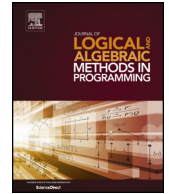




Contents lists available at ScienceDirect

# Journal of Logical and Algebraic Methods in Programming

[www.elsevier.com/locate/jlamp](http://www.elsevier.com/locate/jlamp)


## Exploring nominal cellular automata <sup>☆</sup>

Tommaso Bolognesi <sup>\*</sup>, Vincenzo Ciancia <sup>\*</sup>

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", Consiglio Nazionale delle Ricerche, Pisa, Italy

### ARTICLE INFO

#### Article history:

Received 15 December 2016

Received in revised form 7 June 2017

Accepted 8 August 2017

Available online 1 September 2017

#### Keywords:

Cellular automata

Nominal sets

### ABSTRACT

The emerging field of Nominal Computation Theory is concerned with the theory of Nominal Sets and its applications to Computer Science. We investigate here the impact of nominal sets on the definition of Cellular Automata and on their computational capabilities, with a special focus on the emergent behavioural properties of this new model and their significance in the context of computation-oriented interpretations of physical phenomena. An investigation of the relations between Nominal Cellular Automata and Wolfram's Elementary Cellular Automata is carried out, together with an analysis of interesting particles, exhibiting "nominal" behaviour, in a particular kind of rules, reminiscent of the class of *totalistic* Cellular Automata, that we call "bagged".

© 2017 Elsevier Inc. All rights reserved.

## 1. Introduction

In the field of logical and algebraic methods in Computer Science, a particular class of permutation algebras called *nominal sets* has gained quite some momentum in the last decades, for their appealing role as a *constructive* computational model, that albeit very similar to classical set theory, provides explicit support for *names*, that are attached to elements, and can be freshly allocated while respecting the classical axioms of alpha-convertibility.

Nominal sets are a modern recollection of the *permutation* model that Fraenkel and Mostowski had used to prove independence of the *Axiom of Choice* from the other axioms of set theory, and was re-introduced in Computer Science as the grounds for modelling *abstract syntax* in the presence of *binding* [14,23]. The theory of nominal sets has given rise in the last decades to a novel computing paradigm, based on the notion of *pure name*, with proven capacity to embrace and enhance the full spectrum of Theoretical Computer Science, including syntax and semantics [14,21], automata theory [1], regular expressions [18,15], and several other foundational topics, including the most famous theoretical device, namely *Turing machines* [2].

*Cellular automata* [17] (CAs) are a model of computation originally introduced in the 1940's by Stanislaw Ulam and John von Neumann for modelling biological self-reproduction. In a cellular automaton (CA), cells are arranged in regular, finite-dimensional grids, and assume values from a finite alphabet (also called *colours*, or *states*). Cell updates occur simultaneously at discrete time instants (principle of *synchrony*), the new value of each cell depending of the values of the cells in its neighbourhood (principle of *locality*).

ECAs (for Elementary Cellular Automata) [25] are the simplest form of CA: cells are arranged in a 1D, infinite, or finite and circular, array, they assume binary values, and cell neighbourhoods have *diameter* 3, meaning that the evolution of each cell is determined by its immediate left and right neighbours. In other words, the value  $c_i(t+1)$  of cell  $c_i$  at time  $t+1$  depends

<sup>☆</sup> Research partially supported by ISTI-CNR. The research of the second author was partially supported by the EU FP7 Project QUANTICOL, Grant agreement n. 600708.

<sup>\*</sup> Corresponding authors.

E-mail addresses: [t.bolognesi@isti.cnr.it](mailto:t.bolognesi@isti.cnr.it) (T. Bolognesi), [vincenzoml@gmail.com](mailto:vincenzoml@gmail.com) (V. Ciancia).

on the value at time  $t$  of  $c_i$  itself and of its immediate left and right neighbours, and is expressed by  $f(c_{i-1}(t), c_i(t), c_{i+1}(t))$ , where  $f$  is a boolean function used for all cells (principle of *uniformity*). There exist  $2^8 = 256$  boolean functions of three variables, thus there exist 256 distinct ECAs. According to Wolfram's numbering scheme, if  $f(1, 1, 1) = b_1$ ,  $f(1, 1, 0) = b_2, \dots, f(0, 0, 0) = b_8$ , then the ECA based on function  $f$  is numbered by the decimal representation of bit tuple  $(b_1, \dots, b_8)$ .

Being based on principles of locality and uniformity, CAs have been used for modelling a wide array of natural phenomena, from crystal formation to fluid dynamics, from reaction-diffusion in chemistry to pattern formation in biologic substrata. The contrast between the extreme simplicity of their definition and the richness of their emergent behaviours, as widely explored in [24,25], has induced some scientists to attribute to CAs, or similar simple models, a fundamental role for explaining the complexity of the physical universe as a whole. The idea that the complexity observed in nature be the manifestation of the emergent properties of a computation taking place at the tiniest spacetime scales is sometimes referred to as the *computational universe conjecture*.

The quest for simple theoretical models of digital spacetime is still open. Classical cellular automata provide a foundational example. However, CAs already assume a universal meaning of colours in cells. In this work, instead, we consider a model which is solely based on equality, where the meaning of colours is only defined at the local level. In order to achieve this, we resort to the machinery of *nominal sets*. The main purposes of this work are (i) to come up with a revised definition of cellular automata based on nominal computing concepts, and (ii) to present a preliminary exploration and discussion of the potential advantages that Nominal Cellular Automata might offer over classical CAs when dealing with the computational universe conjecture and, more generally and abstractly, with the emergence of complexity from simple premises.

In Section 2 we provide the general definition of Nominal Cellular Automaton (NCA), which merges naturally the classical definition of CA with the basic assumptions of nominal computing, we define a “bagged” variant of Nominal CA, closely related to the “totalistic”, classical CA, and we classify the spaces of NCAs and bagged NCAs, determining class sizes.

In Section 3 we examine the simplest form of NCA, that we call ENCA (“E” for “elementary”).

In Section 4 we start studying the relations between ENCA and Wolfram's Elementary CA (ECA), identifying the subset of the ECAs that can be *directly* simulated by ENCAs.

In Section 5 we introduce a simple technique for simulating any ECA – including Turing-complete ECA 110 – by NCAs, thus obtaining a first instance of a Turing-complete NCA.

In Section 6 we present some of the emergent patterns that can be observed in the spatio-temporal diagrams of NCAs, compare them with the classical CA scenario, and speculate on the potential benefits that the new model might offer with respect to the computational universe conjecture.

In Section 7 we extend our analysis of emergent properties to the case of bagged NCA, and discover a number of particles that exhibit linear trajectories. Among them, we illustrate existence of particles that have an “internal” and an “external” (boundary) colour, can move the internal colour to the outside upon collision, and can bind the internal colour to a freshly generated name at a specific point in space-time. We find such particles particularly interesting in light of the emergence of complex computational capabilities that can take into account causal dependencies, identity of entities, and communication capabilities. A formal correspondence between *some* totalistic and *some* bagged CA is also proved, analogies and differences of particles that emerge in the two cases are discussed, and possible advantages of the new types of nominal particles in terms of physical spacetime modelling are suggested.

This paper is an updated version of [5]. Besides catering for a more detailed presentation of NCA, previous work is extended by introducing the notion of bagged NCA, studying the corresponding class sizes, and starting to explore their behaviour (see Section 2.2 and Section 7). Such notion proved fundamental for discovering truly “nominal” emergent behaviour, or more specifically, name binding at particle creation time (Section 7.2.1).

## 2. Nominal Cellular Automata (NCA): definitions and classes

While in *classical* cellular automata the values assumed by cells range in a *finite* alphabet of symbols, or states (often called *colours*), in a *Nominal Cellular Automaton* (NCA) cell values are drawn from a *countably infinite* alphabet: they range in an infinite set of symbols, or *names*, that we denote  $N$  and conveniently represent by the natural numbers. A name denotes nothing but itself, and the only operations on names are: (i) compare names for equality, (ii) make a copy of a name, (iii) create a fresh name.

Similar to classical CA, the *transition rule* of a NCA establishes the next value of cell  $c_i$  based on the current values of the cell and of those in its neighbourhood or, more precisely, on the *equality pattern* exhibited by these names. In this paper we restrict to one-dimensional CAs, in which the cells are arranged in a one-dimensional finite or infinite array. We shall use the term *context* of cell  $c_i$  to denote the tuple consisting of  $c_i$  and some of its left and right neighbours, and the term *diameter* to precisely define the number of cells in the context (including  $c_i$ ). For example, the diameter-3 context of  $c_i$  is  $(c_{i-1}, c_i, c_{i+1})$ .

The new value to be assigned to a cell by the transition rule can be either one of the names found in the context at a given position, or a new name. The idea is that the rule is not allowed to access the names generated up to the current computation step: it can only access those immediately found in the context to which it applies, or create a new, fresh one.

It is worth remarking that NCAs are in fact cellular automata defined in a different kind of set theory – namely, that of nominal sets. Let us briefly recall the needed definitions (the interested reader may refer to [14,23] for more details). First,

Download English Version:

<https://daneshyari.com/en/article/4951373>

Download Persian Version:

<https://daneshyari.com/article/4951373>

[Daneshyari.com](https://daneshyari.com)