An Algebraic Characterization of
Fuzzy Cellular Automata

SAMIRA EL YACOUBI AND ANGELO B. MINGARELLI

LAMPS-University of Perpignan 66860, 52, Paul Alduy Avenue,
Perpignan, France
School of Mathematics and Statistics, Carleton University, Ottawa,
Canada, K1S 5B6. E-mail: yacoubi@univ-perp.fr

Received: March 4, 2009. Accepted: April 27, 2009

It is well known that, in the case of a one dimensional two-neighbor situation, there are 256 fuzzy cellular automata obtained by the fuzzification of the disjunctive normal form in classical (Wolfram) boolean cellular automata using classical aristotelian logic. Starting from a single polynomial in three variables we find its invariants under affine linear transformations and show that among these there are precisely eight equivalent polynomials whose span over \( \mathbb{Z}_2 \) generates a vector space of dimension 8 over the field \( \mathbb{Z}_2 \) containing 256 distinct elements. Its elements are, in fact, the transition (local) rules of the fuzzy cellular automata studied by previous authors. Our result allows for an alternate characterization of such fuzzy cellular automata and permits their generalization to arbitrary number of variables over general (finite or infinite) fields thus bypassing the need for a disjunctive normal form approach.

Keywords: fuzzy systems, elementary cellular automata, characterisation

1 INTRODUCTION

Those elementary cellular automata (ECA) considered by Wolfram and others are good examples of discrete systems with simple rules that may produce unusually complex behavior [11]. The introduction of fuzzy cellular automata (or FCA) into the framework is an attempt to make the underlying processes...
continuous; that is, starting from an ECA we “fuzzify” its disjunctive normal form (DNF), as the latter describes its rule [1], and then use this to define a new class of continuous cellular automata. The resulting transition rules are now functions of a continuous variable in a product space, basically a finite product of copies of the closed interval $[0, 1]$.

The FCA we consider were originally introduced to study the impact that state-discretization has on the behavior of these systems. They have also been used to investigate the result of perturbations (e.g. noisy sources, computation errors, mutations, etc.) on the evolution of boolean CA [3]. In addition, these FCA have immediate applications to the theory of neural networks, and we will present this in a study that will be undertaken in a future paper.

Using an analytical approach general methods for detecting the evolution and dynamics of FCA have been recently investigated and results appeared in [2, 5, 7]. It was shown that apart from nine exceptional rules (handled separately), the limiting behavior of these FCA is decidable for finite initial configurations in a homogeneous background of zeros. In fact, with the appropriate interpretation, it was shown in [5] that the limiting behavior (at infinity) of these rules is a continuous function of the initial string. Varying the underlying logic leads to even more general classes of discrete or continuous CA (see [6]) and in some cases (e.g., probabilistic rule 110) it appears as if the continuity of the limiting values on the initial string is preserved so long as the logic itself is a continuous logic.

The aim of this research is to explore the construction of FCA independently of an/the underlying disjunctive normal form. In classical (aristotelian) logic this DNF is necessarily unique but this uniqueness generally fails for continuous or many valued logics (see [6] and the references therein). This led us to consider an algebraic approach to the larger problem of generalizing FCA to more than two neighbors, to more than one dimension, and to arbitrary continuous or many valued logics.

In this paper we keep to the one dimensional case and show how can define the known FCA algebraically for arbitrary neighborhood structures (independently of whether a cell has an even or odd number of neighbors). Starting with the case where a cell has only two neighbors (before and after) we consider a single polynomial in three variables, find its invariants under affine linear transformations and then show that among these there are precisely eight equivalent polynomials in three variables whose linear span over $\mathbb{Z}_2$ generates a vector space of dimension 8 over the field $\mathbb{Z}_2$. The elements of this finite dimensional linear space make up, in fact, all the transition (local) rules of the FCA studied recently by previous authors. This now allows for an alternate characterization of such FCA and permits their generalization to arbitrary number of variables over general fields, thus by-passing the need for the unusually onerous approach using a/the disjunctive normal form.
2 CELLULAR AUTOMATA

A (classical) one-dimensional cellular automaton (CA) may be considered as a linear collection of cells where all cells share the same local space (i.e., the set of values for the cells) the same neighborhood structure (i.e., the cells on each side of a cell), and the same local function or rule (i.e., the function defining the effect of the neighbors on each cell, also called the transition or rule function). The global evolution of the CA is then defined by the synchronous update of all cell values according to repeated applications of the local function to the neighborhood of each cell. A configuration of the automaton is a state of all lattice cells [10].

Cellular automata formed one of the first abstract models for parallel computing. Conceived by John von Neumann [9] in the early 1950’s (apparently on a suggestion by Ulam) to investigate self-reproduction, CA have been used mainly for studying parallel computing methods and the formal properties of model systems. Many more applications may be found in [11].

For a bi-infinite lattice of cells on a line, the local (boolean) space \( \{0, 1\} \), the usual neighborhood structure (left neighbor or before, itself, right neighbor or after), and a rule function \( g : \{0, 1\}^3 \rightarrow \{0, 1\} \), the global dynamics of an elementary CA are defined by a function \( f \) where

\[
f : \{0, 1\}^Z \rightarrow \{0, 1\}^Z
\]

and, for \( x = (\ldots, x_{-2}, x_{-1}, x_0, x_1, x_2, \ldots) \), \( f(x) = y = (\ldots, y_{-1}, y_0, y_1, \ldots) \) where for all \( i \in \mathbb{Z} \),

\[
y_i = g(x_{i-1}, x_i, x_{i+1}).
\]

where \( g \) is the transition or rule function or local rule, defined by the 8 possible local configurations a cell detects in its direct neighborhood:

\[
(000, 001, 010, 011, 100, 101, 110, 111) \rightarrow (r_0, \ldots, r_7),
\]

\[
(r_i \in \{0, 1\}, 0 \leq i \leq 7),
\]

where each triplet above represents a local configuration of the left neighbor, the cell itself, and the right neighbor. In general, the value of the sum \( \sum_{i=0}^{7} r_i \) is referred to as the name of the rule. It is known that the local rule of any boolean CA can be expressed canonically as a disjunctive normal form (DNF), that is,

\[
g(x_1, x_2, x_3) = \lor_{i=0}^{1} \land_{j=1}^{3} x_j d_{ij}
\]

where \( d_{ij} \) is the \( j \)-th digit, read from left to right, of the binary expression of \( i \), and \( x^0 \) (resp. \( x^1 \)) stands for \( \neg x \) (resp. \( x \)).
2.1 Fuzzy Cellular Automata

Instead of using cell values from the set of boolean values \{0, 1\} as initial conditions, the initial string now consists of a set of arbitrary but fixed real numbers (also called fuzzy states) in the closed interval \([0, 1]\) (as opposed to the two-point set \{0, 1\}). The process of fuzzification described below consists in redefining the local discrete rule \(g\) above so that it can now act on triples of real numbers (as opposed to triples of boolean numbers) and map the unit cube \([0, 1]^3\) into the unit interval \([0, 1]\).

The key benefit of this procedure lies in the fact that fuzzification will allow one to move from the discrete (boolean CA) to the continuous (fuzzy CA, or FCA) by extending the domain of definition of the rule in such a way that the new “rule” agrees with the original rule when we restrict its domain to the boolean set \{0, 1\}. Next, we describe the natural method of fuzzifying a given boolean rule the idea having originally appeared in [1]. We adopt the now standard terminology from Flocchini et al. [2].

**Definition 2.1.** A “fuzzy” cellular automaton (or FCA for brevity), is obtained by fuzzifying the local function of a given boolean CA in the following way: For real numbers \(a, b \in [0, 1]\) we redefine the operations \((a \lor b)\) to be \((a + b)\), \((a \land b)\) to be \((ab)\), and \((\neg a)\) to be \((1 - a)\) in the DNF. Thus \(a \lor b = a + b,\) \(a \land b = ab,\) and \(\neg a = 1 - a\), where \(+\) and \(\cdot\) are ordinary addition and multiplication of real numbers.

**Example 2.1.** Since \(90 = 2^1 + 2^3 + 2^4 + 2^6\) we see that its rule number, \(90 = \sum_{i=0}^{7} r_i \cdot 2^i\), forces \(r_i = 1\) for \(i = 1, 3, 4, 6\). Using the DNF above we get the transition function for rule 90, that is,

\[
g_{90}(x_1, x_2, x_3) = \lor_{i,j=1} \land_1 x_j^{d_{ij}},
\]

\[
= (x_1^{d_{11}} \land x_2^{d_{12}} \land x_3^{d_{13}}) \lor (x_1^{d_{11}} \land x_2^{d_{22}} \land x_3^{d_{13}}) \lor (x_1^{d_{11}} \land x_2^{d_{22}} \land x_3^{d_{23}}) \lor (x_1^{d_{11}} \land x_2^{d_{22}} \land x_3^{d_{33}}),
\]

\[
= (x_1^0 \land x_2^0 \land x_3^1) \lor (x_1^0 \land x_2^1 \land x_3^0) \lor (x_1^1 \land x_2^0 \land x_3^0)
\]

\[
\lor (x_1^1 \land x_2^1 \land x_3^0),
\]

\[
= (\neg x_1 \land \neg x_2 \land x_3) \lor (\neg x_1 \land x_2 \land x_3)
\]

\[
\lor (x_1 \land \neg x_2 \land \neg x_3) \lor (x_1 \land x_2 \land \neg x_3) \lor (1 - x_1)(1 - x_2)x_3 + (1 - x_1)x_2x_3
\]

\[
+ x_1(1 - x_2)(1 - x_3) + x_1x_2(1 - x_3),
\]

\[
= x_1 + x_3 - 2x_1x_3.
\]
Hence, the local rule or transition function of FCA 90 is of the form
\[
g(x, y, z) = x + z - 2xz,
\]
for \((x, y, z) \in \{0, 1\}^3\).

Note that the local fuzzy rule 90 defined here maps the triples of zeros and ones as follows:

\[
000, 001, 010, 011, 100, 101, 110, 111 \rightarrow 0, 0, 1, 1, 1, 0, 1, 0.
\]

As required by the process of fuzzification this set of values on the right matches exactly the set of values of boolean rule 90 for the triples on the left above (i.e., both rules have the same “look-up tables”). Thus, (boolean) rule 90 is given by (1) above while fuzzy rule 90 is given by (2) (or (3)).

**Example 2.2.** Rule \(18 = 2 + 2^4\) has a transition function given by
\[
(000, 001, 010, 011, 100, 101, 110, 111) \rightarrow (0, 1, 0, 0, 1, 0, 0, 0).
\]

Its canonical expression being
\[
g_{18}(x_1, x_2, x_3) = (\neg x_1 \land \neg x_2 \land x_3) \lor (x_1 \land \neg x_2 \land \neg x_3),
\]
we obtain its fuzzification in the form
\[
g_{18}(x_1, x_2, x_3) = (1 - x_2) \cdot (x_1 + x_3 - 2x_1 \cdot x_3).
\]

**Example 2.3.** Rule \(30 = 2 + 2^2 + 2^3 + 2^4\) has its local rule expressed by
\[
(000, 001, 010, 011, 100, 101, 110, 111) \rightarrow (0, 1, 1, 1, 1, 0, 0, 0).
\]

Its canonical expression being
\[
g_{30}(x_1, x_2, x_3) = (\neg x_1 \land \neg x_2 \land x_3) \lor (\neg x_1 \land x_2 \land \neg x_3) \lor (\neg x_1 \land x_2 \land x_3) \lor (x_1 \land \neg x_2 \land \neg x_3)
\]
\[
\lor (x_1 \land \neg x_2 \land x_3)
\]
its fuzzification turns it into
\[
g_{30}(x_1, x_2, x_3) = x_1 + x_2 + x_3 - 2x_1x_2 - x_2x_3 - 2x_1x_3 + 2x_1x_2x_3.
\]

### 3 AN ALGEBRAIC APPROACH TO FCA

Prior to presenting our equivalent definition of an FCA rule we adopt the following multi-index notation for ease of use: As usual \(\mathbb{R}^d\) (resp. \(\mathbb{N}\)) will denote...
n-dimensional Euclidean space (resp. the set of all non-negative integers).
Let $x \in \mathbb{R}^n, x = (x_1, x_2, \ldots, x_n), \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n), \alpha_i \in \mathbb{N}, 1 \leq i \leq n$, a multi-index. The symbol $x^\alpha$ will stand for an ordinary product of $x_i^{\alpha_i}$, i.e.,

$$x^\alpha = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdots x_n^{\alpha_n}.$$ 

For a given multi-index $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, the expression $a_\alpha$ will denote a term of the form

$$a_\alpha = a_{\alpha_1, \alpha_2, \ldots, \alpha_n}.$$ 

The symbol $\mathbb{N}^n$ will denote the space of all integer valued multi-indices defined by

$$\mathbb{N}^n = \{ \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n) : \alpha_i \in \mathbb{N}, 1 \leq i \leq n \}.$$ 

For a given multi-index $\alpha \in \mathbb{N}^n$ we use the notation

$$|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_n,$$

for the length of an element in $\mathbb{N}^n$.

With this notation a polynomial $f(x)$ in the $n$ (real) variables $x = (x_1, x_2, \ldots, x_n)$ may be written in the more compact form

$$f(x) = \sum_{\alpha \in \mathbb{N}^n} a_\alpha x^\alpha. \quad (6)$$

Thus an arbitrary cubic polynomial $f(x_1, x_2, x_3)$ in the three variables $(x_1, x_2, x_3)$ is necessarily of the form $f(x) = \sum_{|\alpha| \leq 3} a_\alpha x^\alpha$ where the summation extends over all those multi-indices $\alpha$ in $\mathbb{N}^3$ whose length does not exceed 3. From this it follows that the general cubic polynomial of degree 3 in the three variables $(x, y, z)$ takes the form

$$f(x, y, z) = a_{000} + a_{100}x + a_{010}y + a_{001}z + a_{110}xy + a_{101}xz + a_{011}yz + a_{111}xyz + a_{200}x^2 + a_{201}xy^2 + a_{202}xz^2 + a_{210}x^2y + a_{212}x^2z + a_{220}y^3 + a_{203}z^3 + a_{030}y^3 + a_{003}z^3. \quad (7)$$

We recall some basic notions from invariant theory [8]. Let $f, g$ be two polynomials in $n$ real variables. We say that $f$ is equivalent to $g$ if and only if there is a collection of affine transformations $x_i \mapsto a_i x_i + b_i, i = 1, 2, \ldots, n, a_i, b_i \in \mathbb{R}$ such that

$$f(a_1 x_1 + b_1, a_2 x_2 + b_2, \ldots, a_n x_n + b_n) = g(x_1, x_2, \ldots, x_n),$$

for all $(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$. 
Example 3.1. The polynomial

\[ f(x, y, z) = -24 + 8x + 6y + 12z - 2xy - 3yz - 4xz + xyz \]  

(8)

is equivalent to the polynomial \( g \) defined by \( g(x, y, z) = xyz \) since

\[ f(x + 3, y + 4, z + 2) = xyz, \quad (x, y, z) \in \mathbb{R}^3. \]

On the other hand, the polynomials \( g \) and \( h \) where \( h(x, y, z) = x^3 \) are not equivalent since there is no affine transformation that relates \( g \) to \( h \).

In order to characterize the traditional FCA rules algebraically we start with the special polynomial \( g(x, y, z) = xyz \) and consider all those polynomials \( f \) equivalent to it over the base field \( \mathbb{R} \). This class can be characterized in a straightforward manner so that, for instance, the function \( f \) defined in (8) is one of its elements. We focus here on the case of classical FCA as defined in the previous sections.

Lemma 3.1. Let \( f \) be a polynomial of three variables (also called a ternary form) of degree \( m \geq 1 \) with real coefficients expressed in the form (6).

1. If \( f \) is equivalent to the polynomial \( g \) defined by \( g(x, y, z) = xyz \) then \( a_{ijk} = 0 \) whenever any one of the indices \( i, j, k \) exceeds 1.

2. Let \( f \) be a polynomial of the form

\[ f(x, y, z) = a_1 + a_2x + a_3y + a_4z + a_5xy + a_6xz + a_7yz + a_8xyz, \]  

(9)

where \( a_i \in \mathbb{R}, i = 1, 2, \ldots, 8 \). Then there are constants \( a, b, \ldots, g \) such that for all \( (x, y, z) \in \mathbb{R}^3 \) we have \( f(ax + b, cy + d, ez + g) = xyz \) if and only if the following relations hold:

\[ a_8ace = 1 \]  

(10)

\[ a_7 + a_8d = 0 \]  

(11)

\[ a_6 + a_8b = 0 \]  

(12)

\[ a_5 + a_8g = 0 \]  

(13)

\[ a_4 + a_6d + a_7b + a_8bd = 0 \]  

(14)

\[ a_3 + a_5b + a_6g + a_8bg = 0 \]  

(15)

\[ a_2 + a_5d + a_7g + a_8dg = 0 \]  

(16)

\[ a_1 + a_2b + a_3d + a_4g + a_5bd + a_6dg + a_7bg + a_8bdg = 0. \]  

(17)

Proof. 1) Write \( f \) in the form (6) with \( n = 3 \) and let its degree be arbitrary (so long as it exceeds 1). Since

\[ f(ax + b, cy + d, ez + g) = xyz \]  

(18)
for all \((x, y, z) \in \mathbb{R}^3\) it follows that \(a \neq 0\). Taking the second partial derivative of both sides of (18) with respect to \(x\) it follows that for fixed \(y, z\) the polynomial \(f(x, y, z)\) cannot contain any terms of degree two or higher in \(x\). Repeating this argument for the other two variables we find that \(c \neq 0\) and \(e \neq 0\) and, as a consequence, our polynomial \(f(x, y, z)\) cannot contain any terms of degree two or higher in either \(y\) or \(z\). It follows that \(f(x, y, z)\) can be at best of degree 1 in each variable separately and this is equivalent to the stated result.

2) Using the first part we know that \(f\) is of the form (9). The result is now obtained by comparing coefficients and simplifying the rest. Note that comparing both sides of (18) gives the set of equations

\[
\begin{align*}
    a_8 ace & = 1 \quad (19) \\
    a_7 ae + a_8 ade & = 0 \quad (20) \\
    a_6 ce + a_8 bce & = 0 \quad (21) \\
    a_5 ac + a_8 agc & = 0 \quad (22) \\
    a_4 e + a_6 de + a_7 be + a_8 bde & = 0 \quad (23) \\
    a_3 c + a_5 bc + a_6 cg + a_8 bcg & = 0 \quad (24) \\
    a_2 a + a_5 ad + a_7 ag + a_8 adg & = 0 \quad (25) \\
    a_1 + a_2 b + a_3 d + a_4 g + a_5 bd + a_6 dg + a_7 bg + a_8 bdg & = 0. \quad (26)
\end{align*}
\]

Now (19) and (26) are the same as (10) and (17) respectively. Since \(ae \neq 0\) from the first part, we see that (11) is equivalent to (20). Similarly, since \(ce \neq 0\), we get that (12) is equivalent to (21). Once again since \(ac \neq 0\) there follows that (13) is equivalent to (22). Continuing in this way we find that (14) (resp. (15), (16)) is equivalent to (23) (resp. (24), (25)) since \(e \neq 0\) (resp. \(c \neq 0, a \neq 0\)). This completes the proof.

**Corollary 3.2.** The class of all ternary forms equivalent to \(g\) (and distinct from \(g\) itself) is non-empty.

**Proof.** Indeed, it contains the form appearing in Example 3.1.

We narrow down this previous class by requiring that the affine transformations in question be surjective maps on the closed unit interval \([0, 1]\).

**Theorem 3.3.** Let \(f\) be any ternary form equivalent to \(g\) and assume that every affine transformation effecting this equivalence (cf. (18)) when restricted to \([0, 1]\) is surjective on \([0, 1]\). Then \(f\) must be one of eight ternary forms:

\[
\begin{align*}
    &xyz, yz - xyz, xz - xyz, xy - xyz, z - xz - yz + xyz, y - xy - yz + xyz, z - xz - xy - xyz, 1 - x - y - z + xy + xz - xyz.
\end{align*}
\]
Proof: First we make the simple observation that if \( A : [0, 1] \to [0, 1] \) is a surjective affine transformation then it is necessarily of the form \( A(t) = t \) or \( A(t) = 1 - t \) and that such ternary forms are necessarily of the form (9), by Lemma 3.1. Next, let \( f \) be a ternary form such that, say, \( f(1 - x, y, z) = xyz \) for all \((x, y, z) \in [0, 1]^3 \). Since, for \( x \in [0, 1], \ 1 - x \in [0, 1] \) we may interchange \( x \) and \( 1 - x \) to find that \( f(x, y, z) = (1 - x)yz = yz - xyz \). Hence \( f \) is uniquely determined by the surjective criteria imposed upon the affine transformations. Since each affine map under consideration is one of two types there are only \( 2^3 = 8 \) possible combinations for such affine maps, namely \((x, y, z), (1 - x, y, z), (x, 1 - y, z), (x, y, 1 - z), (1 - x, y, 1 - z), (1 - x, 1 - y, z), \) and \((1 - x, 1 - y, 1 - z) \). The remaining possibilities for \( f \) are now obtained as before. 

Let \( \mathbb{Z}_2 \) be the finite field consisting of the two elements \( \{0, 1\} \). We use the notation \( \mathbb{Z}_2^m \) for the cartesian product of \( m \)-copies of \( \mathbb{Z}_2 \). The set \( S \) of eight ternary forms found in Theorem 3.3 can be rewritten more compactly as

\[
S = \{x^{[i]}y^{[j]}z^{[k]} : \{i, j, k \} \in \mathbb{Z}_2^2 \} \quad (27)
\]

where \( x^{[1]} \equiv x \) and \( x^{[0]} \equiv 1 - x \). We assimilate our results in our main theorem.

**Theorem 3.4.** The set \( S \), defined in (27), forms a basis for a vector space \( V \) over \( \mathbb{Z}_2 \) of dim \( V = 8 \) (\( = 2^3 \)). The space \( V \) consists of all the 256 (\( = 2^8 \)) traditional FCA transition functions.

### 3.1 FCA rules for arbitrary neighborhood structures

It is now straightforward to define FCA rules for arbitrary neighborhood structures, including those that are not necessarily symmetric (such as FCA rules defined on \( \mathbb{R}^n \) with \( n > 3 \) an even integer).

Let us examine briefly the case \( n = 4 \). We consider all those polynomial maps in four variables (quaternary forms) and of these we find those maps that are equivalent (via general affine transformations) to the polynomial \( xyzw \) over \( \mathbb{R} \). Narrowing down this class by considering only affine surjective maps on \([0, 1] \) we will obtain \( 2^4 = 16 \) elements whose linear span over the base field \( \mathbb{Z}_2 \) will yield a new class of FCA in four variables corresponding to (necessarily) asymmetric neighborhood structures.

The 16 elements whose span over \( \mathbb{Z}_2 \) gives the \( 2^{16} = 65536 \) new FCA in four variables are: \( xyzw, (1 - x)yzw, x(1 - y)zw, xy(1 - z)w, xy(1 - w), (1 - x)(1 - y)zw, (1 - x)y(1 - z)w, (1 - x)y(1 - z)w, (1 - x)y(1 - z)w, (1 - x)y(1 - z)w, (1 - x)y(1 - z)w, (1 - x)(1 - y)zw, (1 - x)(1 - y)zw, (1 - y)(1 - z)w, x(1 - y)(1 - z)w, x(1 - y)(1 - z)w, x(1 - y)(1 - z)w, x(1 - y)(1 - z)w, \) a set of polynomials that is more compactly written as

\[
S = \{x^{[i]}y^{[j]}z^{[k]}w^{[m]} : \{i, j, k, m \} \in \mathbb{Z}_2^4 \}.
\]
Observe that any $\mathbb{Z}_2$-linear combination of these special elements maps the unit hypercube (i.e., $[0, 1]^4$) onto $[0, 1]$ since each such polynomial is non-negative in the hypercube and the sum of all these 16 elements is identically equal to 1. In this case, the vector space $V$ over $\mathbb{Z}_2$ spanned by the basis elements in $S$ has dimension equal to 16. Of course, the general situation where we consider an $n$-ary form gives a vector space of dimension equal to $2^n$ where $n$ is the number of variables. This algebraic setting appears to be more natural than using the traditional approach of fuzzifying the disjunctive normal form (which would lead to exactly the same class of rules).

It is also to be noted that when the construction of this section is applied to the FCA rules defined on odd dimensional spaces (corresponding to a cell having an odd number of neighbors) we obtain the exact same set of rules. Hence our generalization is very effective for any number of variables and, of course, for any base field other than $\mathbb{Z}_2$.

Remark 1. Observe that our construction depends on the representative, here denoted by $g$, chosen initially from the class of transition rules of one-dimensional FCA (see [4] for a complete list of FCA transition rules). In the case of two-neighbors we chose $g(x, y, z) = xyz$ as our distinguished element since it has the simplest form. We could, however, have chosen another FCA transition function in its place so long as each of the variables $x, y, z$ appear in the range of the transition rule.

Thus, for example, let’s define $g$ by $g(x, y, z) = x - xy + z - xz$ (the transition rule for FCA 58). Our construction would then give rise to 8 basis elements (possibly different from those appearing in Theorem 3.3) but their $\mathbb{Z}_2$-span would still generate the same vector space $V$. The reason for this is built into the construction.

Finally note that not every transition rule can be used in place of the special ternary form, $xyz$, since, e.g., $g(x, y, z) = 1 - xy$ (the transition function for FCA 63) can never, under our construction, generate transition functions containing a $z$-term and so the resulting set of 8 elements cannot span all of $V$.

3.2 FCA rules for continuous or multivalued logics
For background material on this section we refer the reader to [6]. A few remarks are in order: First, we can modify the underlying logic in the traditional definition of FCA (cf., Definition 2.1) to any other continuous logic thereby obtaining new classes of FCA. This program was initiated in [6].

Thus, for example, let us consider CA defined using a probabilistic logic. In this logical system a continuous logic is defined on the closed interval $[0, 1]$ as follows: For real numbers $a, b \in [0, 1]$ we define the “or”, “and”, “not” operations respectively by $(a \lor b) = (a + b), (a \land b) = a + b - 2 \cdot a \cdot b$, and $(\neg a) = (1 - a)$ where $+$ and “.” are ordinary addition and multiplication of real numbers.
Even though a discussion of (the probabilistic) CA 110 in this logic was presented in [6] it would be very time-consuming to write down all the remaining 253 CA transition functions because of the laborious nature of the calculations derived from using the DNF. Nevertheless, we can approach this study algebraically as follows. For a given cell consider the two-neighbor situation for simplicity (so that our transition functions are defined on \([0, 1]^3\)). Then the “polynomial” transition function \(g\) originally defined by using the DNF for CA 128, now gets interpreted as

\[
g(x, y, z) = (x(1 - y) + y(1 - x))(1 - z) + z(1 - x(1 - y) + y(1 - x))
\]

\[
= x + y + z - 2xy - 2xz - 2yz + 4xyz.
\]

Of course, \(g\) still maps the unit cube onto the unit interval. Arguing as before we consider all ternary forms \(f\) such that \(f\) is equivalent to \(g\) by way of affine transformations that map \([0, 1]\) surjectively onto itself. Once again we get 8 such functions and the \(f\)’s are defined by setting \(f(x, y, z) = g(1 - x, y, z)\), etc. The \(\mathbb{Z}_2\)-span of these 8 forms (being linearly independent) then generates a vector space of 256 elements but this time, these make up the “probabilistic CA.”

The method can thus be extended to define continuous CA on underlying continuous or multivalued logics albeit with some caution. The trouble may occur when the 8 new forms are created using our construction as these may happen to be linearly dependent. Of course, this linear dependence property depends very strongly on the form of \(g(x, y, z)\) (which is not necessarily a polynomial in this generality). So, one would need to impose an additional condition such as, for given \(g\), “we assume that the 8 functions

\[
g(x, y, z), g(1 - x, y, z), g(x, 1 - y, z), \ldots, g(1 - x, 1 - y, 1 - z)
\]

are linearly independent.” If such an additional condition is imposed then the method presented here will generate a vector space of dimension 8 over \(\mathbb{Z}_2\). Any element \(u\) of this vector space may then be identified with a unique continuous CA having a particular rule name, that is, some number between 0 and 255 (obtained by regarding the coordinates of \(u\) relative to the previous basis as the binary expansion of a number in this range).

Normally, the sum of two transition rules is not a transition rule. However, this result is true provided the transition rules are written as a linear combination of the basis vectors discussed above and the addition is then effected modulo 2. The final result is simply an observation based on the preceding material and so its proof will be omitted.

In the general case define the set (see (27))

\[
S_n = \{x_1^{[i_1]}x_2^{[i_2]}\ldots x_n^{[i_n]} : \{i_1, i_2, \ldots, i_n\} \in \mathbb{Z}_2^n\}
\]
Theorem 3.5. Consider the linear span of the set $S_n$ over $\mathbb{Z}_2$. Define an operation "⊕" on $S_n$ which amounts to addition modulo 2. Then $(S_n, ⊕)$ is an additive abelian group whose elements ($2^n$ of them) correspond to all the FCA in $n$-variables.

4 CONCLUSION

In closing we can say that the classical FCA found in the current literature can be thought of as arising from the $\mathbb{Z}_2$-span of a set of 8 equivalent ternary forms (in the sense of invariant theory). Indeed, we have presented a purely algebraic approach to the definition of fuzzy cellular automata as considered in the many papers [1], [2], [3], [5], [7] etc. In so doing we have defined a very general framework for the development of the area to the case of an arbitrary number of variables (states) in general finite fields.

ACKNOWLEDGEMENT

This research was undertaken while the second author was a Visiting Professor at the Université de Perpignan, France, in May, 2008. Partial funding by a grant from La Région Languedoc Roussillon is hereby acknowledged.

REFERENCES