Template Based Representation of Cellular Automata Rules

Pedro P.B. de Oliveira[1,2] and Maurício Verardo[2]

Universidade Presbiteriana Mackenzie
Faculdade de Computação e Informática[1] & Pós-Graduação em Engenharia Elétrica[2]
Rua da Consolação 896, Consolação – 01302-907 São Paulo, SP – Brazil
pedrob@mackenzie.br , mauricio.verardo@gmail.com

Abstract. A more general form of representing cellular automata rules, than the standard rule table based representation, is discussed, which relies on the notion of *templates*. As a generalisation of the standard rule table, it allows to account not only for a single rule, but for subsets of cellular automata rule spaces. Key for our approach to obtaining a template is the role of the built-in equation-solving capabilities of the *Mathematica* software. The applicability of the notion is illustrated in the simple context of finding representations for families of elementary rules that share the properties of maximum internal symmetry and/or number-conservation. Perspectives for using templates in further contexts are discussed and current limitations are pinpointed.

Keywords: Cellular automaton, rule space, rule table, template based representation, *k*-ary representation, number conservation, internal symmetry.

1. Introduction

In order to better understand how complex behaviour emerges in cellular automata (CAs), many explorations have been made in the context of the power implicit in CA rules. For instance, classical benchmark problems have been used for this, including the *density classification task* [de Oliveira, 2014] and the *parity problem* [Betel, de Oliveira and Flocchini, 2013]. One of the approaches in these contexts is to evaluate every possible CA of a given family, in terms of its capabilities to solve the target problem. This approach is possible in small CA families, like the elementary space (composed of 256 CAs), but is unfeasible in larger families, like the one-dimensional, binary CAs, with radius 3, composed of 2^{128} rules; in these cases, all we can expect is to probe a sample of the space, guided by some direct sampling criterion or by informed searches [Wolz and de Oliveira, 2008].

An alternative is to directly characterise a given subspace, where all its CAs share specific properties. Here, we discuss the concept of a CA *template*, as a possible way to achieve this goal. A CA *template* is a tuple associated with the rule tables of the members of a CA family, that relies on the use of variables. The introduction of these variables makes it possible for to represent a set of rules, unlike the standard, *k*-ary rule table representation, that can only account for an individual CA. By defining equations concerning a set of CAs that share a given property and, by solving them, making use of the built-in equation-solving capabilities and algorithms of the *Mathematica* software [Wolfram Research, 2014], we are able to create templates that represent number-conserving CAs, those with any kind of maximum internal symmetry (i.e., the properties derived from conjugation, reflection and their compositions [Wolfram, 2002]), as well as any composition of the latter. These cases are given here as examples of the applicability of the templates, but other properties can also be accounted for.

This paper is a significantly stripped down version of a much longer, implementation oriented account, that will appear as [de Oliveira and Verardo, 2014]; in essence they report on the same work, although with distinct emphasis according to the audience. The next section characterises the properties of maximum internal symmetry and number-conservation. Section 3 explains the notion of *template* and exemplifies its use in the very simple context of elementary CAs. Section 4 wraps up the presentation, points at present limitations of using templates, as well as the research directions we are currently pursuing.

3. Cellular Automata Properties

3.1 Number-Conservability

Number-conservability is a property presented by some CAs, in which the sum of the states of the individual cells in any initial configuration does not change during the space-time evolution; in particular, for binary CAs, this means that the number of 1s always remains the same. Number conservation is an important property in various domains and remains an active topic of study [Kari and Le Gloannec, 2012]. Elementary CA 184 is an example of a number-conserving CA.

In order for a one-dimensional CA rule to be number-conserving, Boccara and Fukś established in [Boccara and Fukś, 2002] that the local rule f with neighborhood size n must respect the following necessary and sufficient conditions, for every state transition,

$$f(x_1, x_2, x_3, \dots, x_n) = x_1 + \sum_{i=1}^{n-1} f(0_1, 0_2, \dots, 0_i, x_2, x_3, \dots, x_{n-i+1}) - f(0_1, 0_2, \dots, 0_i, x_1, x_2, \dots, x_{n-i}),$$

where $0_1, 0_2, \dots, 0_i$ corresponds to a sequence of 0s of size i, and n is the neighbourhood size ($2r$+1, r being the radius). Notice that the latter translates into a series of equations that any conservative rule must abide by.

But a simplification of the latter, given in [Schranko and de Oliveira, 2010], showed that it suffices to analyse the state transitions associated with the neighbourhood made up of only 0s, as well as the neighborhoods not starting with 0, therefore totalling k^n-k^{n-1}+1 neighbourhoods, instead of the original k^n. The simplified conditions are the ones used herein to obtain templates that represent number-conserving CAs.

3.2 Internal Symmetry

Given the rule table of a CA, three types of transformations can be applied on it that results in dynamically equivalent rules. For binary CAs, the first, black and white transform (or, conjugation), is obtained by switching the state of all cells in a rule table. The second type of transformation is obtained by reversing the bits of the neighbourhoods in a rule table and reordering the set of state transitions. The composition of the latter two transformations yields the third type, regardless of the composition order.

These transforms are well documented in the literature [Wolfram, 2002], and are useful because they define equivalent dynamical behaviours. For instance, by applying them to, say, elementary rule 110, elementary rules {137, 124, 193} are obtained, and the four of them are said to be in the same dynamical equivalence class.

By comparing the rule table of a CA with the one that resulted from its equivalent rule obtained out of a given transform, it is possible to count the number of state transitions they share. In a sense, this provides a measure of the amount of *internal symmetry* of a CA, with respect to that transformation, whichever it is. For instance, elementary CA 110 has internal symmetry value of 2 in respect to the black-white transformation, since it shares 2 state transitions with its black-white symmetrical rule (elementary CA 137).

On its part, elementary rule 150 has internal symmetry value of 8 according to the black-white transformation. This is the maximum possible value of this measure with elementary CAs. This comes from the fact that the black-white transformation of rule 150 is itself. In fact, any of the three transformations applied to rule 150 yields rule 150 itself, indicating it has maximum internal symmetry value according to any of the three transformations. Just like the number conservation property, any given value of internal symmetry can be couched in terms of a set of equations that the rule has to meet, in particular the maximum internal symmetry, that is addressed herein.

The degree of internal symmetry of a rule can be a relevant measure in any context where a property is shared among all members of a class of dynamical equivalence. In [Wolz and de Oliveira, 2008] and [Kari and Le Gloannec, 2012], for instance, rules with maximum internal symmetry with the composite transformation were key for their findings related to DCT.

4. Cellular Automata Templates

A CA template is a generalisation of the rule table representation, since it is allowed to have variables in the place of the specific cell states that define the standard *k*-ary representation. As a consequence, CA templates have the power to represent whole subsets of CA rule spaces, instead of a rule alone.

As an illustration, consider, for instance, the template (0, 1-x_1, 0, 1, x_2, 1, x_1, 0), where the indexed x symbols are variables representing a state. It represents the subset of elementary CAs with fixed bits at positions 1, 3, 5, 6 and 8, free variables at positions 2 and 4, and the bit at position 7 being the complement of the one at position 2. It is straightforward to obtain the four binary representations of the CAs associated with that template, namely, {(0,1,0,1,0,1,0,0), (0,1,0,1,1,1,0,0), (0,0,0,1,0,1,1,0), (0,0,0,1,1,1,1,0)}, as well as their respective rule numbers, {84, 92, 22, 30}.

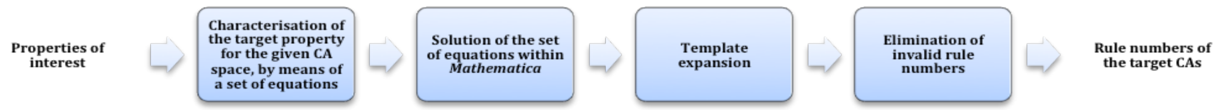

Figure 1: Schematic description of the stages for template creation.

Empowered by *Mathematica*'s built-in equation-solving capabilities, as well as specific algorithms we developed, it became possible to set fixed, variable and dependent state transitions on a rule table, whose result acts as representatives templates of CAs that share the properties of number conservability and maximum internal symetry; these are shown below. It is important to notice that our approach can only be employed on properties that derive directly from relations with the CA rule table.

Figure 1 sketches the sequence of processes related to template creation. So, given a set of properties that should be the basis for defining a family of CAs of interest, first they have to be formalised in terms of a a system of equations. These equation are then given to the *Mathematica* software to be solved; as a result, a set of templates is obtained. These templates are then expanded, so as to generate the rule numbers they represent. However, some of the templates may lead to invalid rule numbers, in the sense that they may not belong to the domain target, that is, the CA family of interest; the invalid rule numbers are eliminated in the subsequent step. The resulting rule numbers are those of the CA family that possess the target property originally defined. Various *Mathematica* based tools have been implemented to facilitate the various processes involved at the four stages, and were reported in [de Oliveira and Verardo, 2014].

4.1 Templates for Number-Conserving Rules

The necessary and sufficient conditions established by Boccara and Fukś [Boccara and Fukś, 2002] for a one-dimensional rule to be conservative can be translated into a set of equations which, when solved by *Mathematica*, yields the equivalent of a template that represents all conservative CAs of a specific space.

Performing this operation for the elementary space, the following template is obtained: (1, 1+x_3-x_4, 1-x_3, 1-x_2-x_3, x_4, x_3, x_2, 0). When expanded, it yields the representations: {(1,0,1,1,1,0,0,0), (1,2,0,0,0,1,0,0), (1,1,0,0,1,1,0,0), (1,1,1,0,0,0,1,0), (1,0,1,0,1,0,1,0), (1,2,0,-1,0,1,1,0), (1,1,0,-

1,1,1,1,0), (1,1,1,1,0,0,0,0)}.

However, it is clear that not all (supposedly) k-ary representations above are valid, since some of them rely on state values outside the range of integers [0, k-1], namely, states 2 and -1. Hence, by discarding the three of them, the complete set of five number-conserving rules of the elementary space is obtained: {184, 204, 226, 170, 240}.

4.2 Templates for Rules with Maximum Internal Symmetry

As the internal symmetry of a CA is also a property that derives directly from its rule table, it is a valid candidate to be generalised into a template. By listing a CA rule table along with its respective transformations, it is possible to establish equality relations between them which, when solved by *Mathematica*, yields a template that represents all CAs that have the maximum value of internal symmetry, according to any subset of the three transformations.

So, performing the operation to find a template that represents all elementary CAs with maximum symmetry according to the black-white transformation, the template ($1-x_1$, $1-x_2$, $1-x_3$, $1-x_4$, x_4, x_3, x_2, x_1) is obtained, which expands to the following elementary rules: {232, 212, 204, 178, 170, 150, 142, 113, 105, 85, 77, 51, 43, 23, 15, 240}.

Analogously, a template representing all CAs with maximum symmetry according to all transformations can be obtained following the same procedure, which leads to ($1-x_1$, $1-x_2$, $1-x_3$, x_2, $1-x_2$, x_3, x_2, x_1), whose expansions leads to the corresponding rule numbers: {204, 178, 150, 105, 77, 51, 23, 232}.

4.3 Composition of Templates

Either templates of number conservation and maximum internal symmetry may be taken as a starting point to the other, therefore leading to a composition of the templates.

For instance, in order to generate all the elementary conservative CAs with maximum internal symmetry values according to the black-white transformation, it suffices to use the template for number-conserving rules of the elementary space as the starting point to the maximum symmetry template; a *Mathematica* based function was implemented with the purpose of accounting for this composition. This leads to the template (1, $1-x_2$, $1-x_3$, $1-x_2-x_3$, x_2+x_3, x_3, x_2, 0), which, once again, can be expanded so as to yield the target rule numbers {204, 170, 240}.

Alternatively, the template with maximum internal symmetry could be used as the starting point for the number conservation template to obtain the same result, that is, (1, $1-x_2$, $1-x_3$, $1-x_2-x_3$, x_2+x_3, x_3, x_2, 0), and the corresponding rule numbers {204, 170, 240}.

5. Concluding Remarks

The idea of representing subsets of a specific cellular automata rule space, where the rules in the set can share a common property, is feasible with the use of CA templates, an extension to the standard rule table based representation of CAs. Although the examples used herein as an or illustration only referred to one-dimensional, binary rules, in fact, the elementary space, the idea seems readily applicable to larger CAs, with larger number of states and more dimensions.

The key idea supporting the use of templates is that the properties to be used have to be couched in terms of well-established relations among the state transitions of the rule. This was the case of the two properties employed in the examples, namely, number-conservation and maximum internal symmetry. As a counterpoint, the notion of reversibility of one-dimensional rules does not seem to be, at least in principle, amenable to template representation, since it is currently not known how (if at all possible) to characterise reversibility in terms of the rule table of a CA.

Also key in our approach is the use of the built-in equation-solving capabilities and algorithms of the *Mathematica* software, in order to solve the set of equations that comes out of the target property characterisation. Although this is not an absolute necessity, manual solution of the set of equations become more and more impractical and time consuming, as the size of the CA space increases. For instance, the set of equations for number conservation in binary CAs increases exponentially with the neighbourhood size of the CA.

Templates for the rules in the same dynamical class in the elementary space have appeared previously in the CA literature, such as in [Li and Packard, 1990]. But, in these cases, the notion was not at all couched in the conceptual framework we have put forward, that allows templates to be effectivelly defined for rules having maximum internal symmetry value, let alone the possibility of representing further CA properties.

It stands as a future work to find new algorithms that would allow template representations of other properties, as well as the enhancement of the current algorithm related to internal symmetry templates, so as to extend the current constraint of only generating maximum internal symmetry, towards also allowing the generation of templates with specific values of internal symmetry, not necessarily maximum. Also, template expansion does not scale up well to very big templates, because of computational demands; this should also be addressed in a follow-up.

Acknowledgements

Pedro de Oliveira thanks the financial support provided by FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo and by Universidade Presbiteriana Mackenzie, and Maurício Verardo is thankful for a fellowship provided by CAPES, the Brazilian agency of its Ministry of Education.

References

[Betel, de Oliveira and Flocchini, 2013] H. Betel, P.P.B. de Oliveira and P. Flocchini. "Solving the parity problem in one-dimensional cellular automata". *Natural Computing*, 12(3):323-337, 2013.

[Boccara and Fukś, 2002] N. Boccara and H. Fukś. "Number-conserving cellular automaton rules". *Fundamenta Informaticae*, 52(1-3):1-13, 2002.

[de Oliveira, 2014] P.P.B. de Oliveira. "On density determination with cellular automata: results, constructions and directions". *Journal of Cellular Automata*, vol.9, 2014. To appear.

[de Oliveira and Verardo, 2014] P.P.B. de Oliveira and M. Verardo. "Representing families of cellular automata rules". *The Mathematica Journal*, Fall 2014. To appear.

[Fukś, 2000] H. Fukś. "A class of cellular automata equivalent to deterministic particle systems". In: S. Feng, A.T. Lawniczak and S.R.S. Varadhan, eds. *Hydrodynamic limits and related topics*. Amer. Math. Soc., Providence, RI, USA, 57-69, 2000.

[Kari and Le Gloannec, 2012] J. Kari and B. Le Gloannec. "Modified traffic cellular automaton for the density classification task". *Fundamenta Informaticae*, 116(1-4):141-156, 2012.

[Li and Packard, 1990] W. Li and N. Packard. "The structure of elementary cellular automata rule space". *Complex Systems*, 4(3):281-297, 1990.

[Schranko and de Oliveira, 2010] A. Schranko and P.P.B. de Oliveira. "Towards the definition of conservation degree for one-dimensional cellular automata rules". *Journal of Cellular Automata,* 5:383-401, 2010.

[Wolfram, 2002] S. Wolfram. *A New Kind of Science*. Wolfram Media Inc., 2002.

[Wolfram Research, 2014] Wolfram Research. "*Mathematica* 9", *http://www.wolfram.com/mathematica*, 2014.

[Wolz and de Oliveira, 2008] D. Wolz and P.P.B. de Oliveira. "Very effective evolutionary techniques for searching cellular automata rule spaces". *Journal of Cellular Automata*, 3(4):289-312, 2008.