Nat Comput (2018) 17:663–671
https://doi.org/10.1007/s11047-017-9650-1

A portfolio of classification problems by one-dimensional cellular automata, over cyclic binary configurations and parallel update

Marco Montalva-Medel[1] · **Pedro P. B. de Oliveira**[2] · **Eric Goles**[1]

Published online: 30 October 2017

Abstract Decision problems addressed by cellular automata have been historically expressed either as determining whether initial configurations would belong to a given language, or as classifying the initial configurations according to a property in them. Unlike traditional approaches in language recognition, classification problems have typically relied upon cyclic configurations and fully paralell (two-way) update of the cells, which render the action of the cellular automaton relatively less controllable and difficult to analyse. Although the notion of cyclic languages have been studied in the wider realm of formal languages, only recently a more systematic attempt has come into play in respect to cellular automata with fully parallel update. With the goal of contributing to this effort, we propose a unified definition of classification problem for one-dimensional, binary cellular automata, from which various known problems are couched in and novel ones are defined, and analyse the solvability of the new problems. Such a unified perspective aims at increasing existing knowledge about classification problems by cellular automata over cyclic configurations and parallel update.

Keywords One-dimensional cellular automata · Classification problem · Decision problem · Language recognition · Density · Parity · Emergent computation

✉ Marco Montalva-Medel
marco.montalva@uai.cl

[1] Universidad Adolfo Ibáñez, Av. Diagonal Las Torres 2640, Peñalolén, Santiago, Chile

[2] Universidade Presbiteriana Mackenzie, Rua da Consolação 896, Consolação, São Paulo 01302–907, Brazil

1 Introduction

Cellular automata (CAs) can be regarded as a model of distributed computation with the constraint that the processing units are only allowed to perform a local action. In their standard form, they consist of a d-dimensional lattice of cells with identical pattern of local connections, and a state transition rule that defines how each cell should have its state synchronously updated, according to the joint states of the cell itself and of its neighbours; in general the state can take on k discrete possibilities, assumedly varying from 0 to $k-1$ (Kari 2005; Wolfram 2002).

Computational problems addressed by cellular automata have taken the form of computing mathematical functions and attempting to solve decision problems (Gramß et al. 2005; Mazoyer and Yunès 2012). As far as decision problems are concerned, they have been typically expressed in either of two forms: *language recognition*, in which case a given cell of the lattice has to enter a specially defined accepting state, if the initial configuration represents a string in the language (Kutrib 2009; Terrier 2012), or *classification* of the initial configurations, according to the value of a property in them, so that the lattice has to converge to either of two homogeneous fixed points. Nevertheless, such a distinction in the formulation of these decision problems is merely due to historical reasons; after all, not only the formulations are potentially interchangeable, as well as alternative formulations could be defined for them.

Classification problems have been extensively studied, especially in the binary case, with cyclic configurations. This perspective includes the classical benchmark problem of determining whether the number of 1s in the initial configuration is larger than that of the 0s—the *Density Classification Problem* (Oliveira 2014), or the parity of the initial number of 1s—the *Parity Problem* (Betel et al. 2013).

While classification problems have typically relied upon cyclic configurations and two-way update of the cell states, language recognition with CAs have been traditionally based upon bounded configurations, and/or rules that operate on the cells in a one-way fashion (Kutrib 2009; Terrier 2012); for the sake of clarity, we can think of the two-way update as fully parallel, and to one-way update as sequential. The fact is that, with bounded configurations and/or one-way update, the action of the cellular automaton becomes much more controllable and simpler to analyse, which in turns reflects on many results on languages that can be recognised, their limitations, complexity classes and relationships with other models of computations. Furthermore, although the notion of cyclic languages, so to speak, has been studied in the wider realm of formal languages, only recently a more systematic attempt has come into play in respect to CAs with parallel update (Bacquey 2014), by touching the problems of binary density classification, as well as the recognition of cyclic regular languages.

Our purpose here is to push forward in the direction of a more systematic effort regarding CAs for classification problems with cyclic configurations and parallel update. We go about our objetive by proposing a unified definition of classification problem for one-dimensional, binary cellular automata, from which various known problems are couched in and novel ones are defined, and analyse the solvability of the new problems. By looking at all those problems under a unified perspective, our immediate motivation is to increase existing knowledge about classification problems by cellular automata over cyclic configurations and fully parallel update.

For present purposes we assume that each cell is binary ($k = 2$), the lattice is one-dimensional and subjected to periodic boundary conditions, which equates to any global state configuration of the lattice being a cyclic string.

In Sect. 2 we introduce the framework and show how it accounts for the classical problems of density and parity determination. Afterwards, novel classification problems are defined. The subsequent section provides analyses of the new problems previously defined, in terms of the possible existence of a CA rule, of any adequate space, that would solve them. The last section discusses various aspects of the work reported and make concluding remarks.

2 Classification problems in one-dimensional, cyclic, binary cellular automata

In this section we provide the notation and main definitions required for the paper, in particular, we introduce the general framework based upon which every classification problem is formalised (Definition 2).

2.1 Notation and basic definitions

Definition 1 *(Cellular Automaton—CA)* A *cellular automaton* (CA) is a fully discrete dynamical system defined by the quadruple (S, N, f, d), where $S \subset \mathbb{N}$ is the *state set*, $N = (\overrightarrow{v_1}, \overrightarrow{v_2}, \cdots, \overrightarrow{v_m}) \subset \mathbb{Z}^d$ is the *neighbourhood vector*, $f : S^m \longrightarrow S$ is the *local transition function* and $d \in \mathbb{N}^+$ is the *dimension* (Kari 2005). In particular, an *elementary cellular automaton* is a CA with $S = \{0, 1\}$, $N = (-1, 0, 1)$ and $d = 1$.

For present purposes, we assume one-dimensional CAs, with the size of the neighbourhood extending over a distance r (its *radius*) from the cell being considered; this means that $N = (-r, \ldots, -1, 0, 1, \ldots, r)$ in the case of symmetrical neighbourhood or $N = (-r, \ldots, -1, 0, 1, \ldots, r-1)$ in the asymmetrical case, thus leading to a neighbourhood with $m = \lfloor 2r + 1 \rfloor$ cells.

In the particular case of elementary CAs ($r = 1$), the 8 possible neighbourhoods entail $2^8 = 256$ possible rules, which are referred to by their Wolfram *number* (Wolfram 2002), given by

$$\sum_{(a_1,a_2,a_3)\in\{0,1\}^3} f(a_1, a_2, a_3) 2^{2^2 a_1 + 2^1 a_2 + 2^0 a_3}$$

For one-dimensional, binary CAs, a *configuration* is a map $c : \mathbb{Z} \longrightarrow \{0, 1\}$ that associates a state (0 or 1) to each integer, implying that a configuration is a bi-infinite binary sequence. By imposing periodic boundary condition on the lattice, the configurations remain infinite but spatially periodic (a *cyclic* configuration), for which there exists a $p \in \mathbb{N}^+ = \{1, 2, 3, \ldots\}$ such that $c_{i+p} = c_i, \forall i \in \mathbb{Z}$, that is, as if the cells are arranged over a ring.

For better referencing, Table 1 summarises the notation to be used in the paper and their associated meanings.

Definition 2 *(Classification Problem—CP)* For present purposes, a Classification Problem (CP) is a 5-tuple (S, $\mathcal{P}$, $\mathcal{Q}$, $\mathcal{A}_{\mathcal{P}}$, $\mathcal{A}_{\mathcal{Q}}$), consisting of

$$\text{CP}: \begin{cases} \text{A } \textit{definition set } S \subseteq \{0,1\}^* \\ \text{A } \textit{property set } \mathcal{P} \subseteq S \\ \text{The } \textit{complementary property set } \mathcal{Q} \equiv S \backslash \mathcal{P} \text{ (i.e., } S = \mathcal{P} \cup \mathcal{Q}) \\ \text{The } \textit{classification sets } \mathcal{A}_{\mathcal{P}} \subseteq \mathcal{P} \text{ and } \mathcal{A}_{\mathcal{Q}} \subseteq \mathcal{Q} \end{cases}$$

with the goal of deciding whether there exists a CA A with radius r and periodic boundary condition, such that

(a) If $x \in \mathcal{P}$, then x should converge to an arbitrary $u \in \mathcal{A}_{\mathcal{P}}$

(b) If $x \in \mathcal{Q}$, then x should converge to an arbitrary $v \in \mathcal{A}_{\mathcal{Q}}$

for all $x \in S$, with $|x| \geq \lfloor 2r + 1 \rfloor$, and where the elements of $\mathcal{A}_{\mathcal{P}}$ and $\mathcal{A}_{\mathcal{Q}}$ must be attractors of the CA A (i.e., limit cycles or fixed points). If such a CA exists, we say that the

Table 1 Notation and the associated meanings

Notation	Meaning
B^*	Setofallwordsobtainedoutofconcatenatingthewords insetB
1^* (0^*)	$1^* \equiv \{1\}^* = \{\epsilon, 1, 11, 111, \ldots\}$ $(0^* \equiv \{0\}^* = \{\epsilon, 0, 00, 000, \ldots\})$
$\overrightarrow{1}$ $(\overrightarrow{0})$	A non-empty finite configuration 1^j (0^j), $j \in \mathbb{Z}^+$
x^l	Selfconcatenationof$x \in \{0,1\}^*, l \in \mathbb{Z}^+$times $: x^l \equiv \underbrace{x \cdots x}_{l\text{times}}$ E.g. $: x = 011$and$l = 3 \Rightarrow x^l = (011)^3 = 011011011$
$x^{t_{expression}}$	Configuration$x$, atthetimedefinedby$t_{expression}$. E.g. $: x^{t=t_0}$ is theconfiguration x at time t_0
$\|x\|_w$	Numberofoccurrencesof$w$incyclicstring$x$, with$x, w \in \{0,1\}^*$ E.g. $: x = 10100$and$w = 01 \Rightarrow \|x\|_w = 2$
$\|x\|$	Size of configuration x, i.e., $\|x\| \equiv \|x\|_0 + \|x\|_1$
S_{all}	$S_{all} \equiv \bigcup_{n \geq 3} \{0,1\}^n$
S_{odd}	$S_{odd} \equiv \bigcup_{n \geq 3, n\, odd} \{0,1\}^n$
S_{even}	$S_{even} \equiv \bigcup_{n \geq 4, n\, even} \{0,1\}^n$

CP *has solution*. As such, $\mathcal{P}$ and $\mathcal{Q}$ can be regarded as a partition of S that we want to classify with CA A.

Remark 1 Three trivial consequences of well defined classification problems are:

(i) Any $x^{t=t_0} \in \mathcal{P}$ must verify $x^{t=t_0+1} \in \mathcal{P}$ (otherwise, $x^{t=t_0} \in \mathcal{P}$ could converge to a $v \in \mathcal{A}_\mathcal{Q}$).

(ii) If $\overrightarrow{0} \in \mathcal{A}_\mathcal{P}$ and $\overrightarrow{1} \in \mathcal{A}_\mathcal{Q}$ (or viceversa), then any CA with local rule f that solves a given classification problem must satisfy $f(\overrightarrow{0}) = 0$ and $f(\overrightarrow{1}) = 1$.

(iii) Closure: $x \in \mathcal{P} \Rightarrow x^l \in \mathcal{P}$.

Remark 2

1. The *Synchronisation Problem* in CAs—as in Fatès (2015), by which the rule would lead any initial configuration to a cyclic regime made up of configurations $\overrightarrow{1}$ and $\overrightarrow{0}$, alternating at subsequent time steps—is left out of our framework. Although acknowledging that it could naturally be extended so as to also account for that problem, this is beyond present purposes.
2. We are concerned here only with one-dimensional cellular automata, over cyclic binary configurations and parallel update, according to Definition 1.

Remark 3 Considering that the focus in Bacquey (2014) is language recognition, our definition of a classification problem is the exact counterpart of what is referred to therein as *strong recognition* of a language.

2.2 Classification problems

This section introduces several classification problems under the framework above, including six new ones. Let us start with the two most classical problems in the literature:

Definition 3 (Density Classification Task—DCT (Oliveira 2014))

$$\text{DCT}: \begin{cases} S = S_{odd} \\ \mathcal{P} = \{x \in S_{odd} : |x|_1 > |x|_0\} \\ \mathcal{Q} \equiv S_{odd} \backslash \mathcal{P} = \{x \in S_{odd} : |x|_1 < |x|_0\} \\ \mathcal{A}_\mathcal{P} = 1^* \cap S_{odd} \text{ and } \mathcal{A}_\mathcal{Q} = 0^* \cap S_{odd} \end{cases}$$

Definition 4 (Parity Classification Problem—PCP (Betel et al. 2013))

$$\text{PCP}: \begin{cases} S = S_{odd} \\ \mathcal{P} = \{x \in S_{odd} : |x|_1 \text{isodd}\} \\ \mathcal{Q} \equiv S_{odd} \backslash \mathcal{P} = \{x \in S_{odd} : |x|_1 \text{iseven}\} \\ \mathcal{A}_\mathcal{P} = 1^* \cap S_{odd} \text{ and } \mathcal{A}_\mathcal{Q} = 0^* \cap S_{odd} \end{cases}$$

Let us carry on with six new classification problems we are introducing here. As will be clear, they constitute problems related to the recognition of more complex languages than those associated with (the classical) DCT and PCP and, to some extent, variants inspired by them.

Definition 5 (*Striped Language Problem—SLP)*

$$\text{SLP}:\begin{cases} S = S_{even} \\ \mathcal{P} = \{x \in S_{even} : x = (0^k 1^k)^l \vee x = 1^{2kl},\ 2kl \geq 4\} \\ \mathcal{Q} \equiv S_{even} \backslash \mathcal{P} \\ \mathcal{A}_{\mathcal{P}} = 1^* \cap S_{even} \text{ and } \mathcal{A}_{\mathcal{Q}} = 0^* \cap S_{even} \end{cases}$$

Example 1 $x = (0^2 1^2)^2 = 00110011$ belongs to $\mathcal{P}$ but not $y = 0011000111$. So, if SLP has solution, then x should converge to $\overrightarrow{1} = 11111111$ and y to $\overrightarrow{0} = 0000000000$.

Remark 4 It can be shown that SLP refers to a context-sensitive language, in terms of Chomsky's hierarchy. This is in contrast with DCT and PCP which are related, respectively, to the recognition of a context-free language and a regular language, both of which are more restricted language types.

Definition 6 [*fg*-Classification Problem—*fg*-CP]

$$fg\text{–CP}:\begin{cases} S = S_{all} \\ \mathcal{P} = \{x \in S_{all} : x = (0^{f(k)} 1^{g(k)})^l \vee x = 1^{f(k)g(k)l}, f(k)g(k)l \geq 4\} \\ \mathcal{Q} \equiv S_{all} \backslash \mathcal{P} \\ \mathcal{A}_{\mathcal{P}} = 1^* \cap \mathcal{P} \text{ and } \mathcal{A}_{\mathcal{Q}} = 0^* \cap \mathcal{Q} \end{cases}$$

such that f and g are functions satisfying:

($fg1$) f and $(f+g)$ must be increasing functions with $f, g \geq 1$.
($fg2$) Given $m \in \mathbb{N}$, $\exists k_0 \geq m : (f+g)(k_0)$ is a prime number.

Example 2 The functions $f(k) = k$ and $g(k) = k+1$ verify both conditions ($fg1$) and ($fg2$) of *fg*-CP. In this case, configuration $x = (0^2 1^3)^2 = 0011100111$ would belong to $\mathcal{P}$, while $y = 0011000111$ would not. So, if *fg*-CP has solution, then x should converge to $\overrightarrow{1} = 1111111111$ and y to $\overrightarrow{0} = 0000000000$.

Remark 5 Although the definition of *fg*-CP was inspired by SLP, the former defines in fact a family of infinite (closed) languages, different from SLP, as the latter does not verify condition ($fg2$). Also, in order to analyse whether they have solution (as Propositions 2 and 3 address in the next section), it is necessary to use arguments related to configurations with lattice sizes multiple of prime numbers, unlike what happens with the further problems to be proposed below, where the closure property of Remark 1-(iii) plays an important role.

Definition 7 (*q-Absolute Classification Problem – q-ACP)*

$$q\text{–ACP}:\begin{cases} S = S_{all} \\ \mathcal{P} = \{x \in S_{all} : |x|_1 \geq q\} \\ \mathcal{Q} \equiv S_{all} \backslash \mathcal{P} = \{x \in S_{all} : |x|_1 < q, q \in \mathbb{Z}^+\} \\ \mathcal{A}_{\mathcal{P}} = 1^* \cap S_{all} \text{ and } \mathcal{A}_{\mathcal{Q}} = 0^* \cap S_{all} \end{cases}$$

Example 3 Let $q = 2$. It is easy to check that $x = 01011 \in \mathcal{P}$ and $y = 10000 \in \mathcal{Q}$. So, if 2-ACP has solution, then x should converge to $\overrightarrow{1} = 11111$ and y to $\overrightarrow{0} = 00000$.

The idea of q-ACP is to define a classification problem according to the absolute number of 1s in the lattice, regardless of its size. In doing so, q-ACP defines a simpler problem than the variations that include relative threshold densities of 1s (see details in Oliveira 2014), from which the standard DCT is a special case, with threshold density 0.5. As a consequence, q-ACP departs from the latter in that the lattice size can now be odd or even.

Let $n \geq 3$, $x \in \{0,1\}^n$, $w \in \mathbb{R}^n$, the scalar product between w and x vectors $w \cdot x = \sum_{i=1}^n w_i x_i$ and a parameter $\alpha(n) > 0$. The following is yet another new classification problem that generates a huge family of classification problems to analyse, especially associated with numerical series.

Definition 8 *((S,w,α)-Numerical Classification Problem – (S,w,α)-NCP)*

$$(S,w,\alpha)\text{–NCP}:\begin{cases} S \subseteq S_{all} \\ \mathcal{P} = \{x \in S : w \cdot x \geq \alpha(n) \wedge n = |x|\} \\ \mathcal{Q} \equiv S \backslash \mathcal{P} = \{x \in S : w \cdot x < \alpha(n) \wedge n = |x|\} \\ \mathcal{A}_{\mathcal{P}} = 1^* \cap S \text{ and } \mathcal{A}_{\mathcal{Q}} = 0^* \cap S \end{cases}$$

Evidently, the choice of (S,w,α) should be such that (S,w,α)-NCP is well defined in the sense that always $\overrightarrow{1} \in \mathcal{P}$ and $\overrightarrow{0} \in \mathcal{Q}$. For example, if $w = \overrightarrow{0}$, then the (S,w,α)-NCP is not well defined because all configurations (in particular $\overrightarrow{1}$) will belong to $\mathcal{Q}$.

In this context, DCT and q-ACP are particular cases of (S,w,α)-NCP, namely:

– $\text{DCT} \equiv \left(S_{odd}, \overrightarrow{1}, \frac{n}{2}\right)\text{–NCP}$
– $q\text{–ACP} \equiv \left(S_{all}, \overrightarrow{1}, q\right)\text{–NCP}$

With (S,w,α)-NCP in mind, we can derive an interesting instance of it, as defined below:

Definition 9 *(α-Harmonic Classification Problem – α-HCP)*

$\alpha\text{–HCP} \equiv \left(S_{all}, w^h, \alpha\right)\text{–NCP}$

where $w_i^h = \frac{1}{i}$, $i \in \{1,\ldots,n\}$ and $0 < \alpha \leq \sum_{i=1}^n \frac{1}{i}$.

This classification problem is well defined because if $x = \overrightarrow{1}$ then $w^h \cdot x = \sum_{i=1}^n w_i^h x_i = \sum_{i=1}^n w_i^h = \sum_{i=1}^n \frac{1}{i}$ is the harmonic series (when $n \to \infty$) and $w^h \cdot x \geq \alpha$, i.e.,

$x = \overrightarrow{1} \in \mathcal{P}$. Similarly, $x = \overrightarrow{0} \in \mathcal{Q}$. Clearly, if $\alpha > \sum_{i=1}^{n} \frac{1}{i}$ then the above classification problem is ill defined.

Example 4 Consider $\alpha = \sum_{i=1}^{9} \frac{1}{i}$ and $w^h = \left(1, \frac{1}{2}, \ldots, \frac{1}{8}, \frac{1}{9}\right)$. If $y = 011000001$, then $w^h \cdot x = \frac{1}{2} + \frac{1}{3} + \frac{1}{9} < \alpha$, hence, $y \in \mathcal{Q}$. On the other hand, it easy to see that $\mathcal{P} = \{111111111\}$, because of Proposition 5-(i). So, if α-HCP has solution, then y should converge to $\overrightarrow{0} = 000000000$.

Finally, we introduce a classification problem related with the odd-even nature of the distance between two consecutive 1s in the initial configuration:

Definition 10 (*Parity Distance Classification Problem—ParDisCP*)

$$\text{ParDisCP}: \begin{cases} S_{all} \\ \mathcal{P} = \{x \in S_{all} : \exists i,j,\ x_i = x_j = 1 \land d_0(x_i, x_j) is even\} \\ \mathcal{Q} \equiv S_{all} \backslash \mathcal{P} \\ \mathcal{A}_{\mathcal{P}} = 1^* \cap S_{all} \ and\ \mathcal{A}_{\mathcal{Q}} = 0^* \cap S_{all} \end{cases}$$

where we suppose that $\overrightarrow{0} \in \mathcal{Q}$ and $d_0(x_i, x_j)$ is the number of 0s between the elements x_i and x_j of the vector $x \in S_{all}$.

Example 5 For $x = 01011000$ there exists $x_4 = x_5 = 1$ such that $d(x_4, x_5) = 0$, i.e., $x \in \mathcal{P}$ but $y = 01000101 \in \mathcal{Q}$. So, if ParDisCP has solution, then x should converge to $\overrightarrow{1} = 11111111$ and y to $\overrightarrow{0} = 00000000$.

Remark 6 In ParDisCP, all odd-sized configurations are in $\mathcal{P}$ with the only exception of $\overrightarrow{0}$. So, it is straightforward that the OR rule (i.e., elementary rule 254) solves this problem for odd-sized configurations.

Remark 7 Since ParDisCP allows any number of 1s in the input string, it generalises the *Even-Odd Classification Problem* introduced in Auer et al. (2011), where only two 1s are allowed.

In the next sections, we study the existence (or non-existence) of solutions in classification problems, with respect to the previous problems.

3 Analysing the existence of solution in classification problems

In this section we delve into the details of the existence proofs of solution of the problems we proposed; but we start by observing some important facts regarding the most classical classification problems.

Remark 8 Observe that:

(i) Both PCP and DCT are defined over odd-sized configurations (i.e., over S_{odd}) in contrast with q-ACP, which allows for any kind of configurations (i.e., S_{all}).

(ii) PCP has solution with the constraints supposed until now; one-dimensional CA with periodic boundary conditions. More precisely, Betel et al. (2013) shows that there is no solution with $r = 2$ and there is solution with at least $r = 4$; it still open whether $r = 3$ admits a solution.

(iii) DCT has no solution in one dimension, with periodic boundary conditions (see Land and Belew 1995, for instance). In fact, there is no solution for DCT, with any number of dimensions or any number of states (Chau et al. 1999).

The following results show the non-existence of solution for SLP and fg-CP. In both cases, the size of a configuration is a key factor when its length is a multiple of a prime number; this situation might give insights for analysing the solution of other classification problems.

3.1 SLP has no solution

Proposition 1 *Let p be a prime number. If SLP has solution, then the configuration $x^{t=t_0} = 0^p 1^p$ verifies that $x^{t=t_0+1} = (01)^p$ or $x^{t=t_0+1} = \overrightarrow{1}$.*

Proof If p is a prime number, SLP has solution and $x^{t=t_0} = 0^p 1^p$ then, by Remark 1(i), $x^{t=t_0+1}$ must belong in $\mathcal{P}$, that is, either $x^{t=t_0+1} = \overrightarrow{1}$ or there exist numbers k and l such that $x^{t=t_0+1} = (0^k 1^k)^l$. Since $|x^{t=t_0}| = |x^{t=t_0+1}|$, then $2p = 2kl$, or, $p = kl$. But, since p is a prime number, there are only two possibilities: $[k = p \land l = 1]$, or $[k = 1 \land l = p]$. □

Proposition 2 *SLP has no solution.*

Proof Suppose, on the contrary, that there exists a CA with radius r that solves the problem, that is, the CA converges a configuration $x^{t=0}$ to $\overrightarrow{1}$, if and only if, $x^{t=0} = \overrightarrow{1}$ or $x^{t=0} = (0^k 1^k)^l$ with $2kl > 2r + 1$. In particular, let us consider the initial configuration $x^{t=0} = 0^p 1^p$, such that p is a prime number and $p > > r$ (large enough). By Proposition 1, $x^{t=1} = (01)^p$ or $x^{t=1} = \overrightarrow{1}$. In both cases, the above implies that there exists cell position $j = \lceil \frac{p}{2} \rceil$ or $j = \lceil \frac{p}{2} \rceil - 1$ such that $x_j^{t=1} = f(\overrightarrow{0}) = 1$, which contradicts Remark 1(ii). □

3.2 An infinite family of problems with no solution

Proposition 3 *fg -CP has no solution.*

Proof Once again, let us suppose, on the contrary, that there exists a CA with radius r that solves the problem. Considering $m >> 2r+1$ (m large enough), condition ($fg2$) ensures that $\exists k_0 \geq m : (f+g)(k_0) = p$, p prime. In particular, the initial configuration $x^{t=0} = 0^{f(k_0)}1^{g(k_0)} \in \mathcal{P}$ has prime length p. From the definition of $\mathcal{P}$, we necessarily have that $x^{t=1} = F(x^{t=0}) = \overrightarrow{1}$. In fact, if $x^{t=1} = F(x^{t=0}) = (0^{f(k_1)}1^{g(k_1)})^{l_1}$ for some k_1 and l_1 in $\mathbb{N}$, then, $|x^{t=0}| = |x^{t=1}| \Rightarrow f(k_0) + g(k_0) = p = [f(k_1) + g(k_1)] \quad l_1 \Rightarrow [f(k_1) + g(k_1) = p \wedge l_1 = 1]$ or $[f(k_1) + g(k_1) = 1 \wedge l_1 = p]$. The first case means that a fixed point different from $\overrightarrow{1}$ would have been reached, and the second case is impossible, since $f(k_1) + g(k_1) = 1$ would contradict ($fg1$). Therefore, $x^{t=1} = \overrightarrow{1}$. Since $k_0 \geq m$ and $m >> 2r+1$ (m large enough), by reasoning similarly to the proof of Proposition 2, there exists a cell position $j = \lceil \frac{f(k_0)}{2} \rceil$ such that $x_j^{t=1} = f(\overrightarrow{0}) = 1$, once again contradicting Remark 1(ii). □

3.3 Problems with no general solution

The idea behind the two following propositions is to show that the classification problems involved have a partial solution depending on the parameters that characterise them.

Proposition 4 *For $q \in \mathbb{Z}^+$, q -ACP only has solution for* $q = 1$.

Proof For $q = 1$ we have that

$$\text{1-ACP:} \begin{cases} S = S_{all} \\ \mathcal{P} = \{x \in S : |x|_1 \geq 1\} \text{ and } \mathcal{Q} = S \backslash \mathcal{P} = 0^* \cap S \\ \mathcal{A}_\mathcal{P} = 1^* \cap S \text{ and } \mathcal{A}_\mathcal{Q} = \mathcal{Q} \end{cases}$$

Therefore, putting it simply, the problem consists of separating the configuration with only 0s (the set $\mathcal{Q}$) from all the others (the set $\mathcal{P}$). Clearly, any CA rule where the 1-state is the output of all neighbourhoods, except the one made up of only 0s, solves the problem; this is the case, for instance, of elementary rule 254, whose operation can be represented by the Boolean function $\phi(x,y,z) = x \vee y \vee z$, therefore meaning that any configuration having at least one 1 will converge to $\overrightarrow{1}$ (i.e., 1 is the rule's so-called *spreading* state).

If $q > 1$, the problem does not verify the closure property of Remark 1-(iii) and, consequently, has no solution. In fact, considering $x \in \mathcal{Q}$ such that $|x|_1 = q-1$, then the concatenated configuration $z = xx = x^2$ is such that $|z|_1 = |x^2|_1 = |x|_1 + |x|_1 = 2q - 2 \geq q$, i.e., $z = x^2 \in \mathcal{P}$. □

Proposition 5 *For α -HCP:*

(i) *If $0 < \alpha(n) \leq \frac{1}{n}$ or $\alpha(n) = \sum_{i=1}^{n} \frac{1}{i}$, $\forall n \geq 3$, then α -HCP has solution.*

(ii) *If $\frac{1}{n} < \alpha(n) < \sum_{i=1}^{n} \frac{1}{i}$, $\forall n \geq 3$ and $\lim_{n \to \infty} \alpha(n) = L \in \mathbb{R}^+$, then α-HCP has no solution.*

Proof (i) Suppose that $0 < \alpha(n) \leq \frac{1}{n}$, $\forall n \geq 3$, then it is easy to see that any configuration $x \neq 0^n$ is such that $w^h \cdot x \geq \frac{1}{n} \geq \alpha(n)$, i.e., $x \in \mathcal{P}$ meaning that the classification problem consists of separating the configurations with only 0s (the set $\mathcal{Q}$) from all the others (the set $\mathcal{P}$) and thus, it is equivalent to 1-ACP that has solution according to Proposition 4. A similar reasoning when $\alpha(n) = \sum_{i=1}^{n} \frac{1}{i}$, $\forall n \geq 3$, leads us to conclude that the classification problem consists of separating the configurations with only 1s (the set $\mathcal{P}$) from all the others (the set $\mathcal{Q}$), a discrimination that can be achieved by elementary rule 128, given by the Boolean function $\phi(x,y,z) = x \wedge y \wedge z$, which entails that any configuration having a 0 somewhere will converge to $\overrightarrow{0}$ with the only exception of $\overrightarrow{1}$ (i.e., 0-state is spreading for rule 128).

(ii) Let $x = 0^{k-1}1 \in \{0,1\}^k$, $k \in \mathbb{N}$, $k \geq 3$, then $w^h \cdot x = \frac{1}{k} < \alpha(k)$, i.e., $x \in \mathcal{Q}$ which must converge to $v = 0^k \in \mathcal{A}_\mathcal{Q}$. On the other hand, for $l \in \mathbb{N}$, $w^h \cdot x^l = \frac{1}{k} + \frac{1}{2k} + \cdots + \frac{1}{kl} = \frac{1}{k}\sum_{i=1}^{l} \frac{1}{i}$. Since $\lim_{n \to \infty} \alpha(n) = L \in \mathbb{R}^+$ and $\frac{1}{k}\sum_{i=1}^{l} \frac{1}{i}$ diverges when $l \to \infty$, $\exists l_0 \in \mathbb{N}$ such that $w^h \cdot x^l \geq L$, $\forall l \geq l_0$. In particular, for $l_1 > l_0$ large enough, $w^h \cdot x^{l_1} \geq \alpha(|x^{l_1}|)$, i.e., $x^{l_1} \in \mathcal{P}$. Since $v^{l_1} = 0^{kl_1} \notin \mathcal{A}_\mathcal{P}$, then $\exists x = 0^{k-1}1 \in \mathcal{Q}$, $k \in \mathbb{N}$, $k \geq 3$ and $\exists l_1 > 1 : x^{l_1} \in \mathcal{P} \wedge v^{l_1} \notin \mathcal{A}_\mathcal{P}$. Consequently, α-HCP does not verify the closure property and, therefore, has no solution. □

Remark 9 An interesting case is when $\alpha = \frac{n-1}{n}$, for which the proposition above ensures that $\frac{n-1}{n}$-HCP has no solution. However, the same proposition entails that $(1-\alpha)$-HCP has solution.

Naturally, the lack of closure of a classification problem suffices for its lack of a solution. However, closure alone of a problem is neither necessary nor sufficient for the existence of a solution. So, on one hand, we know that DCT verifies the closure property, but does not have solution. In contrast, it is easy to realise that ParDisCP verifies the closure property, but its solvability is yet to be fully established, since we can so far only ascertain the existence of a solution for odd sized configurations. Notwithstanding such an openness, the full solvability of the problem can be established by composing elementary rules, as shown below.

Theorem 1 *ParDisCP verifies the closure property and can be solved with the OR rule applied (i.e., elementary rule 254) $\mathcal{O}(n)$ times followed by elementary rule 200.*

Proof First, it is easy to check the following facts:

(1) The OR rule leads every configuration with substring 11 to converge to $\overrightarrow{1}$ in $\mathcal{O}(n/2)$ time steps.
(2) Configurations $\overrightarrow{0}$ and $\overrightarrow{1}$ are fixed points both for the OR and 200 elementary rules.
(3) Because 010 is the only active neighbourhood configuration for elementary rule 200—i.e., 010 is the only neighbourhood configuration that causes a state change—if $x \in [(01)^* \cup (10)^*] \cap S_{even}$, then the rule always converges to $\overrightarrow{0}$ in one time step.

Let us consider even-sized configurations. If $x^{t=0} \in \mathcal{P}$ does not include substring 11, then there exist i and j such that $x_i^{t=0} = x_j^{t=0} = 1$ and the number of 0s between $x_i^{t=0}$ and $x_j^{t=0}$ is even. So, in the next iteration, due to the OR rule, $x_{i+1}^{t=1} = x_{j-1}^{t=1} = 1$, and the process goes on for $\mathcal{O}(n/2)$ time steps, when substring 11 will appear and converge to $\overrightarrow{1}$ in further $\mathcal{O}(n/2)$ time steps, according to Fact (1). After that, configuration $\overrightarrow{1}$ is left unchanged by elementary rule 200, in tune with Fact (2).

If $x^{t=0} \in \mathcal{Q}$, since $|x^{t=0}|$ is even, then every 1 is surrounded by 0s and every distance between consecutive 1s is odd. Thus, before $n/2$ time steps, the OR rule converges $x^{t=0}$ to the limit cycle $x \longleftrightarrow \neg x$ where $x \in [(01)^* \cup (10)^*] \cap S_{even}$. Finally, elementary rule 200 leads x to converge to $\overrightarrow{0}$, in accordance with Fact (3).

Finally, as previously stated in Remark 6, for odd-sized configurations the OR rule alone solves ParDisCP. □

The main results obtained so far are summarised in Table 2, together with accounts on how the domain sets are involved in the classification problems previously identified, and on whether the property of a configuration depends on its lattice size or not.

4 Discussions and concluding remarks

Traditional approaches for language recognition by cellular automata have not relied on cyclic languages, so that only recently more systematic efforts have been made towards them (Bacquey 2014). Although the possibility of using bounded configurations allows for the definition of a much larger spectrum of languages, the traditional use of one-way communication between the cells is a limiting feature that avoids using the free interaction among cells, which, in a sense, departs from using CAs in their full nature. So, studying cyclic languages with fully parallel update seems to be a sensible stepping stone towards employing cellular automata in their full potential.

Since the approaches in classification problems with cellular automata have been traditional in tune with cyclic language recognition and fully parallel update, here we couched the latter in terms of the former, and proposed a unified definition of classification problem for one-dimensional, binary cellular automata, from which a number of problems were discussed, and the new ones analysed in terms of their solvability capacity.

Notwithstanding the settings we relied upon, various others can also be envisaged. As such, as far as the CA setting is concerned, the following are found in the literature related to density classification: non-periodic boundary condition, most notably fixed; non-uniform cellular automata; rules with memory; stochastic rules; asynchronous update schemes; infinite size lattices; binary solutions with multiple states; multidimensional definition of the problems, thus requiring rules operating over multidimensional lattices; as well as, whenever possible, problem definition beyond binary strings, by means of multi-state cellular automata. A detailed account on these variations with respect to DCT is discussed in Oliveira (2014). Also, although we restricted ourselves to standard decision problems, one might naturally consider multi-valued classification problems, where our original definition could be generalised as:

$$\text{CP}: \begin{cases} \text{A } \textit{definition set } S \subseteq \{0,1\}^* \\ \text{A partition } \mathcal{P}_1, \ldots, \mathcal{P}_k \text{ of } S \text{ (with each } \mathcal{P}_j \text{ representing a property)} \\ \text{The classification sets } \mathcal{A}_{\mathcal{P}_j} \subseteq \mathcal{P}_j, j \in \{1, \ldots, k\} \end{cases}$$

and with the goal of deciding whether there exists a CA A with radius r and periodic boundary condition, such that, if $x \in \mathcal{P}_j$, then x should converge to an arbitrary $u_j \in \mathcal{A}_{\mathcal{P}_j}$,

Table 2 Summary of some features in the main classification problems discussed in the paper

	Domain	Dependence of $\mathcal{P}$ on lattice size	Closure	Existence of solution
DCT	S_{odd}	YES	YES	$\nexists$ (seeLandandBelew1995)
PCP	S_{odd}	NO	YES	$\exists$ (seeBeteletal.2013)
q-ACP	S_{all}	NO	YES[$q=1$] (Prop.4) NO[$q \geq 2$] (Prop.4)	$\exists$ ($q=1$) $\nexists$($q \geq 2$) (Prop.4)
α-HCP	S_{all}	YES, α constant NO, otherwise	YES (Prop.5i) NO (Prop.5ii)	YES (Prop.5i) NO (Prop.5ii)
ParDisCP	S_{all}	NO	YES	OPEN ($n$even)
SLP	S_{even}	YES	YES	$\nexists$ (Prop.2)
fg-CP	S_{all}	YES	YES	$\nexists$ (Prop.3)

$j \in \{1, ..., k\}$, for all $x \in S$, with $|x| \geq \lfloor 2r+1 \rfloor$, and where the elements of $\mathcal{A}_{\mathcal{P}_j}$ must be attractors of the CA A.

The key point about these possible variations is that, although a given classification problem would be solvable in one setting, it might not be in another; this is the case addressed herein about ParDisCP by which, although it may not have solution in the single rule formulation, a composite-solution does exist.

Another interesting question is what determines that a problem can be solved or not. Notice that, while the notion of recognising the most prevailing state in the lattice, as defined in DCT, can be couched in terms of the recognition of the context-free language $\{0^n 1^n\}, n \in \mathbb{Z}^+$, the notion of parity of the number of 1s, that defines the parity problem, can be translated into the recognition of a regular language of the type $((0^*10^*)(10^*10^*)^*)^+$, which represents odd quantities of 1s. Such a difference is not with a consequence: while there is no solution for solving DCT, at least one rule is known that is able to solve the parity problem (Betel et al. 2013); furthermore, as mentioned earlier, SLP can be shown to be context-sensitive, and we showed it has no solution. Extending the rationale, one might wonder which would be the simplest possible problem, defined within the domain of cyclic regular languages, that might still be useful for probing the computational ability of cellular automata to solve global problems. It is our expectation that addressing cases like this in detail might lead to deeper insights into the CA's computational power and flexibility. All this is very interesting in that it suggests that the corresponding machine type, and maybe its size in terms of its number of states, might provide a measure to indicate the solvability of a classification problem by cellular automata.

Acknowledgements This work was partially supported by FONDECYT 1140090 (E.G.), FONDECYT Iniciación 11150827 (M.M-M.), Basal CMM, as well as the Brazilian funding agencies, Mack-Pesquisa—Fundo Mackenzie de Pesquisa and FAPESP.

References

Auer C, Wüchner P, De Meer H (2011) Target-oriented self-structuring in classifying cellular automata. J Cell Autom 6(1):25–52

Bacquey N (2014) Complexity classes on spatially periodic cellular automata. In: Mayr EW, Portier N (eds) 31st International symposium on theoretical aspects of computer science (STACS 2014), Leibniz international proceedings in informatics (LIPIcs), vol. 25, pp 112–124. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany. http://drops.dagstuhl.de/opus/volltexte/2014/4451

Betel H, de Oliveira PPB, Flocchini P (2013) Solving the parity problem in one-dimensional cellular automata. Nat Comput 12(3):323–337

Chau HF, Siu LW, Yan KK (1999) One dimensional n-ary density classification using two cellular automaton rules. Int J Mod Phys C 10(05):883–887

de Oliveira PPB (2014) On density determination with cellular automata: results, constructions and directions. J Cell Autom 9(5–6):357–385

Fatès N (2015) Remarks on the cellular automaton global synchronisation problem. Springer, Berlin, Heidelberg, pp 113–126. https://doi.org/10.1007/978-3-662-47221-7_9

Gramß T, Bornholdt S, Groß M, Mitchell M, Pellizzari T (2005) Computation in cellular automata: a selected review. Wiley-VCH Verlag GmbH & Co. KGaA, Hoboken, pp 95–140. https://doi.org/10.1002/3527602968.ch4

Kari J (2005) Theory of cellular automata: a survey. Theor Comput Sci 334(1):3–33

Kutrib M (2009) Cellular automata and language theory. In: Meyers RA (ed) Encyclopedia of complexity and systems science. Springer, Berlin, pp 800–823

Land M, Belew RK (1995) No perfect two-state cellular automata for density classification exists. Phys Rev Lett 74(25):5148–5150

Mazoyer J, Yunès JB (2012) Computations on cellular automata. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-92910-9_5

Terrier V (2012) Language recognition by cellular automata. In: Rozenberg G, Bäck T, Kok JN (eds) Handbook of natural computing. Springer, Berlin, pp 123–158

Wolfram S (2002) A new kind of science, vol 5. Wolfram Media, Champaign