Gödel Agents in a Scalable Synchronous Agent Framework

Jörg Zimmermann*, Henning H. Henze, and Armin B. Cremers*

* Institute of Computer Science,
University of Bonn, Germany
{jz, abc}@iai.uni-bonn.de

Abstract. A synchronous framework for the interaction of an agent and an environment based on Moore machines is introduced. Within this framework, the notion of a Gödel agent is defined relative to a family of agents and environments and a time horizon T. A Gödel agent is the most flexible, adapting and self-improving agent with regard to the given environment family. It scores well across many environments, and not only in a selected few. Ideas from infinite game theory and ruin theory are used to get well-defined limits for $T \to \infty$ by introducing negative goals or repellors. This allows to score actions of the agent by how probable an action makes the survival of the agent till the end of time. Score functions of this type will be called "liveness" scores, and they provide a solution to the horizon problem from a foundational point of view. Additionally, by varying the agent and environment families, one gets a scalable and flexible testbed which could prove to be well-suited for analyzing phenomena of adaptation and self-improvement, both theoretically and empirically.

1 A Scalable Synchronous Agent Framework

Theoretical investigations have to be conducted within a conceptual framework. The process of taking a notion of colloquial language and turn it into a formal, precisely defined one often is not a straight path from the colloquial notion to the formal one, but a long and intertwined development resulting in several precise, but different versions of the colloquial term. These differences often are very subtle, but can have profound implications for the results obtainable within the respective frameworks. This conceptual dynamics also holds for the notion of an agent, which plays a central role in computer science, but especially in artificial intelligence. The agent concept underlying much of the research in foundations of artificial intelligence is, for example, defined by M. Hutter in [5], p. 126. It consists of two interacting Turing machines, one representing the agent and one representing the environment. If the environment produces an output, it is written on the percept tape of the agent. Then the agent starts its computations, deliberating the new percept, and finally produces an action as output, which is written on the action tape of the environment. While one machine is computing its next output, the other one is effectively suspended.

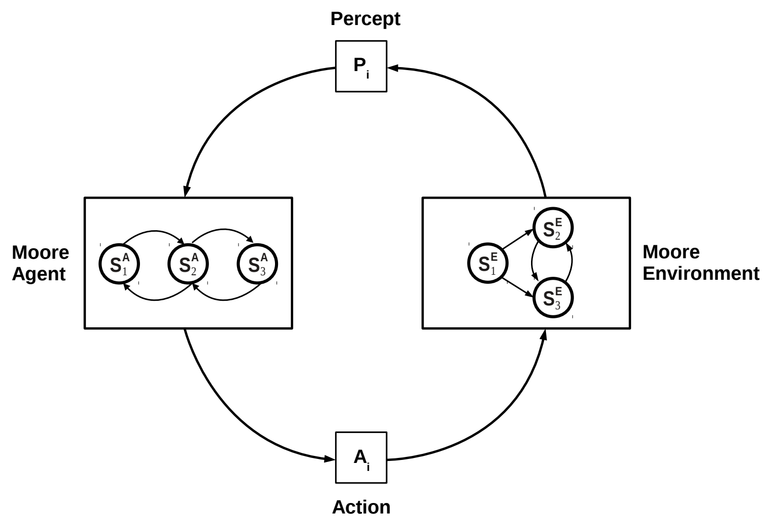

Fig. 1. Two Moore machines, agent and environment, interacting in a synchronous manner. The output of the environment at time i is the input of the agent at time $i+1$ and vice versa.

We call this agent framework *locally synchronous*, because the time structures of the agent and the environment are independent, as if they would exist in different universes, but are locally interconnected via percepts and actions. For a detailed discussion of local synchrony, which lies between global synchrony and asynchrony, see [3]. The locally synchronous framework was also used by R. Solomonoff in his seminal articles on universal induction [12, 13]. Full Solomonoff induction is incomputable, but in [14] it is outlined how *effective and universal induction* is possible when the agent and the environment are embedded into a synchronous time structure. This is one example for surprising implications resulting from seemingly small changes to a conceptual framework, stressing the point that some results are not as absolute as they might appear, but depend crucially on the details of the chosen framework.

Here we want to modify the locally synchronous framework in two ways, calling the new framework *globally synchronous* or just synchronous: First we replace Turing machines by Moore machines (see below), and second we do not assume the agent or environment are suspended while the other one is computing, but the Moore agent and the Moore environment are interacting in a simultaneous fashion: their transitions are synchronized, the output of one machine is the input of the other, and the output is generated and read in each cycle (see figure 1). A Moore agent can conduct complex calculations using its internal states and multiple cycles, but during these calculations the last output (or whatever the Moore agent produces as preliminary output while the complex calculation is running) of the Moore agent is used as input for the environment. Thus the Moore agent has to act in real-time, but on the other hand the environment is scanned in real-time, too, excluding the possibility that the environment takes more and more time to generate the next percept. In fact, in the locally synchronous framework, the agent does not know whether the current percept was generated within a second or one billion years.

Moore machines are finite state machines which read in an input symbol and generate an output symbol in each cycle. They do not terminate, but translate a stream of input symbols into a stream of output symbols, accordingly they are also called finite state transducers. Moore machines are named after E. F. Moore, who introduced the concept in 1956 [8].

A Moore machine is a 6-tuple $(S, S_0, \Sigma, \Lambda, T, G)$ where:

- S is a finite set of states,
- $S_0 \in S$ is a start state,
- Σ is a finite set called the input alphabet,
- Λ is a finite set called the output alphabet,
- $T : S \times \Sigma \rightarrow S$ is a transition function mapping a state and an input symbol to the next state,
- $G : S \rightarrow \Lambda$ is an output function mapping each state to an output symbol.

We replace Turing machines by Moore machines in order to have a better control of the complexity of the agents and environments, where the number of states of a Moore machine provide a natural complexity measure. This enables us to investigate notions of learning, adapting, and self-improving in scaled-down versions of the full Turing model, simplifying theoretical and empirical analysis. Furthermore, the synchronous agent framework is not only closer to real world applications than the locally synchronous one (the world doesn't stop while we are thinking, unfortunately), but also allows the distinction between adaptability and self-improvement within the framework. This distinction follows when one basically defines adaptability as *finding better actions for the same situation*, driven by past observations (which can be modeled in the locally synchronous framework), and self-improvement as *finding the same action, but quicker*, driven by internal self-modification (which doesn't count in the locally synchronous framework, but in the synchronous one).

2 Gödel Agents

An *arena* is a triple $(\mathcal{A}, \mathcal{E}, S)$, where $\mathcal{A}$ is a family of agents, $\mathcal{E}$ is a family of environments, and $S : \mathcal{A} \times \mathcal{E} \rightarrow \mathbf{R}$ is a score function assigning every pair of agent A and environment E a real number measuring the performance of agent A in environment E. First we assume that the agent and the environment families are finite, the cases where agent or environment families or both become infinite is discussed in section 4. In the finite case, the following notions are well-defined:

Definition 1. *For all environments $E \in \mathcal{E}$ the score $S^{pre}(E) = \max_{A \in \mathcal{A}} S(A, E)$ is called the* pre-established score *of E. An agent $A \in \mathcal{A}$ is called an* pre-established agent *of E if $S(A, E) = S^{pre}(E)$.*

So pre-established agents a priori fit best into a given environment, so there is no need for adaptation or self-improvement, but in general a pre-established agent for environment E_1 will fail miserably for environment E_2. The term "pre-established" is borrowed from Leibnizian philosophy. Gottfried W. Leibniz introduced the concept of "pre-established harmony" to describe that there is no need for "substances" (especially mind and body) to interact or adapt because God has them programmed in advance to "harmonize" with each other ([4], p. 197).

Definition 2. *The* loss *of agent A wrt. environment E is defined as: $L(A, E) = S^{pre}(E) - S(A, E)$.*

Definition 3. *A* Gödel agent *wrt. agent family $\mathcal{A}$ and environment family $\mathcal{E}$ is defined as an agent minimizing the maximal loss, i.e., as an element of the set $G(\mathcal{A}, \mathcal{E}, S) = argmin_{A \in \mathcal{A}} \max_{E \in \mathcal{E}} L(A, E)$.*

Definition 4. *The maximal loss of an agent is called its* global loss*. The global loss of a Gödel agent is called* Gödel loss*.*

In our case of finite agent and environment families, G contains at least one element, i.e., one or more Gödel agents exist. In section 4 we will see that this is also the case when the environment family becomes infinite but the agent family stays finite.

A Gödel agent can be seen as an agent which is most flexible, adapting and self-improving with regard to the given environment family. It scores well across the whole set of environments, and not only in a selected few. Thus a Gödel agent can be regarded as intelligent (at least wrt. the given environment family) in the sense introduced by S. Legg and M. Hutter in [7], where intelligence is defined as "the ability to achieve goals in a wide range of environments". Additionally, a Gödel agent operates within the real-time restrictions of the synchronous agent framework. If the environment family is diverse and complex, a Gödel agent has to be extremely adaptive and, driven by real-time pressure, self-improving. In this regard, Gödel agents are closely related to Gödel machines, which were introduced by J. Schmidhuber [11], and which represent self-improving and, in a certain sense, optimally efficient problem solvers. While Schmidhuber describes in detail the internal structure of Gödel machines, we try to characterize Gödel agents by their externally observable behavior. To elucidate the exact relationship between Gödel agents and Gödel machines is the topic of ongoing investigations.

If the loss is interpreted as a distance measure between an agent and an environment, then a Gödel agent would be located at the place which minimizes the maximal distance. In this sense Gödel agents are located in the center of the environment family.

3 Infinite Games, Ruin Theory, and the Horizon Problem

One goal of foundational investigations is to reduce contingent aspects like arbitrary parameters, often called "magic numbers", or reasonable but not necessary design decisions. One such parameter is the "horizon", a finite lifespan or maximal planning interval often necessary to define for an agent in order to get well-defined reward-values for agent policies. But especially in open environments existing for an indefinite timespan, this is an ad hoc parameter containing contingent aspects which may prevent the agent from optimal behavior. To stress this point, we quote M. Hutter ([5], p. 18):

"The only significant arbitrariness in the AIXI model lies in the choice of the lifespan m."

where AIXI is a learning agent aiming to be as general as possible.

In order to eliminate this parameter and to tackle the horizon problem from a foundational point of view, we will look into the notion of an infinite game, and, in a probabilistic context, into ruin theory.

An infinite game is a game which potentially has no end, but could go on forever. And for at least one of the players this is exactly the goal: to stay in the game till the end of time. A good illustration of this abstract concept is the Angel and Devils Game, introduced by J. H. Conway in 1982 [2]. The game is played by two players called the angel and the devil. The playground is $\mathbf{Z} \times \mathbf{Z}$, an infinite 2D lattice. The angel gets assigned a power k (a natural number 1 or higher), which is fixed before the game starts. At the beginning, the angel is located at the origin. On each turn, the angel has to jump to an empty square which has at most a distance of k units from the angel's current square in the infinity norm. The devil, on its turn, can delete any single square not containing the angel. The angel can jump over deleted squares, but cannot move to them. The devil wins if the angel cannot move anymore. The angel wins by moving, i.e., surviving, indefinitely. In [2] it was proved that an angel of power 1 can always be trapped by the devil, but it took 25 years to show that an angel of power 2 has a winning strategy [6], i.e., an angel of power 2 using the right strategy can survive forever.

This game nicely illustrates that the angel has not a definite or finite goal it wants to reach, but aspires to *avoid* certain states of the world. This seemingly innocuous transition from a positive goal, an attractor, to a negative goal, a repellor, solves the horizon problem from a foundational point of view, avoiding the introduction of arbitrary parameters. Now actions do not have to be scored with regard to the positive goals they can reach within a certain time frame, but according to the probability they entail for avoiding the repellor states forever.

A classical probabilistic example to illustrate the concept of an infinite horizon is from ruin theory. Ruin theory was developed as a mathematical model for the problem an insurance company is typically facing: there is an incoming flow of premiums and an outgoing flow of claims [1]. Assuming that the flow of premiums is constant and the time and size of claims is exponentially distributed, the net capital position of an insurance company can be modeled as a biased

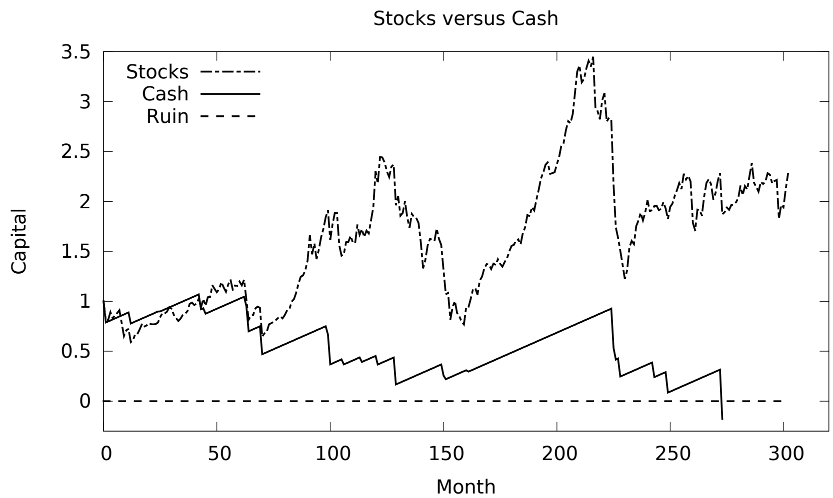

Fig. 2. In the insurance example, different actions (investing in either stocks or cash) lead to different capital position outcomes (survival or ruin) while getting the same premiums and the same claims occur in both scenarios. The safer cash investment scenario initially fares better but in the end ruin occurs (in month 273), while the riskier stock investment scenario is able to accumulate enough reserves over a longer horizon to survive. The simulation uses 302 monthly periods from 1990 to 2015, an initial capital of 1, a constant premium of 0.01 per month, exponentially distributed claim sizes ($\lambda = 5$) occurring with a probability of 0.10 per period and investment in either a stock performance index (DAX) or interest-free cash.

random walk. Ruin is defined as a negative net capital position. Now the maybe surprising fact is that there are parameter values for which ruin probability even for an infinite time horizon stays below 1, i.e., an indefinite survival has a positive probability. For the above model of exponentially distributed claims and interclaim times, there is an analytical formula for the ruin probability ψ with infinite time horizon [9]:

$$\psi(u) = \frac{\mu}{c\lambda} \exp((\frac{\mu}{c} - \lambda)u),$$

where $u > 0$ is the initial capital, $c > 0$ is the premium received continuously per unit time, interclaim times T_i are distributed according to $Exp(\mu), \mu > 0$ and the sizes of claims Y_i according to $Exp(\lambda), \lambda > 0$. For example, if the initial capital is $u = 1$, premium rate is $c = 0.2$, the expected interclaim time $E(T_i) = 2$ ($\mu = 0.5$), and the expected size of claims $E(Y_i) = 0.2$ ($\lambda = 5$), we get an infinite horizon ruin probability of $\psi = 0.04$, i.e., in this case the probability to stay in business forever, the *liveness*, is $1 - \psi = 96\%$.

In a more general example one can imagine that the insurance company can invest its capital in stocks. In figure 2, beginning from the same initial capital, two scenarios for the development of the net capital are shown: one conservative, where all the capital is kept as cash, and one aggressive, where all the capital is invested in stocks. In this case, the risky strategy prevails over the less risky one, but the best strategy is probably a smart mix of cash and stocks which is reallocated periodically, i.e., the investment strategy of the insurance company would decide on ruin or indefinite survival.

Both examples, the angel problem and the insurance problem, show how to avoid the horizon problem by switching the definition of goal from *reaching a world state* to *avoiding a world state*. In this sense, the accumulation of reward is only relevant as long as it helps to stay away from the repellor.

The above discussion of negative goals or repellors should serve as an illustration of a principled solution of the horizon problem and should inspire to search for new goal systems of agents. We do not claim that all agent policies should strive to avoid repellors. Negative goals should be seen as complementing, not replacing positive goals.

4 Gödel Agents in Infinite Arenas

Here we discuss the cases when one or both of the agent and environment families become infinite. This is especially relevant for theoretical investigations, because in most settings in machine learning or statistics the set of models assumed to generate the observations is infinite.

First we assume that only the environment family is infinite. The pre-established score is still well-defined wrt. each single environment, because the agent family is finite and thus the maximum in definition 1 exists. The same is true for the loss, so we arrive at an infinite number of losses wrt. one agent. Now this infinite number of losses may have no maximum, but they still have a supremum. In case of unbounded score functions, this supremum could be infinite, but for bounded score functions (which is the case for liveness scores, as a probability they lie in the $[0, 1]$-interval), we will get a finite number. So we can assign this finite supremum to the agents as the loss wrt. to the whole environment family, now calling this the global loss of the agent. This results in a finite family of agents each getting assigned a global loss number. Then there is at least one agent in this finite family having a minimal global loss, i.e., there is at least one Gödel agent.

The situation becomes more complicated if there are infinitely many agents, too. Then we can still define the infimum of all the global losses of all agents, but there does not have to be an agent assuming this infimum as its global loss. But in the case of bounded score functions, there is at least for every $\epsilon > 0$ an agent whose global loss exceeds the infimum less than ϵ, because in every neighborhood of the infimum has to be a global loss value assumed by an agent from the agent family. If we call such agents ϵ-Gödel agents, than we have just proved that for bounded score functions (and, as mentioned above, liveness scores are bounded), even when both agent and environment families are infinite, there are ϵ-Gödel agents for all $\epsilon > 0$.

An even more realistic agent framework should also address spatial aspects of Moore agents, like states per volume or access times for large storage devices, which become relevant when dealing with infinite agent families. A thorough analysis of the spatial aspects may imply the general existence of genuine Gödel agents even in the infinite family case, because minor decreases of global loss by using more states are offset by the associated costs caused by these additional states. This will be the topic of future research.

The above discussion addresses only the existence of Gödel agents in certain situations, not how to construct or approximate them. At least we now know that there is something worthwhile to search for.

5 A Scalable Testbed for Self-Improving Agents

As mentioned in section 1, one motivation for using Moore machines is their scalable complexity. In addition to the synchrony condition, this enables to investigate phenomena of adaptability and self-improvement in a wide range of different agent and environment families, providing a flexible and scalable testbed with regard to available agent resources and environment complexities.

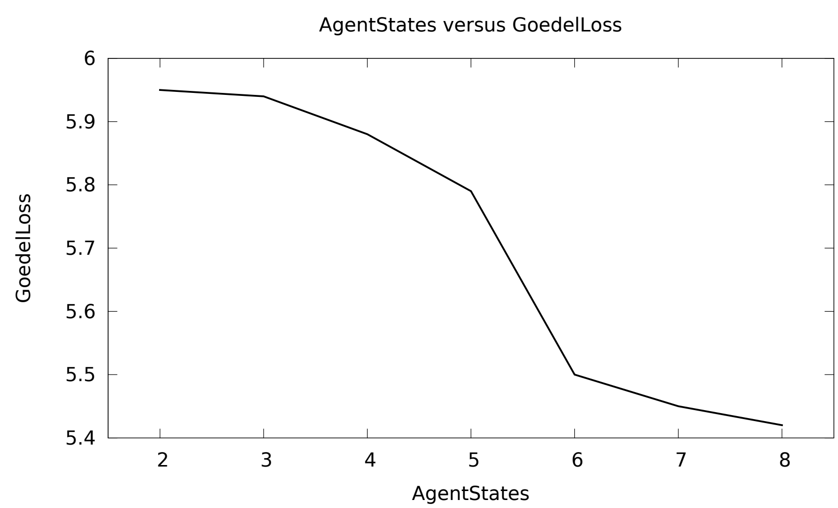

Fig. 3. Increasing agent complexity leads to lower Gödel losses as seen in these preliminary results from a simulation performed with 500 fixed Moore environments (having 5 states, 4 inputs and 6 outputs with random transition tables and random outputs), a fixed score function (using random scores in [0-10] depending on the environment state, the final score is given as average score per simulation step), and 100000 random Moore agents drawn per agent state number, all evaluated for 100 steps per environment-agent pair.

Our working hypothesis is that many aspects of adaptation and self-improvement occur already in scaled-down versions of the full Turing model. The detailed investigation of these questions, both theoretically and empirically, has just started and is the topic of ongoing research. Here we can present only a small, preliminary result, which nevertheless provides an indication of the fruitfulness and power of the proposed framework. Especially, it inspires to ask new questions which otherwise may have stayed unasked.

For example, we want to know how the Gödel loss varies if we increase the number of states in the agent family. Is there a "bang per state" effect and how large is it? In figure 3 the estimated Gödel losses for a fixed environment family, fixed score function, and increasing number of agent states are displayed. We can see a "bang per state" effect, but, like in many saturation phenomena, it finally gets smaller for every added state. Of course these phenomena have to be investigated much more extensive, both theoretically and empirically, but that is exactly what we hope for: that the proposed framework is the starting point for the detailed exploration of the landscape of arenas, adaptability, and self-improvement.

6 Discussion and Outlook

This is primarily a conceptual paper. A crucial part of theoretical investigations, aimed at solving real world questions, is to create a conceptual framework which is a good mix of abstracting away irrelevant or subrelevant details on the one side, but keeping enough structure so that vital aspects of the real world problem are still present on the other side. The scalable synchronous agent framework introduced in this article tries to offer such a good mix between structure and abstraction, hopefully leading to a fruitful testbed for theoretical and empirical investigations into phenomena of adapting and self-improving agents across a wide range of environments.

Especially the synchronization of agent and environment time allows to investigate phenomenons of self-improvement which is not possible in the locally synchronous framework. Our framework allows to explicitly, systematically and quantitativly analyze the trade-off between action quality and action time, which in other frameworks cannot even be formulated and hence has to be dealt with in an implicit and often ad hoc fashion.

This may lead to a new discovery process for agent policies by just looking at their performance with regard to the quality-time trade-off, without the need for a conceptual understanding of how they achieve this performance. From the outside, a Gödel agent for given agent and environment families is just a "bit mixer", and describing its inner workings by stating that it builds models or makes inferences is just a way to try to understand what is going on inside the agent, but is not necessary for its discovery or implementation.

By following this approach, we lose transparency, but we gain access to the whole agent space. In fact, concepts for designing the cognitive structure of an agent like logical inference, probabilistic inference, or utility, can be seen as specific *search biases* in exploring the agent space. But these biases are very focused, leading to the exploration of only some archipelagos, while leaving the great ocean of nonconventional cognitive architectures invisible and undiscovered. Of course, that does not mean that we aim for a bias-free, totally random discovery process for agent policies, but that the search biases should emerge as a result of a self-improving cognitive dynamics, rather than to be hardwired into the agent policy.

Generally, we advocate a change in perspective with regard to agent concepts from defining them via their inner structure (like this is done, for example, in [10]) to characterizing them from the outside using observable properties. This can be seen in analogy to the development in many mathematical areas, where first a "coordinate-dependent" description was introduced and then gradually replaced by a "coordinate-independent" one, often leading to general, elegant and powerful theories.

References

1. S. Asmussen. *Ruin Probabilities.* World Scientific, 2000.
2. E. R. Berlekamp, J. H. Conway, and R. K. Guy. *Winning Ways for your Mathematical Plays.* Academic Press, 1982.
3. A. B. Cremers and T. H. Hibbard. A programming notation for locally synchronized algorithms. In P. Bertolazzi and F. Luccio, editors, *VLSI: Algorithms and Architectures*, pages 341–376. Elsevier, 1985.
4. D. Garber. *Leibniz: Body, Substance, Monad.* Oxford University Press, 2009.
5. M. Hutter. *Universal Artificial Intelligence.* Springer, 2005.
6. O. Kloster. A solution to the angel problem. *Theoretical Computer Science*, 389(1-2):152–161, 2007.
7. S. Legg and M. Hutter. Universal intelligence: A definition of machine intelligence. *Minds & Machines*, 17(4):391–444, 2007.
8. E. F. Moore. Gedanken-experiments on sequential machines. *Automata Studies*, 34:129–153, 1956.
9. W. Rongming and L. Haifeng. On the ruin probability under a class of risk processes. *Astin Bulletin*, 32(1):81–90, 2002.
10. S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, third edition, 2009.
11. J. Schmidhuber. Gödel machines: Fully self-referential optimal universal self-improvers. In B. Goertzel and C. Pennachin, editors, *Artificial General Intelligence.* Springer, 2007.
12. R. Solomonoff. A formal theory of inductive inference, part I. *Information and Control*, 7(1):1–22, 1964.
13. R. Solomonoff. A formal theory of inductive inference, part II. *Information and Control*, 7(2):224–254, 1964.
14. J. Zimmermann and A. B. Cremers. Making Solomonoff induction effective or you can learn what you can bound. In S. B. Cooper, A. Dawar, and B. Löwe, editors, *How the World Computes*, volume 7318 of *Lecture Notes in Computer Science.* Springer, 2012.