FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 43 (2018) No. 3
ISSN 0867-6356
DOI: 10.1515/fcds-2018-0010 e-ISSN 2300-3405
sciendo

Universes and simulations: Civilizational development in nested embedding

Maciej Komosinski [*]

Abstract. The rapid development of technology has allowed computer simulations to become routinely used in an increasing number of fields of science. These simulations become more and more realistic, and their energetic efficiency grows due to progress in computer hardware and software. As humans merge with machines via implants, brain-computer interfaces and increased activity involving information instead of material objects, philosophical concepts and theoretical considerations on the nature of reality are beginning to concern practical, working models and testable virtual environments. This article discusses how simulation is understood and employed in computer science today, how software, hardware and the physical universe unify, how simulated realities are embedded one in another, how complicated it can get in application, practical scenarios, and the possible consequences of these situations. A number of basic properties of universes and simulations in such multiply nested structures are reviewed, and the relationship of these properties with a level of civilizational development is explored.

Keywords: reality, simulation, embedding, perception, complexity

1. Introduction

The thought that our universe may be a simulation run by some other entity has been suggested in many fields of science, including physics [6, 29, 70, 52, 3], computer science [17, 73, 37] and philosophy [56, 10, 11, 43, 27]. The idea that our world is "something other" than what we perceive – which is a milder assumption than it actually being a simulation, but may suggest that it is – has a long history. Such ideas were common in philosophy (Zhuangzi's consideration of dreams and reality from the 4th century BC and further ontological analyses by Plato, Aristotle, and Descartes), in the sciences [9, 58, 45, 24], in art and literature [28, 54], and also in everyday life.

[*] Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland. Email: `maciej.komosinski@cs.put.poznan.pl`

These concepts have now become so popular in art and popular culture [46, 49, 15, 74, 63, 64, 38] that notions of “the matrix”, “virtual reality”, “a game”, “a sim”, “a hologram” are used casually in routine life in various language constructs. Such popularity is reinforced by the fact that humans are developing more and more complex simulations themselves, both for research and for entertainment [57, 30, 34, 75, 39].

The possibility of nested simulations has undergone serious consideration by numerous scientists in recent decades; this work will not discuss the validity of the arguments that concern the probability for existence of such a nested structure, such as the simulation argument [17, 13, 11, 68, 7, 5] or the doomsday argument [47, 62]. I will rather review some of the consequences of complex simulations being embedded recursively in other simulations and discuss their basic properties – such as descriptive and computational complexity – which can be evaluated in such a nested structure and potentially used to estimate or define the level of development of simulated universes.

Since this analysis will concern multiple levels of nesting and, as shown in the following sections, universes that exist on different levels can be entirely different and may share no common properties, it is inherently not possible to use specific terms that are known in human language and would perfectly describe properties of each such universe. Still, a short discussion and explanation of key words such as “universe”, “simulation”, or “civilization” is provided in the following paragraphs. In the future, it may be possible to define such terms precisely in the language of mathematics, along the lines of formalisms presented in Sect. 3.3.

Although traditionally the meaning of the words “world” and “universe” is different, the difference has no bearing on the analyses described here. This is because in general, there may be many civilizations in a single “multiverse”, in a single “universe”, or even in a single “world”, and a single civilization, or even a single being, can develop many independent simulations of any scale (including a “multiverse”). Although these words have a specific association in our reality, in the broad sense they can only express a relative order of scales: multiverse>universe>world. To describe scales more formally, precise and general quantities will be introduced in Sect. 3.1.

The word “simulation” primarily means reproducing some phenomenon with a model and this is an adequate description of a situation where a computer process approximately reproduces a natural process. But what if computation implements a process or an environment that has no existing counterpart? We could be tempted to say that it is also a simulation, because it is run *there* (in a computer, not directly in our world). However, such a “simulation” would be the one and only original, not a model of a true original. To illustrate how complicated this may become, note that such an original environment that is processed in computer memory could actually be (imperfectly) reconstructed later in reality. Then, the environment that exists in reality could actually be called a simulation of the genuinely original environment that exists in a computer [37]. Since in this piece the word “simulation” will concern universes that exist on various levels of embedding hierarchy, I will be using this word as meaning an intentionally (non-randomly) created, working (running) model of a process, so the meaning is not restricted to definitions specific to our particular reality, cf. [25, 16, 57, 30, 75, 27, 18, 5, 76]. As to the latter example of an original process running on a computer, the process should not be strictly called a simulation merely because of the fact that it takes place in a computer – instead, it should be simply called a creation, cf. [56, 17]. However, since "creation" is a very broad term, I will still use the word "simulation", but the meaning of this word will include originally created universes, not only models of other, already existing realities.

In the same vein, while providing examples I will refer to simulations running on a digital "computer" as this is currently the most common and well-known case. However, considerations that concern other universes and simulations of universes would obviously not be limited to such a machine, and would require some functionally analogous substrate that executes another process that represents a simulated universe. Note that even in our reality, a cellular automaton universe may be implemented in different ways, and the substrate underlying its cells may be very different [65, 66, 71].

Just as the other discussed terms, "civilization" in this work can be applied to different levels of the embedding hierarchy, and therefore it is not restricted to human civilization with its traditionally recognized properties such as complex society, symbolic communication, or urban development. This is why the level of "civilizational development", as it refers to the level of development of agents or beings that potentially exist in each created universe, is hard to determine; it could for example correspond to the ability to solve mathematical problems or prove abstract theorems. Just as with other high-level concepts that are not easily defined such as "consciousness" or "life", it is better to speak about specific and measurable properties that constitute a given concept. This work discusses whether it would be possible to estimate or define the level of development of embedded universes by using well-defined properties known to human science today.

Instead of using the words "multiverse", "world", or "environment", in this article I will most often, and more simply, be using "simulation" or "universe", and occasionally I will be using phrases like "a universe knows" or "a universe can create" to implicitly mean beings, agents or civilizations existing in that universe.

The following section summarizes how simulation is understood and employed in computer science today and how operating systems and software applications (such as various simulations) are embedded within each other. Sect. 3 formalizes the hierarchy of embedded universes, introduces a number of useful properties that characterize each universe, and provides a few real-life examples of embedded universes in the software realm. Sect. 4 discusses the relationship of the introduced properties with a level of civilizational development in a universe, exploring whether one can expect a manifestation of the level of development in any of these properties. Sect. 5 summarizes this paper.

2. Universes simulated and universes created by humans today

Before considering the hierarchy of universes and simulations embedded in each other, let us start with the cases that are known to our civilization and that are our contemporary reality.

One could argue that fantasizing and imagining potential scenarios is a form of simulation, where the subject intentionally "creates" more or less realistic universes and agents in order to observe the situation unfolding. This also happens unintentionally during dreaming. While philosophers have discussed for millennia whether it is possible to distinguish reality from a dream, the argument that imaginations can be considered a form of simulation may be supported by the fact that by "simulating" reality, dreams allow animals to (unconsciously) solve problems, find solutions and avoid potential threats in real life [12, 61, 72]. However, such imagined or dreamed "simulations" are short, fragmentary and probably unintentional [20].

As humans clearly have the ability to simulate complex behaviors of other people, both in reality and in imagination [25, 16, 19, 50], the status of purposeful, long-lasting imaginations of various scenarios that might happen in reality is more interesting. Such mental processes are, however, more like descriptions (models) of potential scenarios (and their potential outcomes) than working simulations themselves.

More tangible kinds of simulations occur in the domain of computer science. Simulation gradually emerged in technology and engineering out of necessity, and it is used intentionally, so it is easy to identify and understand complex mechanisms that are employed today in order to make it successful and efficient. Simulations are prevalently used to understand reality, and obviously they only have limited resources (smaller than those used by the pieces of reality that are simulated), so researchers and programmers have to cope with this limitation by developing various solutions and tricks. Some of these will be described shortly because they are useful for discussing the concept of embedded universes and they share many parallels with philosophical ideas regarding simulation and reality.

Where human-computer interfaces are employed, computer scientists do everything to make digital content feel real while using as few computational resources as possible. This concerns data compression (compressed audio and graphics sound and look good, even though as much of the less relevant information as possible is removed from the original). In computer-generated graphics, many versions of 3D objects are designed with different "level of detail" (LOD), and the appropriate (minimal) LOD is used depending on how far a 3D object is from the observer. The same trick is used for textures – higher-resolution textures are used only when the observer is close to the object. Lower and higher LODs and texture resolutions are switched unnoticeably for the user, just before the change becomes noticeable.

Only those objects are rendered that can be perceived by the user, so objects that one is not looking at are not rendered at all. This optimization can go even further: if this does not impair the user's experience and the logic of the simulated universe, parts of the universe that are not visible can be frozen in time or not processed at all. Ideally optimized simulation would create parts of the simulated universe on demand (just in time) so that they come to existence once they can be sensed. What is not perceived does not come into existence, but the program is always ready to create required parts of the universe according to the rules of the simulation. What has been forgotten by users and does not influence other parts of the simulation can also be deleted from the simulation to free up valuable resources.

It is often joked that every computer program will ultimately consume all available resources, and this is quite true for simulations that can be made more and more accurate and detailed. In this context it is worth to appreciate the progress made in technology. Digital calculators and programmable computers started with just a few bytes of memory, and even at this fledgling stage, they proved useful to humans. From individual bytes, over 60 years we went through kilo, mega, giga, tera, and up to storage capacities of petabytes, which is 15 orders of magnitude. Since traditional computers use binary encoding, the industry was incorrectly using standard SI units (kilo, mega, etc.) to describe memory capacity, and as the deviation between decimal and binary-based powers keeps increasing as the numbers grow, standards and trade organizations were forced to introduce a new binary prefix to avoid ambiguity so that now kibi (Ki) means 1024, mebi (Mi) means $1024 \times 1024 = 1048576$, etc.

This exponential growth of memory capacity accompanied by a similar trend in computing power allowed first engineers and developers, and then everyday users to benefit from the embedding of computational environments in other computational environments. The hardware of one machine can be simulated by software running on another machine, creating a compatibility layer, so that applications from one machine can be run on another. A machine can also be *emulated* by another machine – this term is used to emphasize that the simulation goes deeper, not only reflecting the general abilities and properties of the emulated machine, but also its internal structure and specifics. Finally, a new term is gaining popularity – virtualization. This is the simulation of devices or operating systems on other devices or operating systems, where the simulated systems can be scaled, multiplied, and conveniently managed by a virtual machine monitor (a “hypervisor”). Today, even mobile operating systems and web browsers routinely serve as virtual machines or separate “sandboxes”, where applications and scripts can run securely without the risk of harming the main top-level system.

There are no real objections, except from limited computing resources, for this nesting to continue so that simulation of devices and operating systems are recursively embedded multiple times in one another. While we are now considering only digital, electronic devices, the characteristics and nature of the machines can potentially vary from level to level, involving mechanical, biological, chemical, quantum and other substrates. For practical reasons, humans prefer to use deterministic, reliable environments for computation. The end result of a computation (e.g., some simulation) may be the same independently of the substrate that is used by computing machines and independently of the number of intermediate levels of nesting one machine into another. This means that for the outcomes of the implementation of some algorithm (e.g., some simulation), the computing substrate (medium) and its level of embedding may be irrelevant ([37], cf. [22, 10]).

I have discussed above how some aspects of contemporary computer science are related to the concept of simulating embedded universes. There is one more interesting process used in computer technology, again for practical reasons, to increase the efficiency of computation when virtual machines are involved. Each level of virtualization incurs some computational cost (an overhead) because it constitutes an additional layer of simulation – in other words, it needs a different representation of a program. Programmers try to reduce this overhead by avoiding the level of interpretation of programs, bringing them closer to the simulating machine, hence the notion of *native* code. The original program code (the one that is to be simulated, or executed by a virtual machine) is *compiled* to *machine code*, i.e., the code that is natively understood by the underlying machine. This compilation often happens just-in-time, i.e., while the program is running, just before the execution of the individual pieces of code [42, 44]. Such technology is regularly used in most contemporary web browsers and in some programming languages and while it is neutral (undetectable) from the perspective of a running program, it causes the program to run orders of magnitude faster. This means that in order to increase performance, software is brought closer and closer to computer hardware, this hardware can adapt to the software to run it efficiently [35] – and the hardware itself is in turn brought closer and closer to the "hardware of our universe", up to the point where several atoms are used to store one bit of information and quantum effects are exploited. One could metaphorically say that boundaries dissolve as computer software becomes hardware, and hardware becomes the universe. When the boundary between the real (parent) and simulated (child) universes becomes so thin, the simulated universe may exist nearly side by side with the real universe, and the real universe may share some real resources (e.g. some dimensions of spacetime) with the simulation. If, apart from sharing resources, the simulation were created to directly reflect the real universe, it could be considered a ghost (shadow) universe to the real universe.

3. Embedding of universes

As demonstrated in the previous section, current technology not only allows, but routinely takes advantage of the embedding of operating systems and programming environments into one another. Today, the level of such nesting is usually 1, and rarely 2 or 3, and in most cases, computer programs that are embedded are implemented to solve simple problems and to perform predictable tasks. Let us now discuss the possibility of the simulation of universes being nested in other simulations.

The fundamental difference between the embedding scenarios described earlier and universes embedded in other universes is that in the former case, all these hardware and software systems were designed by intelligent entities occupying one universe. Even if these "virtual machines" were embedded multiple times one in another, they all shared the same direct creator. Where universes are to be nested in other universes, we expect them to emerge from within one another, and not to be "implanted" into descendant universes by an intelligent entity existing in an ancestral universe. The "implanting" scenario is possible, but such a case would not count as a level of embedding that emerged distinctly from a descendant universe.

In the following discussion, I will be using the notions of *parent* and *child* universes to describe their relationship. Note that this relationship may concern causation (which created which) or location (which is embedded in which). In typical scenarios, both relationships would be consistent, but in some special cases discussed later universes may (temporarily) communicate or share resources, so that creations of different ancestors may have a common location, or individual universes may have multiple creators. Unless stated otherwise, I will be referring to the causation (creation) relationship as it is more useful. While "parent" and "child" (analogously to super- and sub-) will be used to denote direct relationship (i.e., within one "generation" or one level of embedding), *ancestor* and *descendant* universes will be any universes in the ancestry or descent lines (i.e., with the difference of one or more levels of embedding).

There are a few potential misconceptions that seem to be intuitively correct, so I will address and discuss these first.

- *"Creators of a child (simulated) universe that exists in the parent universe always understand their own creation".* This may seem obvious at first but it can be easily proven false, even by providing examples from existing simulations developed by humans. First, simulations are often developed exactly in order to experience and comprehend selected aspects of some process that is not well understood. Second, software developers today have difficulties understanding the source code of programs written by their colleagues, even though the code has been carefully documented. In some cases, human capacity is also insufficient for understanding the results of running such well-documented programs. Considering that vast amounts of memory mentioned earlier can be filled with sophisticated constructs produced by some algorithm (for example, a simulated undirected evolutionary process), there are many cases where such constructs will be far beyond the human capacity of understanding. This is especially true considering that the simulated (child) universe may be partially or entirely different from our (parent) universe, with different laws – so that complex constructs, intelligence, or consciousness emerging in a descendant universe may even be unnoticeable to human creators, or may be considered uninteresting ([36, Sect. 3]; [77, 21]).

  It is not necessary to resort to comparisons of our universe to simulated universes; cultural, belief, or worldview differences are sufficient enough for people to make understanding other humans almost impossible. Let us just refer to other species from Earth: how can we say whether dolphins are intelligent? If so, how much and on what scale? Are cats conscious? Are sparrows emotional? Do plants feel pain? Are viruses alive? Without even referring to anything outside of our universe, these questions provide some insight on how difficult it would be for conscious beings in some universe to recognize, understand and evaluate other, possibly even descendant, universes [37].

- *"More energy or resources are required for more complex simulation".* While intuitively sound, recall considerations from the previous section on the efficiency of programs running in a virtual machine. It is not difficult to waste resources or to create many efficient, but uninteresting simulations, and there may be no pressure to optimize the universe when resources are abundant, so the amount of resources required by a simulation cannot be a measure of its complexity.

- *"A simple universe cannot create a more complex universe by itself".* This statement seems reasonable because a parent universe may appear to constitute some kind of a limit on the complexity of its child universe. However, this depends on where and how complexity is measured. A civilization existing in their universe may not know the minimal, shortest formula that describes that universe (this formula, program, or set of principles is known to their parent universe), so they may be unable to properly evaluate the complexity of their own universe (this will be discussed later in more detail). Instead, each civilization will resort to estimating the perceived, subjective complexity as seen from the inside of their universe. This estimate will vary depending on what is the currently perceived configuration of their universe.

To illustrate how difficult it may be to predict and place some limits on the complexity of descendant universes, consider two initially identical, simple species evolving for the same amount of time in two nearly identical environments. Due to diverse conditions or minor random factors, one species may stay simple and intact during the entire time, while the other may evolve into multiple species that become extremely complex beings. Similarly, the parent universe may seem simple and stagnant, while a child universe created and embedded within it could thrive and develop until, in the eyes of the parent civilization, it becomes more complex than the parent civilization – just like a simple processor can run a program that simulates a more complex processor. A possible scenario for our civilization would be such that we created a sophisticated simulation (a child universe) and somehow decided not to develop our own science further, but instead to look into what happens inside that child simulation and to help it develop until it exceeds our own level of development. This child simulation would serve as a "sandbox" and if anything goes wrong, it only happens within the child simulation, and this would not have an affect on our civilization. The child universe(s) could therefore drive the development of the parent civilization in a safe way.

This leads to an issue of whether running a simulation which is ultimately capable of producing sentient, possibly conscious beings would be ethical. Currently existing software simulators are more than capable of generating highly sophisticated agents if they are given enough computing power – in this regard we don't have to wait for the discovery of some new, still unknown algorithms, or some quantitatively different finding or breakthrough that could be stopped beforehand. Who or what is going to control and constrain the running of unethical experiments, given that the simulators are available to everyone and the only requirement is a sufficient amount of computing power – and this resource is general-purpose, increasing continuously and will be abundant? Given the scale of potential suffering that could take place in such simulations [17, 32, 2, 13], to avoid suffering one would need to limit the freedom of individuals in this regard.

Another ethical issue concerns stopping a simulation of conscious, intelligent beings – not necessarily erasing or killing the simulation, just stopping it so that no being suffers and no being is aware that the simulation has been stopped. This issue resembles the ethical status of freezing embryos created by means of in vitro fertilization. If the answer is "not ethical", would it be ethical to keep running the simulation at a lower, or near zero, speed? And does it make any difference at all given that simulation speed is imperceptible by the child simulation?

- *"Artificial Intelligence (AI) exceeds humans on simpler tasks, but the more sophisticated the task becomes, the less AI can surpass humans, and for the most difficult problems, AI cannot achieve human level"*. This is true for poor AIs, but when they become increasingly general and intelligent, the gap between the performance of humans and AIs will increase as problem complexity grows.

  One of the reasons AI can be so successful is its ability to build and evaluate a huge number of different variants of models of reality. Just like how AI currently learns how to play a game by playing with itself and evaluating huge numbers of potential scenarios (that will never be actually played by the opponent, but they are still "imagined" or simulated by AI), it is conceivable that the AI, when given a lot of resources and asked about which decisions should be made, in order to answer such a question, will create huge numbers of universes with differing parameter values to evaluate ("imagine" or simulate) future scenarios and predict future outcomes. This would resemble more or less fine-grained, but still imperfect imaginations (simulations) of future scenarios that humans make up in their minds to make decisions and choose actions in order to reach the outcomes they desire and to avoid unwanted consequences, cf. [32, 1, 76].

3.1. Hierarchy of embedded universes and their properties

The least complicated scenario is where each universe has a single creator from another, parent universe, and locational relationships are the same as creational relationships. Therefore parent–child (creator–creation) dependencies emerge and they form a hierarchy (a tree) just as shown in Fig. 1. Letters denote individual universes,

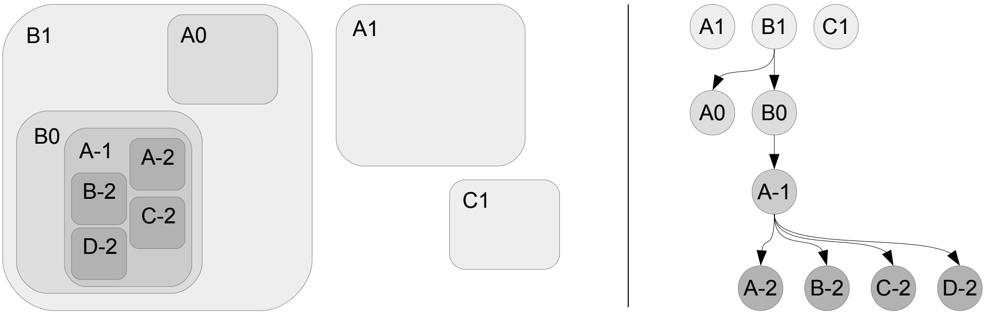


Figure 1: Two alternative, equivalent representations of sample parent–child (creator–creation) relationships between recursively embedded universes in the simplest, hierarchical configuration. Numbers indicate the level of embedding and are relative. Arrows point from parents (creators) to children (creations).

and numbers are used to indicate their level of nesting. These numbers are relative unless there is some reason to assign an absolute meaning to a given number. For example, a useful convention for a conscious being may be to assign the number zero to their universe, with negative numbers extending towards descendant universes they created, and positive numbers denoting ancestor universes.

This naming scheme is convenient for universes that are only aware of their descendants and the potential ancestry line. If a universe becomes aware of other descendants of its ancestors, or their "siblings", another naming scheme could be used to avoid ambiguities – for example, assigning a unique ID to each universe, or describing each universe by a sequence of symbols assigned to universes on a path leading from the top ("root") universe. In the latter naming scheme, A1 from Fig. 1 would become A, A0 would become BA, and C−2 would become BBAC – so the length of the name would reflect the level of nesting.

For each universe in this configuration, a number of useful properties (parameters, measures) can be possibly estimated, such as:

- Complexity. Many measures of complexity have been proposed [23, 31, 53, 55], of which the concepts of Kolmogorov complexity, minimum description length, minimum message length, entropy, and computational complexity seem to be the most useful (but not necessarily practical from the point of view of beings inside a universe) for estimating the complexity of a universe.

- Similarity to ancestral universes. In our world, some simulations are created to resemble reality as much as possible [34, 75], while others are abstract and focus on specific phenomena rather than on realism [26, 39].

- Interaction: influencing ancestral (or descendant) universes and being influenced by ancestral (or descendant) universes, cf. [59]. If created purposefully, descendant universes will likely influence ancestral universes (even if only by observation, inspection, or analysis), because this is why they are created. Influencing descendant universes by ancestral universes will likely be the choice of the ancestors.

- Resources used (absolute and relative to parent), such as processing power, the amount of stored information, or energy, cf. [6, 70, 3].

- Universal and general measures of properties of life, intelligence, and consciousness, when such measures become available ([70]; [36, Sect. 3]; [37, 77, 21, 8]).

It is not clear whether the concepts of dimensions, energy, causation etc., as we define them in our universe, exist in ancestral universes. We can easily get rid of dimensions or create new dimensions, let alone define any new set of rules in our descendant simulations, including principles that can hardly be imagined. Properties described above or their analogues may be applicable to our descendant universes if we choose to, but not necessarily to the ancestors of our universe. Properties that are considered fundamental in some universe may be neither inherited from its parent nor passed to its child universes.

3.2. Towards realism: special cases in embedding

The previous section demonstrated the most obvious, tree-like topology of the embedding structure; such topology has been considered and studied in the literature so far [10, 28, 32, 68, 47, 13, 62]. Now let us look at realistic, more complex cases that our civilization made possible with the development of computers:

- A universe can influence any of its ancestor and descendant universes, so the influence is not limited to parent–child universes. A universe can also "implant" a new universe in any of its descendant universes.
- Universes can be copied and transferred to (embedded into) other universes by ancestral universes. They can also be split and merged. Communication channels that allow universes to interact can be opened and closed.
- A universe can have many parent universes, including parents (creators) from different levels of the hierarchy, assuming that they can communicate and interact.
- Some contents of a descendant universe can be moved to its ancestors. An existing example is building a real robot that emerged from a simulated evolutionary process in a realistic, virtual environment, or implementing an AI (that was optimized in a virtual environment) in a real device that interacts with the real world.
- A universe can create an entire network of possibly interdependent and interconnected descendant universes at once.
- Universes do not have to have clear boundaries – connections may exist so that communication or interaction is more difficult or limited, but still possible. In this way, universes can use resources of ancestor or peer universes.

These cases cause the tree in Fig. 1 to become a graph with causal and locational relationships connecting potentially any pair of universes. Even though such scenarios seem complex, they are performed by computer scientists and hackers today, as outlined in Sect. 2. While programmers do not perform these operations on complex simulations of universes but on regular computer programs and their data, yet from the perspective of computer science, there is no difference between running a process or an operating system and simulating a complex environment with life-like properties. In computer science scenarios, for safety reasons, borders are often created between "universes" so that universes are protected and it is impossible to modify them from outside. One would not want applications on their smartphone to modify its operating system. This would pose a security threat and would be a "security hole" that could be exploited by malicious software, and crackers try to identify such vulnerabilities to break into virtual machines and programs. A similar scenario would apply for embedded universes.

To illustrate this complexity with one practical example of locational intricacy and volatility, let us imagine a software developer who wants to test a new simulation game

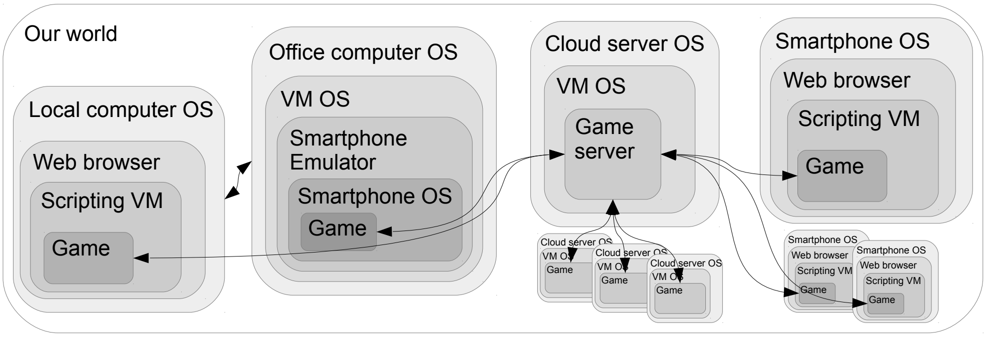
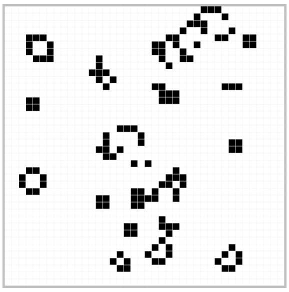


Figure 2: A partial, simplified visualization of locational dependencies in a hypothetical scenario of a development of a virtual reality game described in the text. OS is Operating System, VM is Virtual Machine, arrows indicate selected connections and interactions. This chart only shows programs related to a single instance of the game in each software environment; other (possibly identical) programs running simultaneously in each environment are not shown. For simplicity, advertisement software spanning across multiple devices and systems is also not shown. Cloud servers and smartphones may appear and disappear in time.

that he develops. He uses his computer that runs some operating system to remotely access another machine with another operating system in his office, and there he runs another operating system inside a virtual machine, and in that embedded operating system he runs an emulator of a smartphone. On this emulator, he launches his game in a special unrestricted mode that allows him to investigate all the details on how the game interacts with the smartphone, and to directly influence the game while it is running, bypassing standard security restrictions. The game connects via the internet to remote servers ("the cloud") where parts of the simulation of the virtual game universe are performed. The same servers are actually performing other, unrelated simulations, migrating these processes between physical machines and physical locations around Earth. Some parts of the simulation are distributed and run on the smartphones of people in all continents; some fragments of the simulated universe are stored ("cached") on the smartphones. The game displays advertisements, and this is managed by separate advertisement software that has limited access to many simulation games and gathers basic information about behaviors of agents (human and AI players) across different simulations and devices. The simulated universe occupies new machines as it grows, and it grows in response to the actions of players. This realistic scenario of today's software development is partially illustrated in Fig. 2, and even though it encompasses only one level of a causal parent–child relationship, its locational relationships are quite dynamic, sophisticated and hard to track.



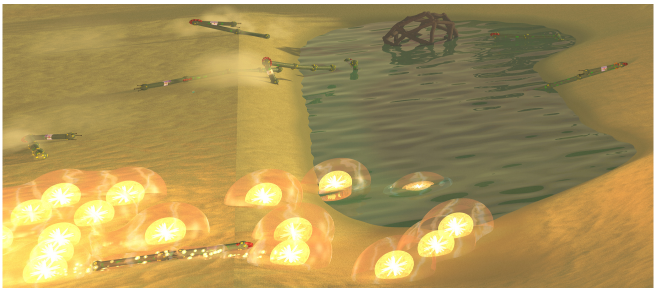


Figure 3: Left: a sample two-dimensional cellular automaton of size 40x40; cells are squares, each cell in this example can have one of two states (shown as white and black). Right: a simulation of three-dimensional creatures, Framsticks [40, 41], moving and interacting with virtual sources of energy.

3.3. Embedding in Artificial Life

In Sect. 3.2 above I discussed an example of a distributed multi-platform game. Now I will proceed with a more comprehensive example that includes two well-known models of universes primarily built in order to simulate selected aspects of life and its spontaneous development. Such models belong to the field of study called Artificial Life [4], where processes and phenomena of life are investigated using software, hardware and wetware (i.e., biochemical experiments).

The first model is a cellular automaton (CA) – a multi-dimensional grid of cells, each in one state from a set of available states [51]. In every simulation step, the state of each cell changes according to a given function (a rule) that depends on the state of the cell and the states of its neighboring cells. A CA is therefore a very simple model that can be easily implemented in software and in hardware; an example is shown in Fig. 3, left.

The second model is a simulation of three-dimensional creatures with "bodies" modeled as articulated solids and "brains" modeled as artificial neural networks of arbitrary topologies. These networks can process signals, acquire information about the states of the environment through sensors, and act through actuators, some of which allow the creatures to communicate (Fig. 3, right). The creatures are described by an artificial genotype (a sequence of symbols or "genes") that can be changed ("mutated" or "crossed over") – this way such creatures, given appropriate environments and selection pressures, can evolve.

One can describe both universes more formally. The state of the CA can be described by a set of pairs that contain indexes of cells and their states, i.e., a subset of the $CA_{Language}$:

$$CA_{Language} = \{(c, s) : c \in \{1, .., C\}, s \in \{1, .., S\}\}$$

where C is the number of cells and S is the number of available cell states. Given that the $UniverseCA(t)$ is the state of the CA in time moment t, the $transition_{CA}$

function will transform the current state of the CA into the next state:

$$UniverseCA(t+1) = transition_{CA}(UniverseCA(t))$$

Assuming that time t is discrete (hence the set of natural numbers $\mathbb{N}$ below), and $\mathcal{P}(X)$ denotes the power set of set X, i.e., the set of all subsets of X, the domains and codomains of the two functions are:

$$\begin{aligned} UniverseCA &: \mathbb{N} \to \mathcal{P}(CA_{Language}) \\ transition_{CA} &: \mathcal{P}(CA_{Language}) \to \mathcal{P}(CA_{Language}) \end{aligned}$$

In the second universe, the one with three-dimensional creatures – assuming that they are simulated in computer memory that can store M bits and that we will be specifically interested in neural activations of all neurons of some creature x – one can describe the memory state as a subset of $Memory_{Language}$, and the neural state as a subset of $Neural_{Language}$:

$$\begin{aligned} Memory_{Language} &= \{(m,b) : m \in \{1,..,M\}, b \in \{0,1\}\} \\ Neural_{Language} &= \{(n,a) : n \in \{1,..,N\}, a \in \mathbb{R}\} \end{aligned}$$

Here the $Neural_{Language}$ describes outputs of simulated neurons, given that a creature has N neurons and activations a of these neurons can be represented as real values. Analogously to the cellular automaton, the $transition_{Memory}$ function that implements the simulation transforms the current state of the memory into the next state. For convenience let us introduce the $map_{MemToBrain}$ function that will select bits from the $UniverseMemory(t)$ that encode neural states $Brain_x(t)$ of some creature x, and translate (decode) these bits into floating-point numbers with a given precision:

$$\begin{aligned} UniverseMemory(t) &= transition_{Memory}(UniverseMemory(t-1)) \\ Brain_x(t) &= map_{MemToBrain}(UniverseMemory(t)) \end{aligned}$$

Analogously to the cellular automaton, in this example the functions have the following domains and codomains:

$$\begin{aligned} UniverseMemory &: \mathbb{N} \to \mathcal{P}(Memory_{Language}) \\ Brain_x &: \mathbb{N} \to \mathcal{P}(Neural_{Language}) \\ transition_{Memory} &: \mathcal{P}(Memory_{Language}) \to \mathcal{P}(Memory_{Language}) \\ map_{MemToBrain} &: \mathcal{P}(Memory_{Language}) \to \mathcal{P}(Neural_{Language}) \end{aligned}$$

The advantage of introducing these descriptive functions is threefold. Firstly, it demonstrates that in such artificial (synthetic) environments this can be done, contrary to our physical and biological world where providing such functions – as one of the main endeavors of science – seems to be theoretically possible, but is in practice quite difficult and complicated. Secondly, by allowing one to formally describe universes and their relationships, it facilitates mathematical manipulations and the application of formal methods of analysis. Thirdly, the high-level notions of “universe”, “simulation”, “imagination” and their relationship discussed earlier can now be described explicitly and unambiguously.

Having these formal descriptions available, let us consider a few more complex configurations of such universes. It is known that a cellular automaton (CA) can perform equivalently to a Turing machine [48, 60]. Therefore a CA can be used to simulate another CA through some $map_{CA1ToCA2}$ mapping function; this function could even preserve the $CA_{Language}$ if both automata are compatible. Such embedding of one CA into another can be recursive, which would correspond to the composition of the mapping functions $(map_{CA2ToCA3} \circ map_{CA1ToCA2} \circ \ldots)$. Moreover, a computationally universal CA can also be used to simulate the work of the program that simulates three-dimensional creatures through some sophisticated $map_{CAToMemory}$ function. Such examples show that embedding universes one in another can go together with the change of language (representation) that may be caused by the mapping functions.

Let us assume that the creatures that evolved in the simulated universe needed to come up with the concept of a CA, so they are able to imagine, using some circuitry in their neural networks, the states of a CA, and can simulate the behavior of some $transition_{CA}$ function. This is a good illustration of the difference between the “real thing” (i.e., the underlying CA that makes the world of the simulated creatures possible) and the imagination of the CA that occurs in the simulated brains of simulated creatures (i.e., the states of $Brain_x(t)$ that can be mapped into the states of some imagined CA using some $map_{BrainToCA}$ function).

The imagined CA can behave exactly like the underlying CA that makes the simulation possible, because the simulated brain can simulate the way the $transition_{CA}$ function works in the underlying CA. This does not mean however that the simulated brain has the sufficient capacity to process arbitrarily complex CA states. It would make sense to call the imagined CA and the underlying CA two different *instances* of cellular automata, but keep in mind that one of them is indirectly embedded in another – “indirectly” meaning that in this particular example, there are the intermediate representations of $Memory_{Language}$ and $Neural_{Language}$ due to the composed function $map_{BrainToCA} \circ map_{MemoryToBrain} \circ map_{CAToMemory}$. Both CAs are therefore not entirely separate – one runs in the other. It may be tempting to call these two CAs *separate instances*, because the first mapping function, $map_{CAToMemory}$, may operate on a very tiny part of the underlying CA – a part that could be considered negligibly small. Another temptation would be to call both CAs *independent instances* due to the lack of the direct influence between the two, reinforced by the fact that the involved mapping functions may transform the representation of information in a sophisticated way, and may lose some information. Ultimately it becomes a matter of the precise definition of the words “instance”, “separate”, and “independent”, but the true situation is best described by the mapping functions themselves.

Could many recursively embedded CAs be imagined by these simulated creatures? Yes, but more complex, multi-level imaginations would require more sophisticated neural networks. The embedding sequence could even be interlaced (A CA that simulates creatures that imagine a working CA that simulates creatures that. . .) – this would correspond to the repeated composition of functions $(map_{BrainToCA} \circ$

$map_{MemoryToBrain} \circ map_{CAToMemory} \circ \ldots)$. Again, the more complex the structure, the more effort, motivation or justification is required for it to emerge or be developed, and there is some computational cost (or some loss of efficiency) associated with each level of nesting, because each mapping function actually performs some computation. Referring this to boundaries between software, hardware and our universe becoming thinner as mentioned in Sect. 2, humans try to make the mapping functions between software and hardware as simple as possible to increase the efficiency of the transition functions and decrease overhead, and for the same reason, engineers try to bring hardware as close as possible to the physics of our universe.

Finally let us consider whether, from within such nested universes, it would be possible to tell the difference between different levels of nesting? If the underlying substrates do not matter (i.e., it is not possible to distinguish a CA implemented in software, in electronic hardware or in some mechanical system, because their states are described in the same language and look identical) and there are no detectable differences in the transition function itself, then it would not be possible. Asking such a question is however closely related to the notions of perception and consciousness, and these aspects of embedded universes have been discussed in a separate article [8].

4. Evaluating and comparing properties of universes

If some universe developed a hierarchy of descendant universes and was able to access and comprehend the hierarchy, it would not only be possible to estimate the properties of individual universes, but the hierarchy itself. Given that there was some purpose of such an experiment, for its creator it would be interesting to investigate the topology of such a structure and to see how deep the tree is, how many branches it has, and to identify similarities in the tree [69].

Considerations from earlier sections of this work demonstrated that in practice, the hierarchy of multiply embedded universes may get more complicated, with intersections, cycles, mutual influences, and changes in time. Locational relationships may dissolve, universes may be copied or moved from one to another, or may be reproduced with variation. Still, given the properties enumerated in Sect. 3.1, it is possible to characterize each universe using several values.

When considering the embedding of universes, it is often useful to restrict parent–child relationships to only purposeful, intentional, and independent (unassisted) creations. Such relationships are more stable as, unlike locational relationships, they cannot dissolve or change. This restriction also excludes the spontaneous creation of universes, an example of which would be cosmological natural selection in the form of self-reproducing black holes [67] – although there are also theories that include intelligence and intentional actions in such replication of universes [14].

In such a structure of nested universes, it would be useful to deduce universal properties or rules that those universes always follow as a result of given causal parent–child relationships. This would allow one to put some bounds on selected properties of universes instead of merely saying that each universe can be different. One such restriction would be monotonicity: are there properties that are consistently increasing or decreasing along descendant lines? Such properties might then correlate with or define the level of civilizational development. One obvious example is energy, but even this needs to be carefully considered, as it is – like most properties – subjective to each universe. Imagine that humans create a simulation which includes intelligent agents. Do they have any concept at all of energy in our universe? Only if we wanted to mimic our universe in the simulation. Can we create any amounts of energy in their universe and come up with any rules of their physics, including manipulating energetic balance at our will? Obviously yes.

However, from the point of view of the parent universe, a measure of resources like parent energy (in our universe we would call it real or true energy) cannot increase when a universe is embedded in another universe, unless the parent universe is unaware of all ways of exploiting such a resource. The absolute amount of such a fundamental resource used by a universe would therefore always decrease along the descendant path. Energy production and consumption as a measure of the technological advancement of a civilization is the idea of Kardashev [33] who proposed a scale with three categories (types) of civilizations, but this particular classification is specific to our universe as it relies on the concepts of planets, stars and galaxies.

As discussed earlier, complexity also suffers from the problem of subjective assessment within each universe. It is parent universes that know the true minimal (algorithmic) description – a set of rules – of child universes. Again, imagine a simulation created by us, human engineers, where seemingly random processes occur in the simulation. We know we used a simple, pseudorandom but deterministic generator defined by just one line of program code, yet its behavior is very difficult to describe algorithmically for intelligent beings inside the simulation if they are unable to obtain sufficient information about these processes. Us making the line of code slightly longer and more complicated would render its logic practically impossible to discover by simulated beings if they can only acquire incomplete, limited information. Analogously, what we as humans perceive as chaos or true randomness (i.e., inherent indeterminism), may be a deterministic large-scale simulation that follows simple rules unknown to us.

Just as with energy, the descriptive Kolmogorov complexity of children cannot be higher than parent complexity (in our universe we would call it true complexity), because the description of a parent must include all its actions including its creations and, consequently, their actions. It is just that most probably the parent does not know its own minimal description – this description is only known to the parent's creator, which is the parent's parent. Therefore, when comparing descriptive complexities of descendant universes, one should do so from the perspective of the parent universe and the knowledge it has about its creations.

This particular definition of complexity is not, however, sufficient to characterize the general complexity of a universe, as it does not consider the resources that are required to produce and process a universe based on its description. Based on the Kolmogorov complexity alone, one could say: "Design a proper yet relatively simple set of rules that generate a stable, open-ended universe that includes some amount of variation and facilitates self-organization, provide a *sufficient* amount of resources (including time), and you will get everything you need, including a hierarchy of descendant universes, consciousness, and superintelligence". This way, the minimal description of superintelligence and consciousness would be reduced to the initial, relatively simple set of rules – hence the need to consider other aspects of complexity that measure the amount of resources required by algorithmic descriptions to run. These computational complexities are also monotonically decreasing along descendant lines, but each universe can only learn about the complexity of its descendants (and probably its own complexity) and express them in terms specific to their own universe.

In discussions on nested simulations, such a configuration of simulations is sometimes called recursive. In mathematics and computer science, "recursion" has a well-defined meaning – it occurs when some function, procedure or rule is applied within its own definition. When such a function is computed, this results in a (possibly, but not necessarily infinite) sequence of evaluations of this very function. In this regard, nested universes may, but do not have to be based on a strictly recursive description – as it was discussed earlier, parent universes do indirectly describe their child universes, but child universes do not have to follow the same part of the description that created their parent universe. In other words, multiply nested universes do not have to be self-similar (e.g., fractal-like) structures along paths in the hierarchy, and child universes do not have to inherit much logic (nor even much description) from their parent universe. Instead, a reasonable scenario is that the description of a parent universe enables the emergence of different descriptions for its child universes. These descriptions in turn may or may not enable the emergence of children's child universes, and so forth. What *can* be considered recursive are some properties and phenomena of embedded universes such as the emergence of intelligence, complexity, consciousness or the ability to create child universes.

In the case of universe properties whose absolute and objective values are hard to tell and can increase or decrease for universes in descendant paths, one could assign to each universe the extreme value of all of its descendant universes. For example, the maximum level of intelligence (whatever definition will be applied to evaluate this property) that emerged in descendant universes could serve as an estimate of the potential of a universe to create intelligent universes within itself. A similar procedure of calculating the extreme (minimum or maximum) value of all of the descendant universes can be used analogously for other non-monotonic properties. Other statistics such as summation that can characterize a particular type of capacity of a universe to produce descendant universes can also provide useful estimates.

A simple and appealing property of a universe that could reflect its level of development is the maximal depth of nested descendants – the length of the longest descendant path. Assuming that the considered universe has level 0 and its descendant universes have negative indexes as in Fig. 1, this property would be the negated minimal index of all descendant universes. For our civilization this value would be 1, because we have purposefully, intentionally, and – as one can assume, independently from potentially existing ancestor universes (cf. [68, 54]) – created a large number of diversified simulations, but so far none of these simulations did the same.

There is, though, an argument against using this property as an indicator of civilizational development: in order to create a child universe, beings in the parent universe most probably need to be intelligent and conscious, and they are probably aware of such concepts as unhappiness and suffering, or at least the possibility of such feelings. Given that a civilization that creates child universes – universes so complex that they themselves can create their own child universes – is highly developed and aware of suffering, an ethical civilization would probably not want to create such child universes, or would ban the creation of descendant simulations in which conscious beings could suffer [17, 32, 2, 13].

However, the assumptions of this argument do not have to be universally valid (consider the earlier example of AI that creates child universes just to evaluate potential future scenarios) and it is not clear whether such actions would be considered unethical in a universe that could routinely create huge numbers of such simulations for utilitarian purposes. There are also other possibilities that would void this argument – for example, the parent civilization may be unaware of what happens in its descendant universes, may be unaware of the existence of a child universe that emerges spontaneously, unofficially, or is created by unethical or non-ethical beings, may follow another definition of ethical behavior, may consider descendant civilizations so primitive that they are not worth caring, may consider the suffering period very short and negligible until descendant civilizations develop, suffering may be considered just a kind of a negative stimulus, or suffering to some degree may be considered acceptable. Still, given that there is no universal pressure for each (especially ethical) civilization to develop child universes that contain intelligent, sentient, and conscious beings, the maximal depth of embedded descendants – while being an important and informative property – cannot be uncritically used to define or measure civilizational development.

5. Summary

In Sect. 2 I described how simulation is understood and employed in computer science today, how operating systems and software applications (such as various simulations) are embedded within each other, and how software, hardware and the physical universe unify as technology advances. Sect. 3 pointed out the difference between multiply embedded software applications that can be encountered today and multiply embedded universes that should emerge spontaneously to constitute a natural, multi-level embedding. I distinguished two distinct kinds of parent–child relationship: causal and locational, I formalized the hierarchy of embedded universes and introduced a number of useful properties that characterize each universe, such as descriptive and computational complexity, similarity to ancestral universes, the degree of interaction, the amount of used resources and the universal measures of life, intelligence and consciousness. Examples of realistic, more complex cases in embedding were enumerated to demonstrate that the traditional tree-like topology cannot represent many practical scenarios, and real-life examples such as a distributed multi-platform game, a cellular automaton and a simulation of creatures were provided to illustrate the complexity of parent–child relationships and the embedding hierarchy in the contemporary software realm.

Sect. 4 discussed the relationship of the introduced properties with a level of civilizational development in a universe, exploring whether one can expect a manifestation of the level of development in any of these properties. These considerations indicate that the meaning and interpretation of even fundamental properties and quantities of each universe, such as energy or complexity, can vary wildly depending on which universe is evaluating them [7, cf. Sect. 4.1]. Each universe has its own truth that is considered objective in this universe, while descendant universes are considered simulations that do not have access to all the information. Universes at higher levels are therefore allowed to know "more truth" or "more-real truth". On the other hand, from the point of view of each universe, what matters most is their truth; the mechanics and logic of their ancestors, albeit interesting (as revealing more-real or more-objective truth), might not influence their own reality and existence at all. Additionally, the ancestors do not necessarily understand their spontaneously emerging descendants (despite theoretical ability) and even less so for greater distances in the hierarchy.

The three interesting properties that could still be used to characterize the level of development of universes are their descriptive complexity (as the minimal description length provided at a sufficiently high level of the embedding hierarchy), the amount of resources used (informational and computational complexity), and the maximal depth of nested descendants, restricted to intentional, unassisted creations. The interpretation of the latter, however, is limited by the fact that there may be no universal tendency to create child universes capable of intentionally and independently creating their own descendants. One should also bear in mind that such discussion is constrained by our lack of experience as our civilization is only of embedding level 1, not (yet) by choice, but by the existence of technological limitations.

References

[1] Barrow J. D. Living in a simulated universe. In Carr B., editor, *Universe of Multiverse?*, pages 481–486. 2007.

[2] Basl J. The ethics of creating artificial consciousness. *APA Newsletter on Philosophy and Computers*, 13(1):25–30, 2013.

[3] Beane S. R., Davoudi Z., and Savage M. J. Constraints on the universe as a numerical simulation. *The European Physical Journal A*, 50(9):1–9, 2014.

[4] Bedau M. A. Artificial life: organization, adaptation and complexity from the bottom up. *Trends in cognitive sciences*, 7(11):505–512, 2003.

[5] Beisbart C. Are we sims? how computer simulations represent and what this means for the simulation argument. *The Monist*, 97(3):399, 2014.

[6] Bennett C. H. The thermodynamics of computation – a review. *International Journal of Theoretical Physics*, 21(12):905–940, 1982.

[7] Birch J. On the 'simulation argument' and selective scepticism. *Erkenntnis*, 78(1):95–107, 2013.

[8] Błądek I., Komosinski M., and Miazga K. Mappism: formalizing classical and artificial life views on mind and consciousness. *Foundations of Computing and Decision Sciences*, 43(4), 2018, submitted.

[9] Bohm D. Quantum theory as an indication of a new order in physics. B. Implicate and explicate order in physical law. *Foundations of Physics*, 3(2):139–168, 1973.

[10] Bostrom N. Are we living in a computer simulation? *The Philosophical Quarterly*, 53(211):243–255, 2003.

[11] Bostrom N. and Kulczycki M. A patch for the simulation argument. *Analysis*, 71(1):54–61, 2011.

[12] Brereton D. P. Dreaming, adaptation, and consciousness: The social mapping hypothesis. *Ethos*, pages 379–409, 2000.

[13] Ćirković M. M. Linking simulation argument to the ai risk. *Futures*, 72:27–31, 2015.

[14] Crane L. Possible implications of the quantum theory of gravity: An introduction to the meduso-anthropic principle. *Foundations of Science*, 15(4):369–373, 2010.

[15] Cronenberg D. eXistenZ. Movie, 1999.

[16] Currie G. Visual imagery as the simulation of vision. *Mind & Language*, 10(1–2):25–44, 1995.

[17] Dainton B. Innocence lost. Simulation scenarios: Prospects and consequences, 2002. `http://www.simulation-argument.com/dainton.pdf`.

[18] DeAngelis D. L. and Grimm V. Individual-based models in ecology after four decades. *F1000Prime Reports*, 6(39):6, 2014.

[19] Decety J. and Grezes J. The power of simulation: imagining one's own and other's behavior. *Brain research*, 1079(1):4–14, 2006.

[20] Domhoff G. W. The case against the problem-solving theory of dreaming, 2003.

[21] Dowe D. L. and Hernández-Orallo J. How universal can an intelligence test be? *Adaptive Behavior*, 22(1):51–69, 2014.

[22] Emmeche C. Life as an abstract phenomenon: Is artificial life possible? In Varela F. and Bourgine P., editors, *Toward a Practice of Autonomous Systems. Proceedings of the First European Conference on Artificial Life*, pages 466–474. MIT Press, 1992.

[23] Gell-Mann M. and Lloyd S. Information measures, effective complexity, and total information. *Complexity*, 2(1):44–52, 1996.

[24] Gershenson C. The world as evolving information. In *Unifying Themes in Complex Systems VII*, pages 100–115. Springer, 2012.

[25] Gordon R. M. The simulation theory: Objections and misconceptions. *Mind & Language*, 7(1–2):11–34, 1992.

[26] Griffeath D. *New constructions in cellular automata.* Oxford University Press, 2003.

[27] Haisch B. Is the universe a vast, consciousness-created virtual reality simulation? *Cosmos and History: The Journal of Natural and Social Philosophy*, 10(1):48–60, 2014.

[28] Hanson R. How to live in a simulation. *Journal of Evolution and Technology*, 7(1), 2001.

[29] Harrison E. R. The natural selection of universes containing intelligent life. *Quarterly Journal of the Royal Astronomical Society*, 36:193, 1995.

[30] Heudin J. *Virtual Worlds: Synthetic Universes, Digital Life, and Complexity.* New England Complex Systems Institute Series on Complexity. Westview Press, 2004.

[31] Hornby G. S. Measuring complexity by measuring structure and organization. In *IEEE Congress on Evolutionary Computation*, pages 2017–2024. IEEE, 2007.

[32] Jenkins P. Historical simulations – motivational, ethical and legal issues. *Journal of Futures Studies*, 11(1):23–42, 2006.

[33] Kardashev N. Transmission of information by extraterrestrial civilizations. *Soviet Astronomy*, 8:217, 1964.

[34] Kneebone R. L. Crossing the line: Simulation and boundary areas. *Simulation in Healthcare*, 1(3):160–163, 2006.

[35] Koeplinger D., Delimitrou C., Prabhakar R., Kozyrakis C., Zhang Y., and Olukotun K. Automatic generation of efficient accelerators for reconfigurable hardware. In *Proceedings of the 43rd International Symposium on Computer Architecture*, pages 115–127. IEEE Press, 2016.

[36] Komosinski M. Sztuczne życie. Algorytmy inspirowane biologicznie (Artificial life. Biologically-inspired algorithms). *Nauka*, 2008(4):7–21, 2008.

[37] Komosinski M. Życie w komputerze: symulacja czy rzeczywistość? (Life in a computer: simulation or reality?). *Nauka*, 2011(2):83–93, 2011.

[38] Komosinski M. Nesting, 2016. `http://www.framsticks.com/nesting`.

[39] Komosinski M. and Adamatzky A., editors. *Artificial Life Models in Software.* Springer, London, 2nd edition, 2009.

[40] Komosinski M. and Ulatowski S. Framsticks: Creating and understanding complexity of life. In Komosinski M. and Adamatzky A., editors, *Artificial Life Models in Software*, chapter 5, pages 107–148. Springer, London, 2nd edition, 2009.

[41] Komosinski M. and Ulatowski S. Framsticks web site, 2018. `http://www.framsticks.com`.

[42] Kotzmann T., Wimmer C., Mössenböck H., Rodriguez T., Russell K., and Cox D. Design of the Java HotSpot(TM) client compiler for Java 6. *ACM Transactions on Architecture and Code Optimization (TACO)*, 5(1), 2008.

[43] Kuhn R. L. Why this universe? toward a taxonomy of possible explanations. *Skeptic*, 13(2):28–40, 2007.

[44] Labschütz M., Bruckner S., Gröller M. E., Hadwiger M., and Rautek P. JiTTree: a just-in-time compiled sparse GPU volume data structure. *IEEE transactions on visualization and computer graphics*, 22(1):1025–1034, 2016.

[45] Lanza R. and Berman B. *Biocentrism: How life and consciousness are the keys to understanding the true nature of the universe.* Benbella Books, 2010.

[46] Lem S. Further Reminiscences of Ijon Tichy I. Professor Corcoran. In *Memoirs of a Space Traveler.* Northwestern University Press, 1982.

[47] Lewis P. J. The doomsday argument and the simulation argument. *Synthese*, 190(18):4009–4022, 2013.

[48] Lindgren K. and Nordahl M. G. Universal computation in simple one-dimensional cellular automata. *Complex Systems*, 4(3):299–318, 1990.

[49] Lisberger S. TRON. Movie, 1982.

[50] Lombardo M. V., Chakrabarti B., Bullmore E. T., Wheelwright S. J., Sadek S. A., Suckling J., Baron-Cohen S., Consortium M. A., et al. Shared neural circuits for mentalizing about the self and others. *Journal of cognitive neuroscience*, 22(7):1623–1635, 2010.

[51] Mainzer K. and Chua L. *The Universe as Automaton: From Simplicity and Symmetry to Complexity.* SpringerBriefs in Complexity. Springer-Verlag Berlin Heidelberg, 2012.

[52] McCabe G. Universe creation on a computer. *Studies In History and Philosophy of Science Part B: Studies In History and Philosophy of Modern Physics*, 36(4):591–625, 2005.

[53] Murtagh F. On ultrametric algorithmic information. *The Computer Journal*, 2007.

[54] Papakonstantinou Y. Created computed universe. *Communications of the ACM*, 58(6):36–38, 2015.

[55] Parrott L. Measuring ecological complexity. *Ecological Indicators*, 10(6):1069–1076, 2010.

[56] Pattee H. H. Simulations, realizations, and theories of life. In Langton C., editor, *Artificial Life, Santa Fe Institute studies in the sciences of complexity*, pages 63–78. Addison-Wesley, 1989.

[57] Peck S. L. Simulation as experiment: a philosophical reassessment for biological modeling. *Trends in Ecology & Evolution*, 19(10):530–534, 2004.

[58] Pribram K. *Brain and Perception: Holonomy and Structure in Figural Processing.* Distinguished Lecture Series. Lawrence Erlbaum Associates, 1991.

[59] Quick T. and Dautenhahn K. Making embodiment measurable. In *Proceedings of '4. Fachtagung der Gesellschaft für Kognitionswissenschaft' (KogWis'99)*, 1999.

[60] Rendell P. Turing universality of the game of life. In *Collision-based computing*, pages 513–539. Springer, 2002.

[61] Revonsuo A. The reinterpretation of dreams: An evolutionary hypothesis of the function of dreaming. *Behavioral and Brain Sciences*, 23(06):877–901, 2000.

[62] Richmond A. M. Why doomsday arguments are better than simulation arguments. *Ratio*, 2016.

[63] Roiland J. and Harmon D. Rick and Morty S01E04: M. Night Shaym-Aliens! Movie, 2014.

[64] Roiland J. and Harmon D. Rick and Morty S02E06: The Ricks Must Be Crazy. Movie, 2015.

[65] Roohi A. and Khademolhosseini H. Quantum-dot cellular automata: computing in nanoscale. *Reviews in Theoretical Science*, 2(1):46–76, 2014.

[66] Shirakawa T., Sato H., and Ishiguro S. Construction of living cellular automata using the physarum plasmodium. *International Journal of General Systems*, 44(3):292–304, 2015.

[67] Smolin L. Did the universe evolve? *Classical and Quantum Gravity*, 9(1):173–191, 1992.

[68] Steinhart E. Theological implications of the simulation argument. *Ars Disputandi*, 10(1):23–37, 2010.

[69] Tan H., Dillon T. S., Hadzic F., Chang E., and Feng L. *IMB3-Miner: Mining Induced/Embedded Subtrees by Constraining the Level of Embedding*, pages 450–461. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[70] Tipler F. J. *The physics of immortality: Modern cosmology, God, and the resurrection of the dead.* Macmillan, 1995.

[71] Tsompanas M.-A. I., Adamatzky A., Sirakoulis G. C., Greenman J., and Ieropoulos I. Towards implementation of cellular automata in microbial fuel cells. *PloS one*, 12(5):e0177528, 2017.

[72] Valli K., Revonsuo A., Pälkäs O., Ismail K. H., Ali K. J., and Punamäki R.-L. The threat simulation theory of the evolutionary function of dreaming: Evidence from dreams of traumatized children. *Consciousness and Cognition*, 14(1):188–218, 2005.

[73] Vidal C. The future of scientific simulations: from artificial life to artificial cosmogenesis. *arXiv preprint arXiv:0803.1087*, 2008.

[74] Wachowski L. and Wachowski A. The Matrix. Movie, 1999.

[75] Warburton S. Second Life in higher education: Assessing the potential for and the barriers to deploying virtual worlds in learning and teaching. *British Journal of Educational Technology*, 40(3):414–426, 2009.

[76] White J. Simulation, self-extinction, and philosophy in the service of human civilization. *AI & society*, 31(2):171–190, 2016.

[77] Yonck R. Toward a standard metric of machine intelligence. *World Future Review*, 4(2):61–70, 2012.

Received 15.07.2018, Accepted 14.09.2018