Emergent computation in CA: A matter of visual efficiency?

Mathieu S. Capcarrere[1]

Logic Systems Laboratory
School of Computer Science and Communication
Swiss Federal Institute of Technology, Lausanne
1015 Lausanne, Switzerland
mathieu@capcarrere.org

Cellular Automata as a computational tool have been the subject of interest from the computing community for many years now. However, WHAT is meant by computation with cellular automata remains elusive. In this paper, I will argue that emergent computation with CAs is a matter of visual efficiency. Basing our argument on past, recent and also previously unpublished results around the density task, I will propose a tentative definition of *emergent behavior*, in this limited scope, and thsu envisage differently the whole question of what may be sought in computing research in CAs. The practical consequences of this approach will alter the HOW question answer, and most notably how to evolve computing CAs.

[1]Now at: Computing Laboratory, University of Kent, Canterbury CT2 8DF, UK.

1.1 Introduction

Cellular Automata as a computational tool have been the subject of interest from the computing community for many years now. More precisely, the development of the Artificial Life field led many to wonder on how to do computation with such tools. Artificial Evolution which gave good results on specific tasks, like density or synchronisation was often given as an answer. However, it appeared that the limitations of such an approach were severe and really the question of WHAT meant computation with cellular automata became pregnant.

The answer to this question is far from obvious. Mitchell, Crutchfield, Hanson *et al* proposed an analysis of "particles" as a partial answer[7, 11]. Wuensche more recently developed the Z parameter as a paraphernalia to treating this question[22]. Before this question appeared in its full-blown form in the A-life/Computer science community, there were already interrogations going this way with Wolfram's class III and, related, Langton's computing at the edge of chaos.

In this tentative paper, I will argue that computation of CAs is a matter of visual efficiency. Basing our argument on past, recent and also previously unpublished results around the density classification task, I propose a definition of what is computation by means of CAs. This will be the occasion to (re)define emergent behaviour, in this limited scope, but also to envisage differently the whole question of what may be sought in computing research in CAs. The practical consequences of this approach will alter the HOW question answer, and most notably how to evolve computing CAs. Our point in this discursive paper is only to bring new paths of research rather than discarding old ones.

First, in section 1.2 I will shortly review the different ways computation in CAs has been tackled and define more precisely the kind of use CAs we are concerned with. Then in section 1.3, I will expose what is the density classification task and present the important results that are behind the reflections developed in this paper. Section 1.4 is, somehow, the core of this paper, where my thesis is detailed and argued for. This section will also be the occasion to open paths for future research in computing CAs.

1.2 Computation in cellular automata

There are four ways to view the computational aspects of Cellular Automata. *The first* and most natural one is to consider them as abstract computing systems such as Turing machines or finite state automata. These kind of studies consider what languages CA may accept, time and space complexity, undecidable problems, etc. There has been a wealth of work in this field [8, 13], most of which is based on or completes the mathematical studies of CAs. *The second way* is to develop structures inside the CAs (a specific initial configuration) which are able to perform universal computation. A prime example of this is the *Game of Life*, which was proved to be a universal computer using structures like gliders and glider guns to implement logic gates [3]. This was also applied to one-dimensional CAs where structures were found to be a universal Turing machine [15]. A *third way* is to view CAs as "computing black boxes". The initial configuration is then the input data and output is given in the form of some spatial configuration after a certain number of time steps. This includes so-called soliton or particle or collision based computing. In these systems, computing occurs on collision of particles carrying information [21]. In these works, usually, computation really occurs in some specific cells, i.e., most cells are only particle transmitters and some do the computation, this even if factually all cells are identical. Hence, parallelism is not really exploited as such. However, there are CAs belonging to this "black box" type which are arguably different. In these, like the density task solver that I will present in section 1.3 and which is demonstrated in Figure 1.1, computation is intrinsically parallel. Their solving property relies on each and every cell. All are involved in the computation and there is no quiescent state as such. The result is then to be considered globally. The frontier between these two sub-categories is not always clear, and in fact, the core of this distinction depends on the ill-defined notion of emergence. The latter type being so, the former not. We will come back to this discussion in section 1.4. Finally, there is a *fourth* way which is to consider CA computational mechanics. This kind of study concentrates on regularities, particles and exceptions arising in the spatial configuration of the CAs considered through time. This is really a study of the dynamical global behaviour of the system. This research does not concentrate on what is computed but rather how it is computed. Though it often took as object of study CAs of the emergent type of the third category, it was also applied to CAs with no problem solving aim, dissecting their behaviour with no further consideration. There has been much work in this domain, accomplished mainly by Hanson, Crutchfield and Mitchell [7] and Hordijk [12].

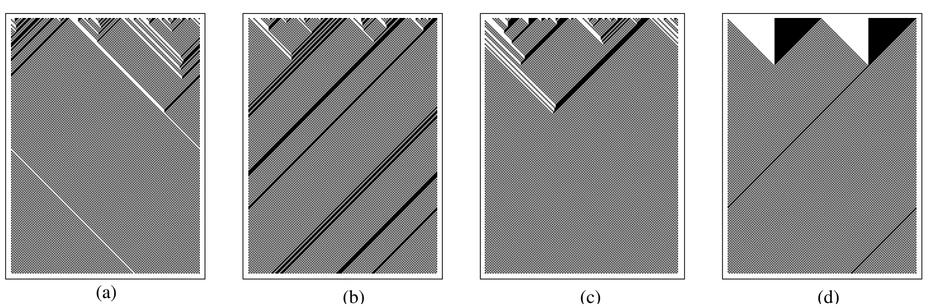

Figure 1.1: This figure presents an example of emergent computation in a one-dimensional, binary CA demonstrated on the density classification task. The rule in all four figures is 184 in Wolfram's notation. Time flows downward. When there are more ones than zeros, only left-going black lines remain, (b) and (d), while when there are more zeros than ones, only right-going white lines remain, (a). No lines are left when the density is exactly 0.5, (c).

In this paper, I propose to concentrate on the third kind of study. More particularly on the second type, the research of problem solving, emergent CAs. The interrogation on what is an emergent, global behaviour will be tackled more precisely in section 1.4. Let's first define more precisely through a few characteristics the kind of computation we are concerned with. – First, all cells are input (and output). This is an important property, in that it guarantees that there is no "storage" area and no separated "computing" area and that all cells are equally involved and necessary. The corollary of this is that there is no quiescent state as such. – Second, there is no special cell,i.e, no cell is actually doing the computation on its own. The previous characteristics prevent this to happen in most cases, but it is essential to explicitly express this as otherwise emergent computation, according to the definition to come in section 1.4 would not occur. – Third, the behaviour is *perfect*. This characteristic is all the more important as it has been often neglected, but by definition if computation occurs effectively, then it must be perfect. Otherwise we are facing some kind of pseudo-computation,some kind of cheating.

1.3 The density task example

In this section I shortly present the density task, demonstrated in figure1.1, as it is a prime example, simultaneously, of the kind of computation that we are looking for in CAs and of the importance of the look we lay on the CA .

The one-dimensional density task is to decide whether or not the initial configuration of a binary CA contains more than 50% 1s. Packard was the first to introduce the most popular version of this problem [17] where the output is the CA relaxing to a fixed-point pattern of all 1s if the initial density of 1s exceeds 0.5, and all 0s otherwise (Figure 1.1). As noted by Mitchell *et al.* [16], the density task comprises a non-trivial computation for a small radius CA ($r \ll N$, where N is the grid size). Density is a global property of a configuration whereas a small-radius CA relies solely on local interactions. Since the 1s can be distributed throughout the grid, propagation of information must occur over large distances (i.e., $O(N)$). The minimum amount of memory required for the task is $O(\log N)$ using a serial-scan algorithm, thus the computation involved corresponds to recognition of a non-regular language. After many fruitless attempts to find a perfect solution to this task, Land and Belew [14] proved that cannot be perfectly solved by a uniform, two-state CA of any radius However many imperfect CA performed relatively well, such as the GKL rule [9], that can correctly classify approximately 82% out of a random sample of initial configurations, for a grid of size $N = 149$ [1]. This lead some researchers to focus on the use of artificial evolution techniques to try to find *non-uniform* two-state CAs to solve this tasks ([20]). But these attempt remained unsuccessful. Actually, I proved recently that for this version of the density classification task, no two-state, non-uniform CA of any radius solves it perfectly[2].

Thus the density classification tasks seemed utterly impossible for two-state

[2] This unpublished results may be found in [4], chapter 4.

CAs (uniform and non-uniform). Nevertheless this was just an appearance and this impossibility only applies to the above statement of the problem, where the CA's final pattern (i.e, output) is specified as a fixed-point configuration. As we proved in [6], if we change the output specification, there exists a two-state, $r = 1$ uniform CA, rule 184 in Wolfram's notation, that can perfectly solve the density problem. Hence, this task, according to its definition, changes status dramatically in terms of "CA" computability. It moves from unsolvable for two-state CAs, to "basic" as it found a solution in the simplest class: the elementary CAs.

These results led us to wonder on what was required to solve that task in its essential form. Essential in the sense of stripping down its definition to the minimal form beyond which the task loses its meaning. From these minimal assumptions, we derived that a perfect CA density classifier must conserve in time the density of the initial configuration, and that its rule table must exhibit a density of 0.5 [5]. Thus, non-density-conserving CAs (such as the GKL rule) are by nature imperfect, and, indeed, any specification of the problem which involves density change precludes the ability to perform perfectly density classification.

The necessary condition of density conservation brings out the question of what is computation by means of cellular automata. As seen the computability of the task was more dependent on its definition rather than its inherent difficulty. Moreover, if we conserve density, then computation here is only re-ordering the "1s" among the "0s", which usually means simply *no computation*. In effect, usually computation is taken as a reduction of informational complexity. For instance, the original question was it not to reduce any configuration, 2^N bits, to an answer yes or no, 1 bit ? But here there seems to be no loss of information through time. However, this is not true of spatial information as rule 184 is not invertible. What we are looking for, hence, is a simplification of the input in terms of apprehensibility by an onlooker. So beyond the aid that this result on necessary conditions might give in the search for locally interacting systems that compute the global density property, it is its consequences on the question of computation that makes it important. Let's now extends on this question.

1.4 Emergent computation and visual efficiency

To answer the question of what is emergent computation with CAs, we shall first explicit the idea of emergence. For sake of brevity and of clarity, we will not engage in a discussion around this theme, but rather expose our own understanding of the term. Beneath the idea of emergence, there is always the key concept that *new properties* that emerge were not present at the level of the individual element. In fact, this is the only tangible aspect of emergence. But, by definition, a property has an effect. A property without effect has no existence. For instance one may say that the emergent property of social insect societies is robustness to attack and to environment hardship. The effect is then their ecological success. However, even if these properties and effects are objective, their consideration and definition is subjective. For instance, the necessity of describing a bunch of H_2O molecules as being liquid is merely a question of viewpoints. It appears as a necessity, given the dramatic effects of the liquid property, but one should remember that the liquid state of the matter is still ill-defined as of today. This inevitable subjectivity lead many to introduce the idea of an observer [2, 19]. The question thus becomes what or who is the observer? In other words, what is the correct viewpoint?

These considerations on emergence call for a reflection on the question of what constitutes emergent computation in CAs. I am not pretending, here, to give a definitive answer nor a universal one, but rather some tracks for thought. As evoked in the preceding section, what constitute the result of rule 184 is two-fold. First, the appropriate look makes it a perfect density-classifier and second this look is possible because of the *visual efficiency* of the final (or rather the temporal) configuration of the CA. That is, if one watches Figure 1.1, one can instantly say if the original configuration was holding more 1s than 0s or the contrary. Of course, this translates also into the fact that a simple three-state automaton can then classify the density, but the most convincing argument that CA 184 does the job is that it is visually efficient. It is hard to define formally what is this efficiency, but we can say, without doubt, at least in this case that it relies on patterns that become visibly obvious as they stand out of a regular background. This view is also what changed between the impossibility of the task in its original form to its evident solution. So the question of computation is finally the hazardous meeting of an actual computation by CAs with the "good" look from an outside observer. "[We] have to act like painters and to pull away from [the scene], but not too far. How far then? Guess!"[3].

The argument in favour of an observer, besides the result on the density classification task are multiple.– First and foremost, most CAs designed in the past have most often been (inappropriately ?) aimed at human, knowingly or unknowingly. For instance, Langton's loop, which does not lie in this category of computation, still relies on the impression of life it leaves in the eye of the spectator. Wolfram's classes are subjectively defined according to the judgement of the human beholder. And even the original definition of the density classification task was taking into account that fact. Effectively, even though it was defined according to a model close to the one of figure 1.2.a, that is of a classical computing machine taking an input and finishing in a yes or no state, the specification of the output itself, all 0s or all 1s, was clearly aimed at a human. – Second, beyond these historical considerations, and more importantly, in emergent systems, the observer is necessary to establish the emergent properties and its effect, but usually not for the property to happen. However in the case of CAs, computation only happens in the eye of the beholder. The CA in itself cannot get from any feedback, such as the environment for social insect and thus the effect of the property can only take place in the observer. This is due to the simple fact that no element, especially in simple, basic binary CA could "grasp" the global, emergent result...by definition. Maybe this is the main

[3] ¡¡Il faut que je fasse comme les peintres et que je m'en éloigne, mais non pas trop. De combien donc ? Devinez!¿¿ Blaise Pascal, pensées 479 in [18].

point of this article, CAs can't be self-inspecting and thus it is meaningless to consider its computation *in abstracto*, *in absentia* of an observer. In other word, **emergent computation in CA is epistemic**. So what we are looking for is really a system made out of a couple CA/beholder. – Finally, third, taking these considerations allows the result not to be a final configuration but rather the global temporal dynamic of the CA, such as illustrated in figure 1.2.b. This last point is interesting in two ways. On the one hand, it discards the need to look after n time steps, which solves one of the usual paradox. Solving the density task without a global counter, but requiring it to get the final result is rather paradoxical. On the other hand, now considering cycle and the global dynamic provides a much richer pool of potentially interesting CAs.

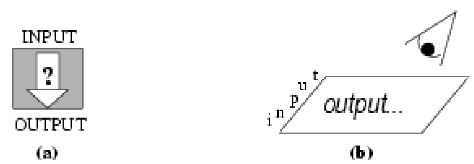

Figure 1.2: We need to change our look on CA computation to grab its full potentialities.

The necessity to consider the couple/CA beholder established[4], this says nothing on the nature of the observer. As said earlier, the natural observer is the human being. However the possibility to create an artificial observer is by no means not discarded. Actually, one could imagine a mechanical search (either exhaustive of by means or evolutionary computation) where the success criteria would be given by an artificial observer. Wuensche [22] proposed a measure of input entropy, based on the frequency of look up of the different neighbourhoods. This measure was devised in order to find "interesting" CAs, interesting in the sense of Wolfram's class III. Wuensche followed the ideas originated by Gutowitz and Langton: is there a quantitative way to define class III CAs and can we find them by means of artificial evolution [10]? All these works were somehow seeking an artificial observer, as Class III is only defined on criteria based on observation. We pursue the same goal but for a different purpose. As said before, interesting CAs for us were CAs which produced regularity, or more exactly irregularity (patterns) on regularity, and these are surely to be found in class II rather than class III. Nevertheless, Wuensche's frequency measure may be a good start for future work. He introduced the idea of filtering the output of the CAs by omitting in the *visual representation* the neighbourhoods that were used the most often. For instance, this would lead in CA 184 to discard patterns 101 and 010. This is definitely an interesting path to follow to discard regularity, and thereby provides a first step in the direction of an automatic observer.

[4] To the least proposed.

1.5 Concluding remarks

The nature of computation in cellular automata is a complex one. As we saw, there are four ways to consider this question, but only the global, emergent type, the second kind of the third type therein, was of real interest to us. In effect, only this type of computation (as defined in the last paragraph of section 1.2 presents the fascinating qualities that one seeks in cellular systems, namely, a global behaviour that goes beyond the capabilities of each of the elements and dismisses artifacts of sequential systems developed on a cellular substrate. But the nature, the essence of the computation of this kind is far from obvious. As seen in the past literature, the idea of the necessity of an observer to establish emergent behaviour is often present, but what we introduce here is the major distinction between ontological and epistemic emergence. In the former case, the system gets feedback from its environment and thus benefits directly from the effects of the emergent properties. The observer is then just there to state the property. In the latter case, however, the observer is the beneficiary from the effect of the emergent property, and as such the property cannot exist without it. We argued that CA lie in that latter category. Then the observer being mainly humans, emergent CA computation is thus merely a matter of visual efficiency.As gleaned from the density, this visually efficient computation occurs in creating order (regularity). To be more accurate, it creates a large "ocean" of regularity which allows the onlooker to see the distinguishing patterns of irregularity. Hence, one may conclude from this, and I would, that problem-solving truly emergent CAs are to be found in class II rather than Wolfram's class III CAs. I believe that this conclusion goes well beyond the density task. So a problem-solving CA is the fortunate meeting of a good look with a good local rule. However, this computation is not elusive, i.e, CA computes really. We are unable to see in the input arrangement and the local rule what will happen after a few time steps. CA computation thus hinges on the weakness of our mind in a certain way. This is not degrading at all and in fact is common to all sorts of computation. If one would see the same thing in $\sqrt{27225}/5$, and in 33, then there would be no need for calculators. This view of problem-solving CAs could have great influence on how one may find such CAs. For instance, by evolution, whereas today how the CA should behave globally is decided a priori, in the future an artificial observer could actually judge the fitness of the CA, the result being then the pair observer/CA.

Bibliography

[1] David Andre, Forrest H. Bennet III, and John R. Koza. Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem. In John R. Koza, David E. Goldberg, David B. Fogel, and R. L. Riolo, editors, *Genetic Programming 1996: Proceedings of the first conference*, pages 3–11, Cambridge, MA, 1996. The MIT Press.

[2] Nils A. Baas and Claus Emmeche. On emergence and explanation. *Intellectica*, 25:67–83, 1997.

[3] Elwin R. Berkelamp, John H. Conway, and R. K. Guy. *Winning ways for your mathematical plays*. Academic Press, 1982.

[4] Mathieu S. Capcarrere. *Cellular Automata and Other Cellular Systems: Design & Evolution*. Phd Thesis No 2541, Swiss Federal Inst. of Tech., Lausanne, 2002.

[5] Mathieu S. Capcarrere and Moshe Sipper. Necessary conditions for density classification by cellular automata. *Phys. Rev. E*, 64(3):036113/1–4, December 2001.

[6] Mathieu S. Capcarrere, Moshe Sipper, and Marco Tomassini. Two-state, r=1 cellular automaton that classifies density. *Phys. Rev. Lett.*, 77(24):4969–4971, December 1996.

[7] James P. Crutchfield and Melanie Mitchell. The evolution of emergent computation. *Proceedings of the National Academy of Science*, 23(92):10742, 1995.

[8] Karel Culik II, Lyman P. Hurd, and Sheng Yu. Computation theoretic aspects of cellular automata. *Physica D*, 45:357–378, 1990.

[9] Peter Gacs, G. L. Kurdyumov, and L. A. Levin. One-dimensional uniform arrays that wash out finite islands. *Problemy Peredachi Informatsii*, 14:92–98, 1978.

[10] Howard A. Gutowitz and Christopher Langton. Mean field theory of the edge of chaos. In F. Morán, A. Moreno, J. J. Merelo, and P. Chacon, editors, *Advances in Artificial Life, Proceedings of the 3rd European Conference on Artificial Life*, volume 929 of *LNAI*, pages 52–64. Springer-Verlag, 1995.

[11] James E. Hanson and James P. Crutchfield. Computational mechanics of cellular automata. *Physica D*, 103:169–189, 1997.

[12] Wim Hordijk. *Dynamics, Emergent Computation and Evolution in Cellular Automata*. Computer Science Dept, University of New Mexico, Albuquerque, NM (USA), Dec. 1999.

[13] Oscar Ibarra. Computational complexity of cellular automata: an overview. In M. Delorme and J. Mazoyer, editors, *Cellular Automata: A parallel model*. Kluwer Acadamecic Publishers, 1999.

[14] Mark Land and Richard K. Belew. No perfect two-state cellular automata for density classification exists. *Physical Review Letters*, 74(25):5148–5150, June 1995.

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

[16] Melanie Mitchell, James P. Crutchfield, and Peter T. Hraber. Evolving cellular automata to perform computations: Mechanisms and impediments. *Physica D*, 75:361–391, 1994.

[17] Norman H. Packard. Adaptation toward the edge of chaos. In J. A. S. Kelso, A. J. Mandell, and M. F. Shlesinger, editors, *Dynamic Patterns in Complex Systems*, pages 293–301. World Scientific, Singapore, 1988.

[18] Blaise Pascal. *Penses*. Folio/Gallimard, 1977. Edition de Michel le Guern.

[19] Edmund Ronald, Moshe Sipper, and Mathieu S. Capcarrre. A test of emergence, design, observation, surprise! *Artificial Life*, 5(3):225–239, 1999.

[20] Moshe Sipper and Ethan Ruppin. Co-evolving architectures for cellular machines. *Physica D*, 99:428–441, 1997.

[21] Kenneth Steiglitz, R. K. Squier, and M. H. Jakubow. Programmable parallel arithmetic in cellular automata using a particle model. *Complex Systems*, 8:311–323, 1994.

[22] Andrew Wuensche. Classifying cellular automata automatically: Finding gliders, filtering, and relating space-time patterns, attractor basins, and the z paramater. *Complexity*, 4(3):47–66, 1999.