Chapter 10
Computational Technosphere and Cellular Engineering

Mark Burgin
Department of Mathematics, University of California, USA

Abstract. The basic engineering problem is to build useful systems from given materials and with given tools. Here we explore this problem in the computational technosphere of computers, smartphones, networks and other information processing and communication devices created by people. The emphasis is on construction of different kinds of information processing automata by means of cellular automata. We call this engineering problem cellular engineering. Various types and levels of computing systems and models are considered in the context of cellular engineering.

Keywords: cellular automaton, computational equivalence, engineering, modeling, construction, model of computation, grid automaton.

1 Introduction

Stephen Wolfram [11] suggested the Principle of Computational Equivalence, which asserts that systems found in the natural world can perform computations up to a maximal ("universal") level of computational power, and that most systems do in fact attain this maximal level of computational power. Consequently, most systems performing recursive computations are computationally equivalent in general and equivalent to cellular automata in particular. Here we consider a technological counterpart of this Principle, which is related not to nature but to the technosphere created by people. The technosphere is the world of all technical devices. In it, computers and other information processing systems play the leading role. Taking all these devices, we obtain the computational technosphere, which is an important part of the technosphere as a whole. The computational technosphere has its own Principle of Computational Equivalence. It is called the Church-Turing Thesis. There are different versions of this Thesis. In its original form, it states that the informal notion of algorithm is equivalent to the concept of a Turing machine (the Turing’s version) or that any computable function is a partial recursive function (the Church’s version). The domineering opinion is that the Thesis is true as it has been supported by numerous arguments and examples. As a result, the Church-Turing Thesis has become the central pillar of computer science and implicitly one of the cornerstones of mathematics as it separates provable propositions from those that are not provable. In spite of
all supportive evidence and its usefulness for proving various theoretical results in computer science and mathematics, different researchers, at first, expressed negative opinion with respect to validity of the Church-Turing Thesis, and then build more powerful models of algorithms and computations, which disproved this Thesis. It is possible to find the history of these explorations in [6]. Here we go beyond the computational technosphere, suggesting the Technological Principle of Computational Equivalence for the whole technosphere. It asserts:

For any technical system, there is an equivalent cellular automaton.

This principle also has a constructive form:

For any technical system, it is possible to build (find) an equivalent cellular automaton.

Here we consider only the computational form of the Technological Principle of Computational Equivalence. It is expressed as the Computational Principle of Technological Equivalence:

For any information processing system, it is possible to build (find) an equivalent cellular automaton.

Note that in this Principle cellular automata are not restricted to classical cellular automata. There are much more powerful cellular automata. For instance, inductive cellular automata can solve much more problems than classical cellular automata or Turing machines. Building technical systems is an engineering problem. That is why in Section 2, we discuss computational engineering, which is rooted in the work of von Neumann who used a special kind of computational engineering, or more exactly, cellular engineering, for building self reproducing automata [28]. He also demonstrated that construction of complex systems using cellular automata allows one to essentially increase reliability of these systems. However, to be able to rigorously demonstrate validity of the Computational Principle of Technological Equivalence, as well as of the Technological Principle of Computational Equivalence and Wolfram’s Principle of Computational Equivalence, it is necessary to ascribe exact meaning to terms used in these principles. That is why in Section 3, we introduce and analyze different types of computational and system equivalence. In Section 4, we demonstrate possibilities of cellular engineering in modeling and construction, giving supporting evidence for the Computational Principle of Technological Equivalence. Some of these results were obtained in [7], while other results are new.

2 Computational Engineering

It is possible to describe an engineering problem in the following way. Given working materials and tools for operation, build/construct a system (object) that satisfies given conditions. Here we consider a specification of such a problem for computational (information processing) systems. Thus, we have the following initial conditions: