can be found in the classic book *Gödel, Escher, Bach* by Douglas Hofstadter.[12]

We can modify a *quine* so that instead of printing the text, the program copies its own code over a computer network to another computer. This is what computer viruses do. A computer virus is a self-reproducing code. Does such a virus achieve what von Neuman set out to do? Only in a rather limited sense. A computer virus replicates because the code is executed by computer hardware using an operating system, which usually is more complex than the code of the virus itself. Those who are familiar with biology will notice that the same is true in respect to biological viruses—to replicate viruses need the machinery of a living cell, which typically is more complex than the virus. Von Neumann's kinematic replicator was supposed to 'float' in an environment simpler than itself.

To have a full analogue to *quine* in the physical world of atoms and molecules, we would need to have a program for a computer equipped with a three-dimensional printer (or a robot), which in addition to copying itself would also have to instruct the computer and printer to 'print' another computer and printer, and finally load itself into the new computer. This would be a version of von Neumann's self-reproducing device, where the program would be analogous to its tape.

As von Neumann noted in his lectures, computers output a class of objects that is rather different than themselves—they can print a text or send a code electronically to another device but they do not make physical objects. Von Neumann wanted to find a mathematical model of a device that would make an object of the same class as itself.

Inspired by his friend and colleague Polish-American mathematician Stanislaw Ulam, von Neumann turned to *cellular automata*—two dimensional structures or configurations (drawn on a plane) that develop according to well-defined rules. The goal was to find a set of rules and a particular initial structure such that after some time, two structures identical to the original would emerge on the plane.

Cellular automata and *Life*

Probably the best-known example of cellular automata is the *Game of Life*, or simply *Life*, invented by the British mathematician John Horton Conway at Cambridge University. This game was introduced to the general public in the 1970 by Martin Gardner, the legendary editor of the column *Mathematical Games* in the popular science magazine *Scientific American*.[13] At that time, computers were becoming cheaper and accessible to many scientists, and increasingly to everyone interested, which most likely contributed to the popularity of the game. Now there are many websites offering *Life* online; to see what fun *Life* can be, the reader not yet familiar with this game might find and try one of these websites out. Less well known than the game itself is that Conway invented *Life* thinking about von Neumann's problem of self-reproduction. In 1986, after leaving Cambridge, Conway became the John von Neumann Chair of Mathematics at Princeton University.

The *Game of Life*, unlike real life, is governed by simple rules, but like real life, it is very rich in possibilities. Although to see the richness of *Life* one needs a computer, it can be played on a sheet of squared paper by one player. Conceptually the sheet is of infinite size, as it can always be extended as necessary by adding another sheet. Each cell can be in one of two *states*; occupied—*alive*, or empty—*dead* (*quiescent* state). Each square is a cell with eight neighbours—one to the left, one to the right, on top, beneath, and four diagonally. The creative part of the game is to fill in the initial configuration—to choose which cells are *alive* at the start. After that, the configuration develops following strict deterministic rules in discrete steps or time-points. For the description of *Life*'s rules and how some initial configurations develop see Figure 1.4.

Amongst the simplest configurations in the figure, *glider* takes a special place: every five steps *glider* rebuilds itself, shifted by one position, thus moving diagonally on the grid. *Gosper glider gun* or *cannon* is a more complex configuration, which periodically

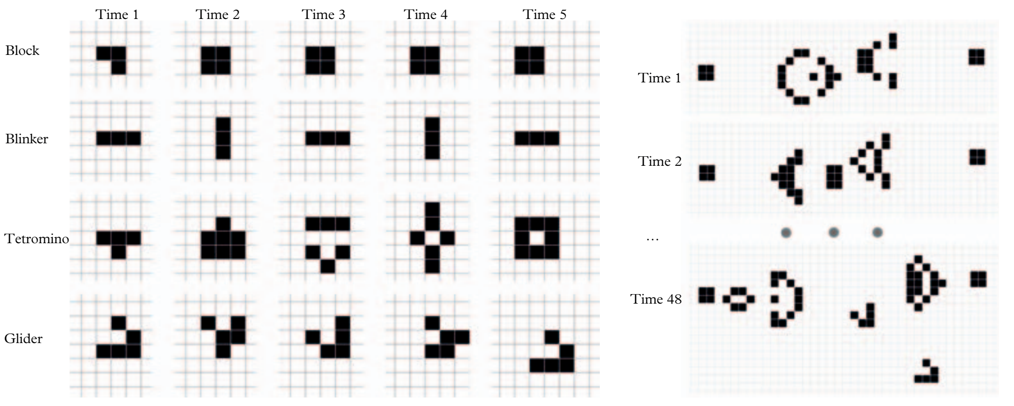

Figure 1.4 ***Left.*** *Simple configurations in Conway's Game of Life and how they develop. At every consecutive time-point, a new configuration is drawn by the following rules. A cell is* alive *(filled) in one of two cases only: 1) it was* alive *in the previous time-point, and it had two or three* alive *neighbours, or 2) it was* not alive *(was empty) in the previous time-point and had exactly three* alive *neighbours. (Thus, exactly three* alive *neighbours of an empty cell give birth to a new* alive *cell, while if an* alive *cell has less than two or more than three* alive *neighbours, it* dies *of 'lonesomeness' or 'overcrowding', respectively.) Following these rules, a block of four cells stays intact. A row of three consecutive cells (*blinker*) periodically turns itself into a column of three cells and back. A* glider *moves diagonally through space—it is easy to check that after four steps it will have reconstructed itself by one position to the right and downwards. The reader may try out what happens to* tetromino *after time 5.* ***Right.*** Gosper glider gun *(or* cannon*) periodically shoots gliders. The first glider appears at time-point 16, and then again, every consecutive 20 time-points. To appreciate the richness of such configurations in full, one needs to use a computer (for instance, Wikipedia offers an interactive website).*

shoots *gliders* in a particular direction. It may be exciting to watch on a computer screen what happens when two cannons are shooting gliders at each other; depending on their relative position, impressive fireworks can happen.

Most *Life* configurations will eventually become static (like *block* in Figure 1.4) or get into a loop of some periodicity (like *blinker*). These can be viewed as *settled* configurations as nothing new can happen. Some configurations, however, may go on developing in a nontrivial way forever; for a mathematically minded person it can be an exciting exercise to try to find some of them. In general, it is not possible to tell if a configuration will settle or not in any other way than following its development until it settles, or maybe it does not. If it settles, we can spot this, but it is impossible to know how long one should keep following a configuration that refuses to settle. This is a mathematical fact (a theorem).[14]

But here we are primarily interested in cellular automata as a model for *self-reproduction*—is there a configuration such that following the rules, at some point two configurations identical to the original would emerge?

Self-reproducing structures

To build a model for a self-reproducing device, von Neumann used a version of cellular automata different from Conway's *Life*. Cells in his automata had 29 different states[15] rather than two, as in *Life*. His construction, schematically shown in Figure 1.5, was designed to emulate a two-dimensional robot.[16] The robot had a 'construction arm', which performed the actual task of assembling a new device, like a robotic arm would be used to build a physical structure. Von Neumann described how to simulate various elements of this robot, including wires through which control signals were sent. These signals were processed in the control units built from the logical gates (also implemented

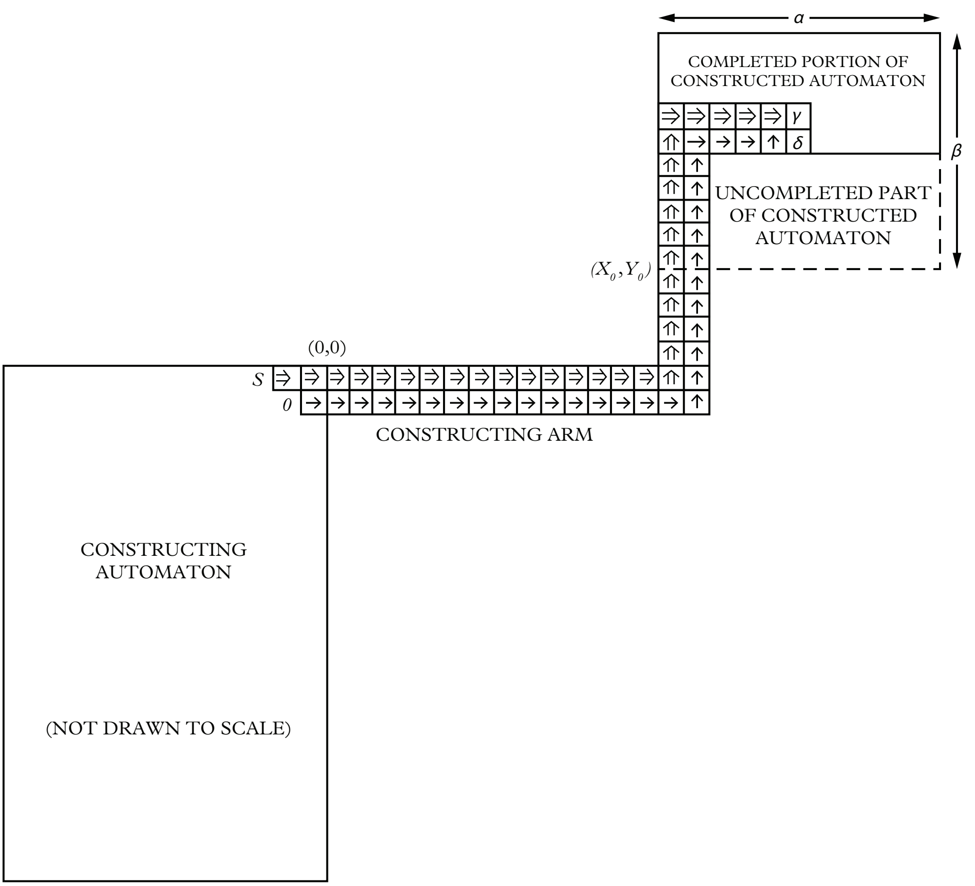

Figure 1.5 *The tessellation structure schematically representing von Neumann's Universal Constructor* (From Burks, 1970).

as cellular automata configurations), mentioned earlier in this chapter. Von Neumann described how a one-dimensional tape could be simulated using his cellular automata, and how this tape could be copied. He sketched how these components could be assembled to make the Universal Constructor, which would read instructions from the 'tape' to build a new two-dimensional object, all following the rules.

Von Neumann realized that creating a device that is already operating while it is still being built would be rather difficult in this framework; its operation would interfere with its further construction. This would be like trying to manufacture a car with the engine already running. The solution he found was first to build a static device and then, when its construction was finished, it was activated by an 'ignition' signal. This is like Frankenstein's creation—the body was first assembled and then set alive by an electric spark. To implement this 'setting alive' idea, every cell needed an unexcitable state and a live state that was induced by the ignition signal, which is one of the reasons why von Neumann needed cells with as many as 29 states.[17]

Von Neumann's two-dimensional self-reproducing robot was quite a baroque construction, which he did not finish; his collaborator Arthur W. Burkes finished it already after von Neumann's death.[18] Burkes too did not draw the construction explicitly but designed its various elements and gave a mathematical proof that the Universal Constructor and a self-reproducing configuration could be built from these. Together this proof filled more than 200 pages. Burkes later simplified the design to the extent that its description could fit on 64 pages.[19] In 1955, another of von Neumann's collaborators, John Kemeny, wrote an article in *Scientific American* estimating that a self-reproducing automaton based on von Neuman's principles could fit into a $80 \times 400 = 32{,}000$-cell rectangle, plus a 150,000-cell long 'tail' representing the tape.[20] Some other estimates range from 50,000 to 200,000 cells.[21]

Although various further simplifications have been reported, little has been published in peer-reviewed scientific literature since the 1970s. It seems that none of the designs has been fully implemented as a computer program allowing for observable simulations.[22] Perhaps one reason for this is that the answer to the most important fundamental question was clear from von Neumann and colleagues' early work: self-reproduction is logically possible, but the structures required to achieve this are not simple. The exact versions of the possible structures are less important as they do not model the laws of physics, and thus what they can tell us about physical self-reproducing devices is limited.

Are there simpler ways of achieving self-reproduction if we do not follow von Neumann's design? In Chapter II we will look at a simple form of self-reproduction found in nature: growing crystals. Whether a crystal growth can be truly viewed as self-reproduction is debatable, but as we will see there is some analogy. Atoms or molecules form periodic crystal lattices the structure of which is reproduced when crystal is growing. Unsurprisingly, it is not difficult to design cellular automata that grows in a way resembling a growing crystal lattice,[23] as shown in Figure 1.6.

An important feature that distinguishes a growing crystal from a living organism is in the amount of information that each of these can carry. The repetitive structures of a pure periodic crystal, as we will discuss in Chapter II, cannot carry much information.

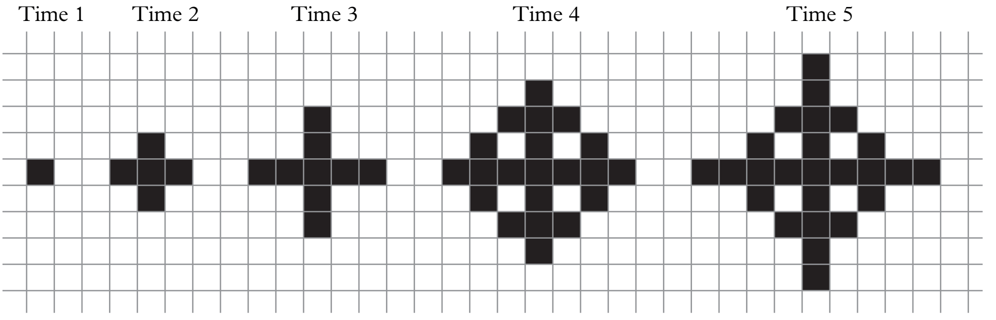

Figure 1.6 *A simple growing tessellation structure. Here we assume that a cell has four neighbours—the cell above, beneath, left, and right—and in that in each next generation a* live *cell is born if and only if it is a neighbour to exactly one* live *cell in the previous time-point. Following such a rule, starting from one* live *cell, the structure will keep growing like a snowflake. The real snowflakes have hexagonal-symmetry, which can be modelled on a honeycomb-like grid.*

In contrast, biological organisms carry large amounts of information, and moreover, like von Neumann's automata, their replication is guided by this information. John Maynard Smith and Eors Szthmary, in their seminal book *Major Transitions in Evolution*,[24] refer to *unlimited heritability*, contrasting it to *limited heritability*. A living organism can carry and pass on potentially unrestricted amounts of information to their descendants,[25] pure periodic crystals cannot. It seems that von Neumann had discovered a similar distinction in the mathematical 'world' of cellular automata a couple of years before biologists uncovered the mechanisms of inheritance. But what is *information*?

The world of information *bits* and *bytes*

Information, like many fundamental concepts, is hard to define. It can be argued that a definition that includes all important features of what we intuitively understand by this concept has not yet been found. Intuitively, most of us will agree that information is something that reduces uncertainty. Reduction in uncertainty can be quantified; 1 *bit* is the amount of information needed to answer one *yes/no* question in absence of prior knowledge. A *byte*—a unit used to measure the information storage capacity of a computer or a size of a file—equals 8 *bits*, thus providing sufficient information to answer eight *yes/no* questions.

As will be argued in Chapter VII, any piece of information can be encoded as a sequence of answers to *yes/no* questions. If we use 1 for *yes* and 0 for *no*, then a series of such answers can be encoded as a sequence of 0s and 1s, for instance, as strings like 00101101, which are often referred to as *bit-strings*. Sometimes using a larger alphabet, for instance, Latin letters or Chinese characters, or perhaps pixels of different colours, may be more convenient way to present a particular piece of information. In DNA information is encoded as a sequence of four different small molecules—four nucleotides—traditionally denoted by letters A, T, G, and C, as will be discussed in Chapter III. Nevertheless, sequences in larger alphabets can always be represented as *bit-strings*. In DNA each nucleotide contains 2 *bits* of information, as it encodes an answer to two consecutive yes/no questions. For instance, A can be represented as 00, T as 01, G as 10, and C as 11. We may also want to arrange the characters in more than one dimension, for instance as two-dimensional configurations of the *Game of Life*, or as pixels to represent an image in a digital photography. Multi-dimensional symbol arrangements too can always be represented as *bit-strings*: conceptually, everything we see on a computer screen is encoded in as a *bit-string* in the computer's memory. Some may ask—what about analogue information, for instance film photography? We will discuss this question in some more detail in Chapter VII, but the short answer is that such information too can be encoded as sequences of 0s and 1s.

Is there a distinction between a random sequence of 0s and 1s, and what we would normally call *information*? Establishing such a distinction is the core challenge in defining information in a way that reflects our intuitive understanding of this concept. A potential answer lies in acknowledging that *we can talk about information in a meaningful way only in the context of an interpreter*. To talk about information, there has to be a mechanism that converts the particular sequences of symbols representing this information into something else, for instance, into some kind of action. Von Neumann's Universal Constructor reads the information from the tape and builds an object. Those familiar with molecular biology will know that in a living cell molecular machinery reads information in DNA to make proteins. A human can read information in a cookbook to prepare dinner or a molecular biology textbook to get ready for an exam. Thus, let me postulate that *a string or an arrangement of characters is information if and only if it can be interpreted.* Though I have not seen such a postulation formulated in the literature exactly this way, it is very much in line how other authors have treated this concept. This is consistent with the approach taken by Douglas R. Hofstadter in his famous book *Gödel, Escher, Bach*, though Hofstadter does not explicitly refer to term information.[26] Obviously, as the concept of interpreter is not defined, I do not claim here to have provided a new formal definition of information. We will return to this discussion in Chapter VII.

Finally, note that although information is defined as an arrangement of symbols, in the physical world information exists only as a material pattern. The same information can be encoded by different physical means and as different patterns, for instance as an arrangement of symbols printed on a paper or an arrangement of holes in a perforated tape. The physical interpreters will have to be different, but the result of their action can be the same; for instance weaving a rug containing the same pattern.

Church–Turing thesis

Paradoxically, it is easier to define *information processing* than to define information. *Information processing is what computers do*; most scientists working in the field will agree with this statement. Examples of information processing can be as simple as the adding